E-TRIALS

Developing a Web Application For Educational Research

by

Nicholas Krichevsky

Kamryn Spinelli

13th of May 2020

Submitted to:
Dr. Neil Heffernan
Dr. Korinn Ostrow
Mr. Ryan Emberling

Abstract

As online learning becomes increasingly common, significant interest in educational research has grown at WPI and other universities. One university-based platform for online learning, ASSISTments, is unique for its built-in capacity for research. Its research testbed, E-TRIALS (an EdTech Research Infrastructure to Advance Learning Science), has evolved through six years of grant funding to support educational technology and learning science researchers to conduct randomized controlled experiments in authentic classwork and homework settings at scale. In this IQP report, we outline the issues inherent to ASSISTments' research features, including scattered documentation, opaque features, and a high reliance on human intervention by the ASSISTments research team. Considering these issues, and coupled with a project based on design thinking and user research, we propose a series of remedies for these faults, presenting a design for an E-TRIALS web application that will address them. Development of this newly designed web application has begun and the tool will ultimately replace the existing testbed infrastructure.

Acknowledgements

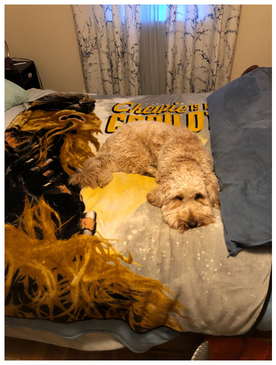
We hereby dedicate this IQP paper to Korinn Ostrow and Ryan Emberling, who often went above and beyond to not only help us achieve our goals, but to make sure that we were well and able to work. This would not have been possible without them.

We also dedicate this paper to the fluffballs that helped us through the challenging times of this paper.

(a) Louie, Ph.D. (Pretty Handsome Dog)

(b) Sassy, the wonderful angry cat found in a woodpile

(c) Emmy, who wants all the attention


(d) Chewie, the dog with a very topical blanket

(e) Cosmo, provider of big wags and naps

The hardworking animals that helped make this project possible

Contents

1 Introduction

1.1 The ASSISTments Platform

ASSISTments is an interactive tutoring system that aims to help teachers provide the best possible educational experience to students. In normal usage, ASSISTments' online platform allows teachers to assign homework problems to their students. ASSISTments is special, however, in that it allows for a wide variety of homework assignments, varying from traditional, multiple choice problem sets, to mastery-based problem sets in which students must answer a certain number of problems accurately in a row (known as "Skill Builders"). These options allow teachers to build homework assignments that can help them pinpoint where their students are struggling with the material, and adjust their lessons appropriately (ASSISTments, 2019b).

The ASSISTments framework also allows researchers to combine several assignment components to produce a randomized controlled trial (RCT), as described in the following section. Researchers can utilize the large population of classrooms using ASSISTments to find potential subjects for their experiments (ASSISTments, 2019c).

Over the years, ASSISTments has undergone numerous changes in personnel, primarily because of students graduating. Despite ASSISTments' broad reach, its constantly changing nature has created a significant amount of technical debt. This has prompted a rewrite of the platform to "version 2.0," which integrates heavily with Google Classroom. Due to the incomplete nature of 2.0, "version 1.0" still exists, and is where research takes place.

1.2 E-TRIALS and RCTs

In addition to ASSISTments' utility as a classroom tool, it can also be used to conduct RCTs to test the efficacy of educational techniques. This feature of ASSISTments is known as E-TRIALS, or and EdTech Research Infrastructure to Advance Learning Science. In short, RCTs can be constructed by randomly assigning students into control or experimental conditions, and measuring their performance. The "if-then-else" section type available to content builders also allows for more sophisticated experiments which measure the efficacy of remedial work.

The strength of E-TRIALS lies in its capability to connect researchers to large sample populations without the financial and time investments typically associated with conducting experiments in education fields. With ASSISTments, researchers do not need to build their own platforms to conduct a study, and as such, can focus their efforts on the content of the study itself. What's more, when the study is ready, a researcher does not have to invest their time and energy into finding schools to participate in their study, as they have access to the wide base of students who use ASSISTments for their classwork and homework.

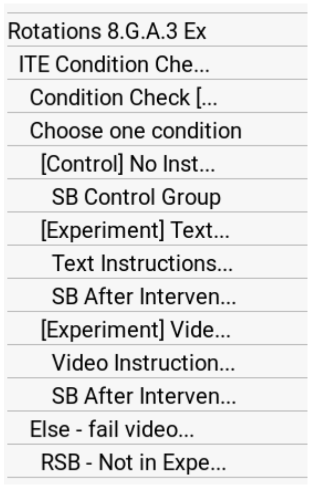
Pre-processing and delivery of RCT data is then handled by the Assessment of Learning Infrastructure (ALI). This is an automatic service that queries the ASSISTments database for all data related to an experiment and delivers that information to the researcher in the form of CSV files at various levels of granularity. Some minor analysis and pre-processing is automatically provided alongside the raw data.

2 E-TRIALS Today

2.1 Usage

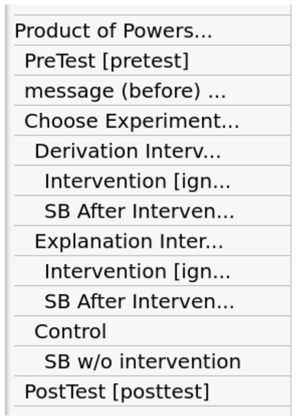
The current ASSISTments study builder (informally and hereafter referred to as the "1.0 builder") represents studies as tree structures. While this aligns naturally with studies' internal representations in the ASSISTments platform, it is incongruous to the way a researcher might describe the flow of their experiment. The crux of this divide is that the components of the study run in sequential order within their tree branch, while a researcher might imagine a flowchart where one component is a child of the component immediately preceding it. This may be confusing to researchers who specialize in education rather than in computer science or technology-based fields.

Figure 1 contains the tree structures of two example studies we built in the 1.0 builder. Sample study A is based on material about rotations in the plane, and compares the efficacy of text hints vs. video hints. Sample study B utilizes a problem set about adding exponents, and aims to evaluate whether it is more effective to give students a derivation or a simple explanation of the technique before giving them exercises.



```
Rotations 8.G.A.3 Ex
 ITE Condition Che...
  Condition Check [...
  Choose one condition
   [Control] No Inst...
    SB Control Group
   [Experiment] Text...
    Text Instructions...
    SB After Interven...
   [Experiment] Vide...
    Video Instruction...
    SB After Interven...
  Else - fail video...
   RSB - Not in Expe...
```

(a) Sample study A in the 1.0 builder



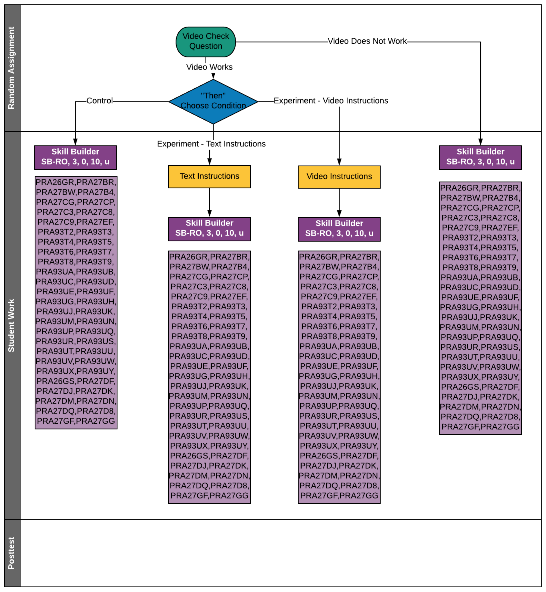
```
Product of Powers...
 PreTest [pretest]
 message (before) ...
 Choose Experiment...
  Derivation Interv...
   Intervention [ign...
   SB After Interven...
  Explanation Inter...
   Intervention [ign...
   SB After Interven...
  Control
   SB w/o intervention
 PostTest [posttest]
```

(b) Sample study B in the 1.0 builder

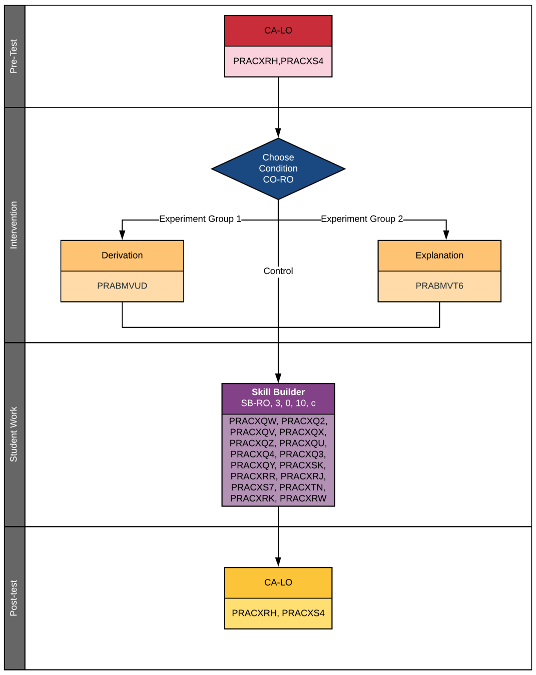
Figure 1: Tree representations of two studies in the 1.0 builder

In sample study A, the top-level if-then-else section type only allows students to participate in the study if they can view YouTube videos (a student may not be able to view YouTube videos if their school blocks such content), which maintains the integrity of collected data. In ASSISTments jargon, this usage of if-then-else is known as a "video check." The "choose one condition" section type randomly separates participating students into control or experimental branches of the study.

Sample study B includes a pretest and a post-test. The post-test is represented in the tree structure as a *sibling* of the random assignment section type, while a researcher might imagine the post-test as a *child*. We can see this difference demonstrated in Figure 2.


Random Assignment
Video Check Question
Video Works
Video Does Not Work
"Then" Choose Condition
Control
Experiment - Video Instructions
Experiment - Text Instructions
Student Work
Skill Builder SB-RO, 3, 0, 10, u
PRA26GR,PRA27BR, PRA27BW,PRA27B4, PRA27CG,PRA27CP, PRA27C3,PRA27C8, PRA27C9,PRA27EF, PRA93T2,PRA93T3, PRA93T4,PRA93T5, PRA93T6,PRA93T7, PRA93T8,PRA93T9, PRA93UA,PRA93UB, PRA93UC,PRA93UD, PRA93UE,PRA93UF, PRA93UG,PRA93UH, PRA93UJ,PRA93UK, PRA93UM,PRA93UN, PRA93UP,PRA93UQ, PRA93UR,PRA93US, PRA93UT,PRA93UU, PRA93UV,PRA93UW, PRA93UX,PRA93UY, PRA26GS,PRA27DF, PRA27DJ,PRA27DK, PRA27DM,PRA27DN, PRA27DQ,PRA27D8, PRA27GF,PRA27GG
Text Instructions
Skill Builder SB-RO, 3, 0, 10, u
PRA26GR,PRA27BR, PRA27BW,PRA27B4, PRA27CG,PRA27CP, PRA27C3,PRA27C8, PRA27C9,PRA27EF, PRA93T2,PRA93T3, PRA93T4,PRA93T5, PRA93T6,PRA93T7, PRA93T8,PRA93T9, PRA93UA,PRA93UB, PRA93UC,PRA93UD, PRA93UE,PRA93UF, PRA93UG,PRA93UH, PRA93UJ,PRA93UK, PRA93UM,PRA93UN, PRA93UP,PRA93UQ, PRA93UR,PRA93US, PRA93UT,PRA93UU, PRA93UV,PRA93UW, PRA93UX,PRA93UY, PRA26GS,PRA27DF, PRA27DJ,PRA27DK, PRA27DM,PRA27DN, PRA27DQ,PRA27D8, PRA27GF,PRA27GG
Video Instructions
Skill Builder SB-RO, 3, 0, 10, u
PRA26GR,PRA27BR, PRA27BW,PRA27B4, PRA27CG,PRA27CP, PRA27C3,PRA27C8, PRA27C9,PRA27EF, PRA93T2,PRA93T3, PRA93T4,PRA93T5, PRA93T6,PRA93T7, PRA93T8,PRA93T9, PRA93UA,PRA93UB, PRA93UC,PRA93UD, PRA93UE,PRA93UF, PRA93UG,PRA93UH, PRA93UJ,PRA93UK, PRA93UM,PRA93UN, PRA93UP,PRA93UQ, PRA93UR,PRA93US, PRA93UT,PRA93UU, PRA93UV,PRA93UW, PRA93UX,PRA93UY, PRA26GS,PRA27DF, PRA27DJ,PRA27DK, PRA27DM,PRA27DN, PRA27DQ,PRA27D8, PRA27GF,PRA27GG
Skill Builder SB-RO, 3, 0, 10, u
PRA26GR,PRA27BR, PRA27BW,PRA27B4, PRA27CG,PRA27CP, PRA27C3,PRA27C8, PRA27C9,PRA27EF, PRA93T2,PRA93T3, PRA93T4,PRA93T5, PRA93T6,PRA93T7, PRA93T8,PRA93T9, PRA93UA,PRA93UB, PRA93UC,PRA93UD, PRA93UE,PRA93UF, PRA93UG,PRA93UH, PRA93UJ,PRA93UK, PRA93UM,PRA93UN, PRA93UP,PRA93UQ, PRA93UR,PRA93US, PRA93UT,PRA93UU, PRA93UV,PRA93UW, PRA93UX,PRA93UY, PRA26GS,PRA27DF, PRA27DJ,PRA27DK, PRA27DM,PRA27DN, PRA27DQ,PRA27D8, PRA27GF,PRA27GG
Posttest

(a) A flowchart representation of sample study A



(b) A flowchart representation of sample study B

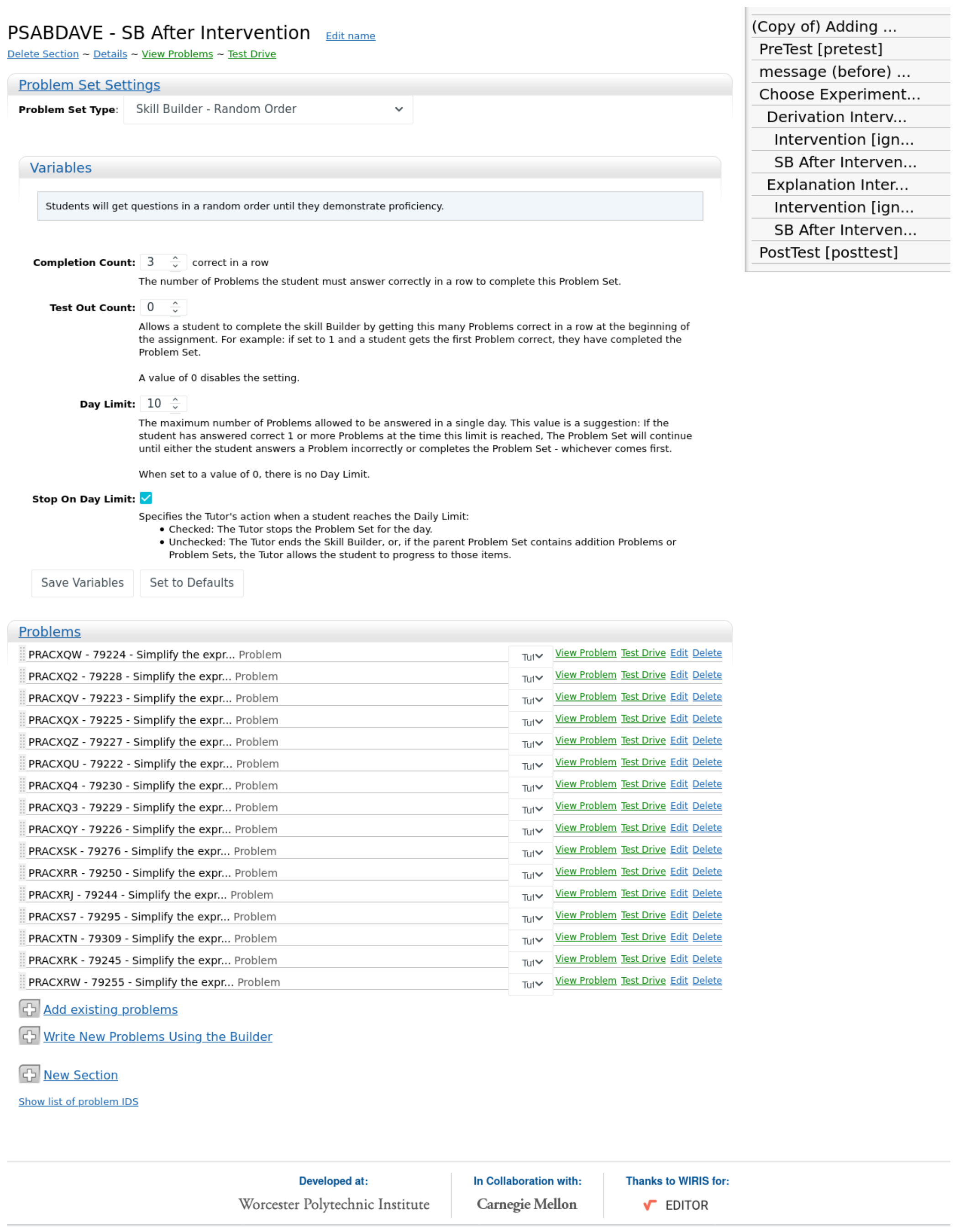
Figure 2: Flowcharts corresponding to the studies in Figure 1

Despite the fact that the tree in Figure 1 and the flowchart in Figure 2 represent the same study, it is much harder to assess what is actually going on from the tree structure. In contrast, the flow chart provides a meaningful visualization of the flow of the study's operations and highlights that, despite there being two branches at the beginning of the study, the Skill Builder after these interventions is the same. As a consequence of the tree structure in the builder, this intervention must be represented by two separate Skill Builders with the same set of problems for the purposes of routing and assignment. This adds potential inconsistencies by which a researcher could remove a problem from one branch but not another, without realizing the discrepancy.

Another drawback of the tree structure used by the builder is the inability to move a child component to a different parent component. While sibling components can be freely reordered, it is impossible to move a component to a higher or lower level in the tree. In the abstract sense, this does not limit the power of the 1.0 builder (since one could simply rebuild the same components in a different location), but it represents extra time investment on behalf of the researcher.

2.2 Documentation

Being on-boarded to the current ASSISTments platform is currently quite time consuming. For basic tasks, the 1.0 builder is simple to understand. For instance, Figure 3 shows the user interface to configure a Skill Builder.



PSABDAVE - SB After Intervention   Edit name

Delete Section ~ Details ~ View Problems ~ Test Drive

Problem Set Settings

Problem Set Type: Skill Builder - Random Order

Variables

Students will get questions in a random order until they demonstrate proficiency.

Completion Count: 3 correct in a row

The number of Problems the student must answer correctly in a row to complete this Problem Set.

Test Out Count: 0

Allows a student to complete the skill Builder by getting this many Problems correct in a row at the beginning of the assignment. For example: if set to 1 and a student gets the first Problem correct, they have completed the Problem Set.

A value of 0 disables the setting.

Day Limit: 10

The maximum number of Problems allowed to be answered in a single day. This value is a suggestion: If the student has answered correct 1 or more Problems at the time this limit is reached, The Problem Set will continue until either the student answers a Problem incorrectly or completes the Problem Set - whichever comes first.

When set to a value of 0, there is no Day Limit.

Stop On Day Limit: ✓

Specifies the Tutor's action when a student reaches the Daily Limit:

- Checked: The Tutor stops the Problem Set for the day.
- Unchecked: The Tutor ends the Skill Builder, or, if the parent Problem Set contains addition Problems or Problem Sets, the Tutor allows the student to progress to those items.

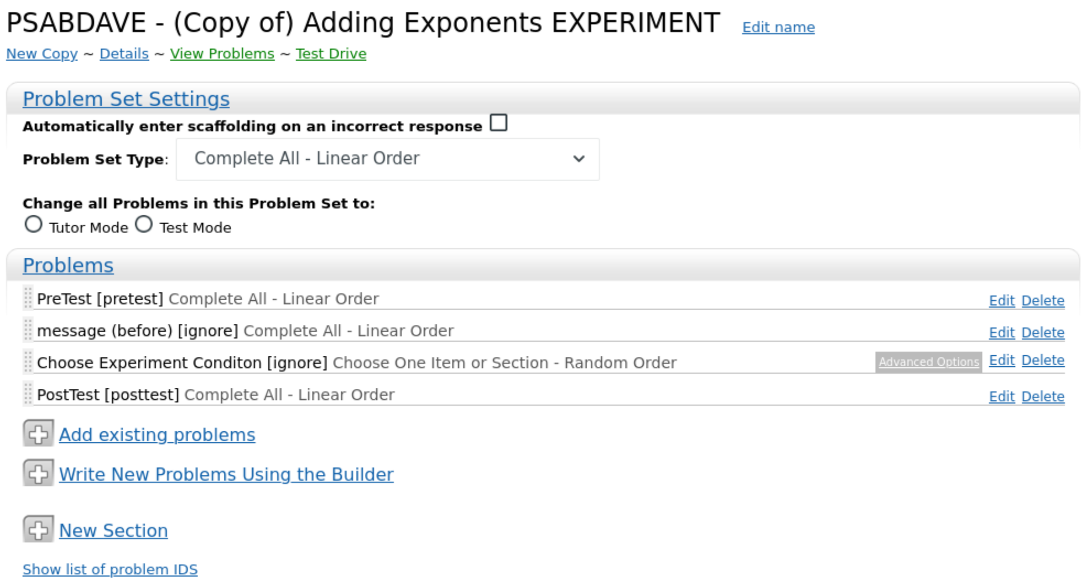
Save Variables   Set to Defaults

Problems

PRACXQW - 79224 - Simplify the expr... Problem
PRACXQ2 - 79228 - Simplify the expr... Problem
PRACXQV - 79223 - Simplify the expr... Problem
PRACXQX - 79225 - Simplify the expr... Problem
PRACXQZ - 79227 - Simplify the expr... Problem
PRACXQU - 79222 - Simplify the expr... Problem
PRACXQ4 - 79230 - Simplify the expr... Problem
PRACXQ3 - 79229 - Simplify the expr... Problem
PRACXQY - 79226 - Simplify the expr... Problem
PRACXSK - 79276 - Simplify the expr... Problem
PRACXRR - 79250 - Simplify the expr... Problem
PRACXRJ - 79244 - Simplify the expr... Problem
PRACXS7 - 79295 - Simplify the expr... Problem
PRACXTN - 79309 - Simplify the expr... Problem
PRACXRK - 79245 - Simplify the expr... Problem
PRACXRW - 79255 - Simplify the expr... Problem

Figure 3: The user interface to set up a Skill Builder within the 1.0 Builder

Despite some jargon, a few things are immediately clear: a student will have to complete three problems to finish the Skill Builder, there is no bypass for students that demonstrate prior knowledge, and they may only attempt ten problems per day. The explanations for these variables are present on the page and are explained in detail. We refer to these details as "embedded" documentation, as they are embedded within the tool.


PSABDAVE - (Copy of) Adding Exponents EXPERIMENT Edit name

New Copy ~ Details ~ View Problems ~ Test Drive

Problem Set Settings

Automatically enter scaffolding on an incorrect response ☐

Problem Set Type: Complete All - Linear Order

Change all Problems in this Problem Set to:
○ Tutor Mode ○ Test Mode

Problems

PreTest [pretest] Complete All - Linear Order Edit Delete

message (before) [ignore] Complete All - Linear Order Edit Delete

Choose Experiment Conditon [ignore] Choose One Item or Section - Random Order Advanced Options Edit Delete

PostTest [posttest] Complete All - Linear Order Edit Delete

Add existing problems

Write New Problems Using the Builder

New Section

Show list of problem IDS

Figure 4: The user interface to add a section within the 1.0 Builder

While some elements of the tool, such as how to add sections to a study, are intuitive, several others are cumbersome. For instance, a researcher cannot make an existing section a subsection of another existing section, without rebuilding the entire tree by hand. More of these idiosyncrasies will be discussed further in the "Exploration" section, but for now, it is worth noting that they exist. Several things are also quite unclear unless you are intimately familiar with the platform, such as the concept of a "problem ID"[1] and how to find one. Worse, important information is often unavailable to researchers (or is difficult to find) at the time of building a study. For instance, the relative difficulty and usage statistics of a problem set are not displayed in the ASSISTments builder, but rather are tabulated in a spreadsheet containing statistics for every single problem on the platform.[2] This information is supremely useful to researchers, but they likely would not know about it from within the tool.

The builder itself is also difficult to use if a researcher is unfamiliar with the structure of existing studies. For instance, many researchers may not even consider the necessity of the previously mentioned video checks, but such a concept is well known by the ASSISTments research team. If these situations or their solutions are not known to researchers, time is spent setting up meetings with the ASSISTments research team, incurring cost to both the researcher and ASSISTments and ultimately inhibiting capacity for both the researcher's study creation and ASSISTments' scalability. To solve this, the research team has built several documentation pages, all attempting to make research on the platform less ambiguous. These are the ASSISTments TestBed page (ASSISTments, 2015), the E-TRIALS TestBed page (ASSISTments, 2019d), the ASSISTments Advanced Builder Instructions Site (ASSISTments, 2017), and a video series on how to use the 1.0 Builder and ALI, produced by Professor Heffernan (Heffernan, 2014).

While this documentation is a good starting point, each of these pages come with their own issues. Universally, they have two major problems: low discoverability and lack of updates. First, these pages are not on any kind of ASSISTments-owned infrastructure, nor on the main assistments.org domain. Instead, they are hosted on Google Sites, and they are not linked on the main ASSSTments website nor the builder. As such, researchers must have the link beforehand, or find them on a search engine. Second, over the years, ASSISTments has grown and more features have been added, but the legacy documentation has not been maintained. This is most obviously apparent when looking at raw data outputs from ALI, where several spreadsheet columns are not even defined in the "Data Dump Glossary" page. At a more fundamental level,

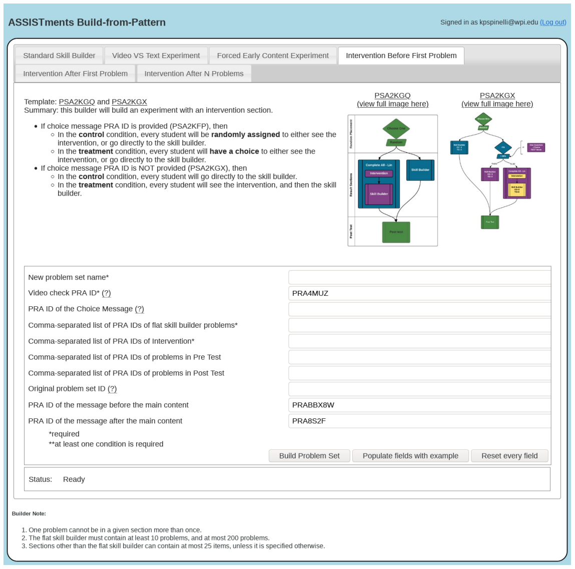
[1]A unique identifier for a problem. Problem IDs aren't inherently problematic to use, but can be unintuitive.
[2]This spreadsheet contained 24,987 rows at the time of writing.

it's unclear how to start research, and how to deploy a study. In the end, these processes are driven by the ASSISTments research team, but a researcher may not be able to gather this from looking at these pages.

Often, even if this kind of information is present, it is buried and difficult to find. This gives the system an extraordinarily steep learning curve that compounds some of the challenges that already exist. This learning curve results in researchers being forced to set up meetings with the ASSISTments research team, which again means both parties are spending resources on resolving the problem.

2.3 Template Tool

One feature aimed at easing the learning curve for study building is the Template Tool. This is a one-way wizard for study creation created by a Ph.D. student at WPI, Mr. Thanaporn "March" Patikorn, for an NSF DIBBs project (#1724889). A researcher can navigate to the template tool, select their desired experimental structure from a handful of templates, and supply the problem IDs to be filled into each component.


ASSISTments Build-from-Pattern

Signed in as kpspinelli@wpi.edu (Log out)

Template: PSA2KGQ and PSA2KGX
Summary: this builder will build an experiment with an intervention section.

- If choice message PRA ID is provided (PSA2KFP), then
  - In the **control** condition, every student will be **randomly assigned** to either see the intervention, or go directly to the skill builder.
  - In the **treatment** condition, every student will **have a choice** to either see the intervention, or go directly to the skill builder.
- If choice message PRA ID is NOT provided (PSA2KGX), then
  - In the **control** condition, every student will go directly to the skill builder.
  - In the **treatment** condition, every student will see the intervention, and then the skill builder.

New problem set name*	
Video check PRA ID* (?)	PRA4MUZ
PRA ID of the Choice Message (?)	
Comma-separated list of PRA IDs of flat skill builder problems*	
Comma-separated list of PRA IDs of Intervention*	
Comma-separated list of PRA IDs of problems in Pre Test	
Comma-separated list of PRA IDs of problems in Post Test	
Original problem set ID (?)	
PRA ID of the message before the main content	PRABBX8W
PRA ID of the message after the main content	PRA8S2F

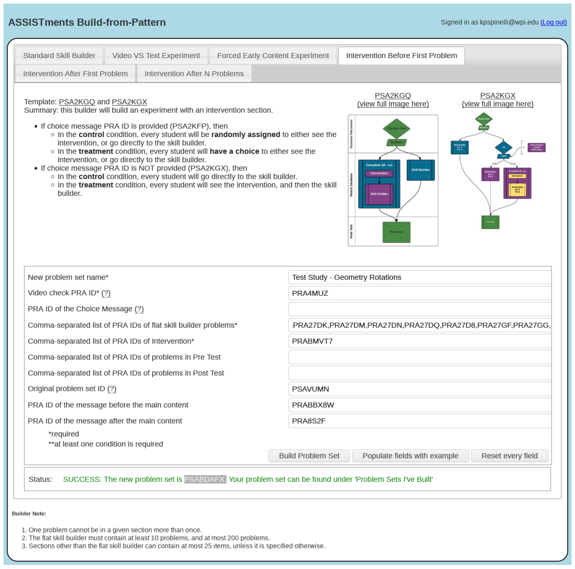
*required
**at least one condition is required

Status: Ready

Builder Note:

1. One problem cannot be in a given section more than once.
2. The flat skill builder must contain at least 10 problems, and at most 200 problems.
3. Sections other than the flat skill builder can contain at most 25 items, unless it is specified otherwise.

(a) An empty template form in the template builder


ASSISTments Build-from-Pattern

Signed in as kpspinelli@wpi.edu (Log out)

Template: PSA2KGQ and PSA2KGX
Summary: this builder will build an experiment with an intervention section.

- If choice message PRA ID is provided (PSA2KFP), then
  - In the **control** condition, every student will be **randomly assigned** to either see the intervention, or go directly to the skill builder.
  - In the **treatment** condition, every student will **have a choice** to either see the intervention, or go directly to the skill builder.
- If choice message PRA ID is NOT provided (PSA2KGX), then
  - In the **control** condition, every student will go directly to the skill builder.
  - In the **treatment** condition, every student will see the intervention, and then the skill builder.

New problem set name*	Test Study - Geometry Rotations
Video check PRA ID* (?)	PRA4MUZ
PRA ID of the Choice Message (?)	
Comma-separated list of PRA IDs of flat skill builder problems*	PRA27DK,PRA27DM,PRA27DN,PRA27DQ,PRA27D8,PRA27GF,PRA27GG,
Comma-separated list of PRA IDs of Intervention*	PRABMVT7
Comma-separated list of PRA IDs of problems in Pre Test	
Comma-separated list of PRA IDs of problems in Post Test	
Original problem set ID (?)	PSAVUMN
PRA ID of the message before the main content	PRABBX8W
PRA ID of the message after the main content	PRA8S2F

*required
**at least one condition is required

Status: SUCCESS: The new problem set is PSABDAFX. Your problem set can be found under 'Problem Sets I've Built'

Builder Note:

1. One problem cannot be in a given section more than once.
2. The flat skill builder must contain at least 10 problems, and at most 200 problems.
3. Sections other than the flat skill builder can contain at most 25 items, unless it is specified otherwise.

(b) An example of a filled template form

Figure 5: Usage of the template tool

The template tool's strength lies in its power to simultaneously reduce both the amount of time a researcher might spend creating a study and the likelihood of mistakes. Especially when a researcher's study naturally aligns with the structure provided by the Template Tool, a researcher could – in theory – avoid touching the 1.0 builder entirely for structural purposes. In practice, this is not quite the case, but the time saved on behalf of both the researcher and ASSISTments research team members is still significant.

The primary limitation of the Template Tool is its one-way nature: a researcher can use the Template Tool to generate a study, but further edits must be made directly in the 1.0 builder. This issue is exacerbated by the rigidity of components in the 1.0 builder. For example, some templates require a video check, but a researcher might not use videos in their study. In this scenario, a researcher who created a study in the Template Tool will have to rebuild the same structure manually, omitting the video check "if-then-else" section at the top level, which to some extent defeats the purpose of having used the Template Tool.

Another, more minor, limitation comes from the fact that it is not always obvious what each of the templates do, unless the researcher has seen a similar study before. For instance, the "Video vs. Text" template does not produce a study that finds the difference between a video or text intervention before a Skill Builder. Rather, it is actually testing whether or not video hints are effective, in lieu of text hints. Though each template comes with a summary, the given summary for this template is "a simple video vs.

text experiment," which does not especially clarify the confusion. Several of the flowcharts available in the Template Tool are confusing, unclear, or inaccurate.

It is important to note that the template tool, in its current form, is a *prototype*. As a proof of concept, the tool demonstrates the power of a template-based approach to expedite study building. However, its flaws also show the need for further development and revision.

2.4 Exploration

With Dr. Ostrow having created more than 100 studies in ASSISTments, Mr. Emberling, Mr. Krichevsky, and Mr. Spinelli also created a total of five example studies to get an understanding of building studies in ASSISTments. We tracked the time spent on each task while building these five studies, as presented in Appendix A. Overall, each study took an average of 200.8 minutes to create. This measures time spent in configuring the structure of the experiment, adding the interventions, and creating a flowchart for the study. However, this time tracking does not address the process of content creation. For studies that utilize videos or custom hints, content creation time could increase dramatically.

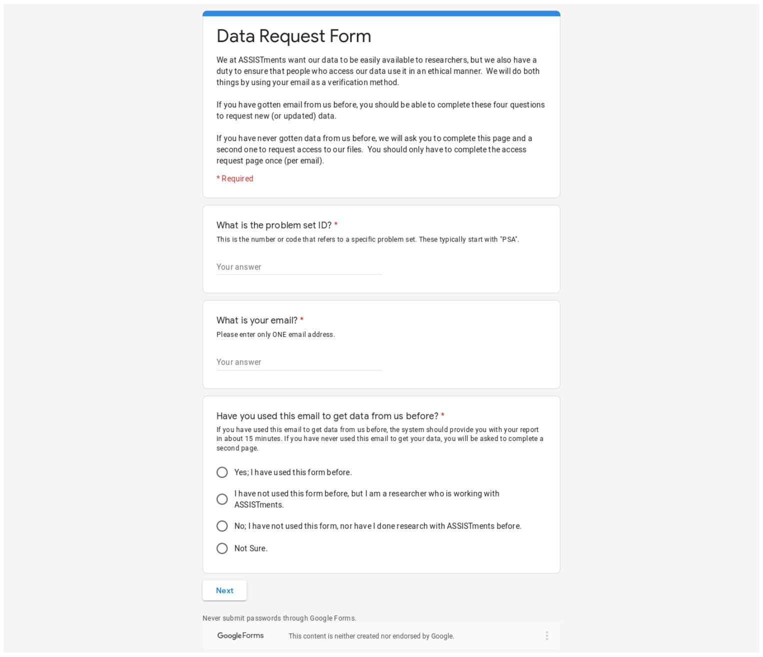
Much of the time spent was associated with two primary timesinks: rebuilding structures created by the template tool, and needing a human (e.g. Dr. Ostrow or another member of the ASSISTments research team) to troubleshoot or guide us in the right direction. Recreating the Template Tool's structures is not just time-consuming, but also raises the probability of errors. Even if a researcher took the time to recreate the structure of a template, it is very hard to know when study building is "done" unless they have prior experience using the tool. These errors often require the ASSISTments research team to step in and either debug the issue or advise on study structure. As ASSISTments continues to scale up, the number of people with enough expertise to assist researchers and debug studies could become a bottleneck for research throughput. Likewise, due to the nature of synchronous communication methods, an email chain between staff and a researcher could take days or weeks to resolve the problem at hand.

Furthermore, despite how useful the Template Tool can be in the situations for which it was designed, it is extremely rigid, lacking options to change the structure of what is produced. If the structure of the template does not match exactly what is desired, it can take a lot of time to figure out how to restructure the study. Furthermore, for a researcher who is new to the platform, the need to manually restructure might not even be obvious without talking to an ASSISTments team member. We personally experienced this when attempting to add alternate interventions to a study. The Template Tool insisted that we add a video check, despite the lack of video in our interventions.

Study troubleshooting is performed via the "Test Drive" feature in the study builder. However, this feature does not allow you to only test drive a specific component, but instead drops you into the top level of the study every time. Due to the nature of random assignment, one might have to test drive several times before being assigned into the tree branch that contains the issue.

2.5 ALI and Data Processing

When a study has run for long enough that a sufficient volume of data has been collected, the natural next step is to retrieve and process that data. The ALI tool handles this step. The researcher begins by filling out a Google form (Figure 6) with their email address and the study's Problem Set ID. ALI queries the ASSISTments database and generates a Google Doc with a report featuring CSV files and brief analysis, all of which are automatically emailed to the researcher.



Data Request Form

We at ASSISTments want our data to be easily available to researchers, but we also have a duty to ensure that people who access our data use it in an ethical manner. We will do both things by using your email as a verification method.

If you have gotten email from us before, you should be able to complete these four questions to request new (or updated) data.

If you have never gotten data from us before, we will ask you to complete this page and a second one to request access to our files. You should only have to complete the access request page once (per email).

* Required

What is the problem set ID? *

This is the number or code that refers to a specific problem set. These typically start with "PSA".

Your answer

What is your email? *

Please enter only ONE email address.

Your answer

Have you used this email to get data from us before? *

If you have used this email to get data from us before, the system should provide you with your report in about 15 minutes. If you have never used this email to get your data, you will be asked to complete a second page.

○ Yes; I have used this form before.

○ I have not used this form before, but I am a researcher who is working with ASSISTments.

○ No; I have not used this form, nor have I done research with ASSISTments before.

○ Not Sure.

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google.

Figure 6: The ALI request Google form

ALI's analysis is driven by study component tags. These are bracketed keywords, such as `[Experiment]` or `[Ignore]`, in the names of components that tell ALI what part of the study to analyze. While the prototype Template Tool automatically inserts ALI tags, one must remember to add them manually when building a study using the 1.0 builder. If a researcher forgets to add ALI tags, or doesn't know they even exist in the first place, the automatic analysis will fail to deliver meaningful statistics. Even in these cases, however, raw data is still provided.[3]

Handling raw data is a difficult task. Users are presented with four CSV files that break down the raw data according to different levels of granularity: action level (which details each action a student takes), problem level (in which each row corresponds to a specific student's work on a specific problem), student level (in which each row corresponds to the performance of a specific student), and student + problem level (in which each row corresponds to a single field or entry of a specific student's work) (ASSISTments, 2019a). These CSV files can contain tens of thousands of rows and are far too large for more researchers to process by hand. Researchers employ a variety of methods for analysis, from processing in Excel to a more programmatic solution such as an R script. The sheer volume of data provided by ALI on a high-traffic study highlights the need for intelligent automatic analysis, although it is simultaneously ill-advised for researchers to rely solely on system generated automation. With that said, researchers must have the proper statistics background to deal with these types of datasets. In an interview with a prospective researcher conducted during the user research portion of this IQP, they indicated that they had a decent background in statistics, but were unsure of some of the nuances that went into demonstrating causal relationships.

An important note is that integration of ALI into the 1.0 builder or Template Tool is currently minor to nonexistent. It is entirely possible for a researcher to build and run a study without ever learning how to access their data. Here the familiar motif repeats: an ASSISTments research team member must step in and help the researcher.

[3]These tags are documented in the ASSISTments Testbed Documentation, but are not in any kind of embedded documentation in the builder itself. It is not obvious to a new researcher that these tags actually have any function, without having sought out this section of the documentation.

3 Design Process

3.1 Considerations

Based on what we learned from our experiences working with the 1.0 builder, in addition to conversations with those on the ASSISTments research team, we knew that we had to ensure that our designs were not only clear to the researcher, but also durable. A considerable amount of the time spent by the research team was regularly focused on testing or debugging studies made by external researchers. In an ideal world, the tooling should constantly validate the study and give the researcher constant feedback. To that end, it should also be abundantly clear what each function does, without having to rely on potentially outdated documentation. With this in mind, we broke our design stage into two phases: finding ways to represent studies that could be built in ASSISTments, and finding ways to make the act of building those studies as intuitive as possible.

3.2 Visualizing Studies

The first problem that we aimed to solve was how to represent studies. We unanimously agreed that the tree structure was not conducive to the complex study designs that ASSISTments had typically made possible. In exploring various options, we considered two factors: flexibility and clarity, neither of which the tree structure met for complex studies.

One design option was the “Scratch” structure, which was modelled after MIT’s Scratch, a tool that allows beginning developers to utilize a drag and drop methodology to building programs (Massachusetts Institute of Technology, n.d.).

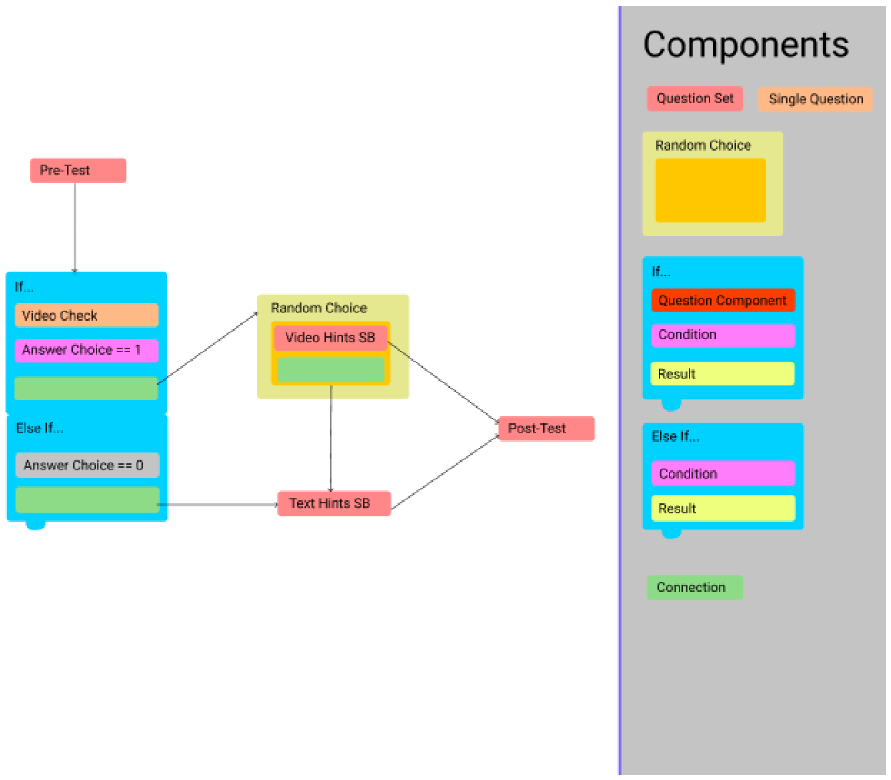


Figure 7: The Scratch structure of study visualization

By connecting different blocks, a researcher could build a study that has different modular components by placing them in predesignated slots, extending the concept of the Template Tool to add both flexibility and clarity. In Figure 7, the researcher has chosen to randomly place subjects into two groups, giving the student either text hints or video hints on their homework problems. This design is highly flexible, and could easily be adapted to a wide array of studies, but what it affords in flexibility, it trades in clarity. This was a common trade-off we saw in many of the designs we investigated. In the Scratch environment, it is not immediately clear that one can connect different components of the study together, but it is clear from looking at the existing structure that such manipulation is at least possible.

Another approach explored was to “stretch” the tree structure out into components which can split into different subcomponents. We will hereby refer to this as the split structure. In this design, each block represents a treatment in the study, and each “split” represents a point at which students are put into different treatments. These splits can later reconvene into a single block, allowing for students that were once divided into different treatments to perform the same task, such as a post-test.

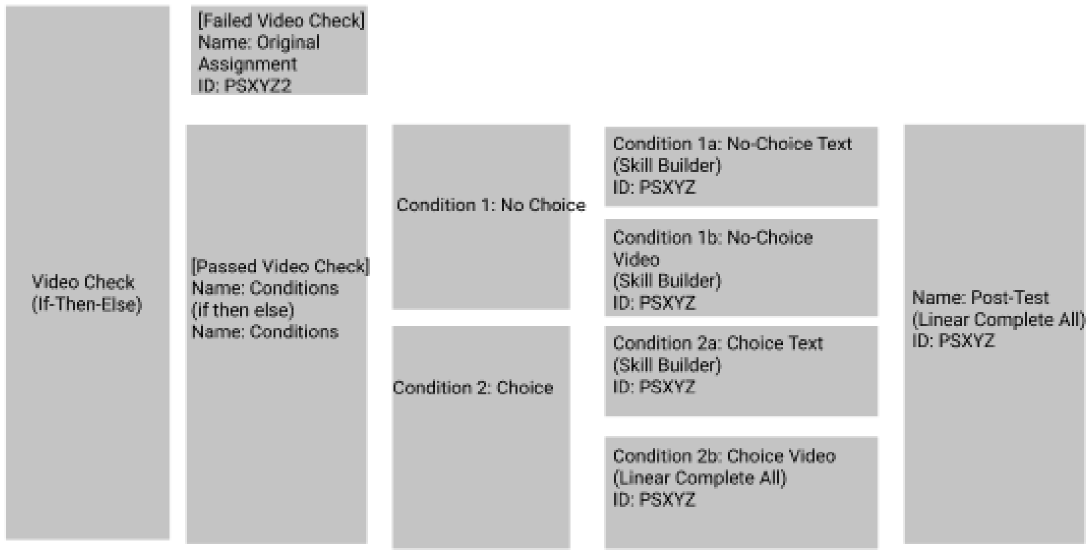


Figure 8: The split structure of study visualization

This design is, initially, quite easy to grasp. One can clearly see where treatments are broken up and where they stay the same. The design went through a couple of iterations in order to more clearly indicate where each treatment ended.

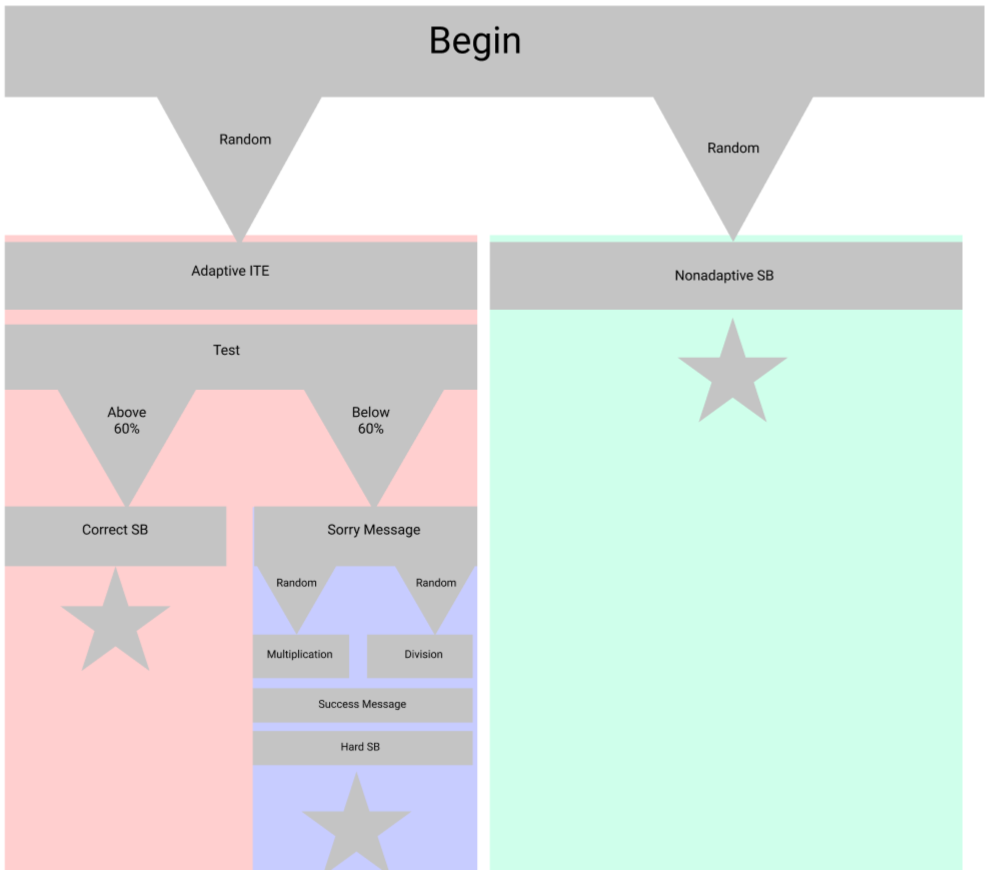


Figure 9: A second iteration of the split structure of study visualization, which shows when each treatment ends

Both versions of the split structure face the same problem where after enough segmentation, treatments will be split so many times they become too small for the text to be legible. The second iteration tried to solve this by splitting each treatment into regions that could be zoomed into focus, but this ultimately proved confusing. This structure had decent flexibility, but it somewhat pigeonholed the design of studies that might need more complicated branching.

Finally, we attempted to revisit the flow chart idea, as it was one that we were familiar with and had potential to be expanded upon. Earlier flow charts we had worked with were helpful and easy to understand, but they were built by hand in LucidChart by ASSISTments research team members, and not using any kind of ASSISTments tooling. If the builder could somehow generate such a flow chart for the researcher in real time, it could provide a more intuitive view of their study.

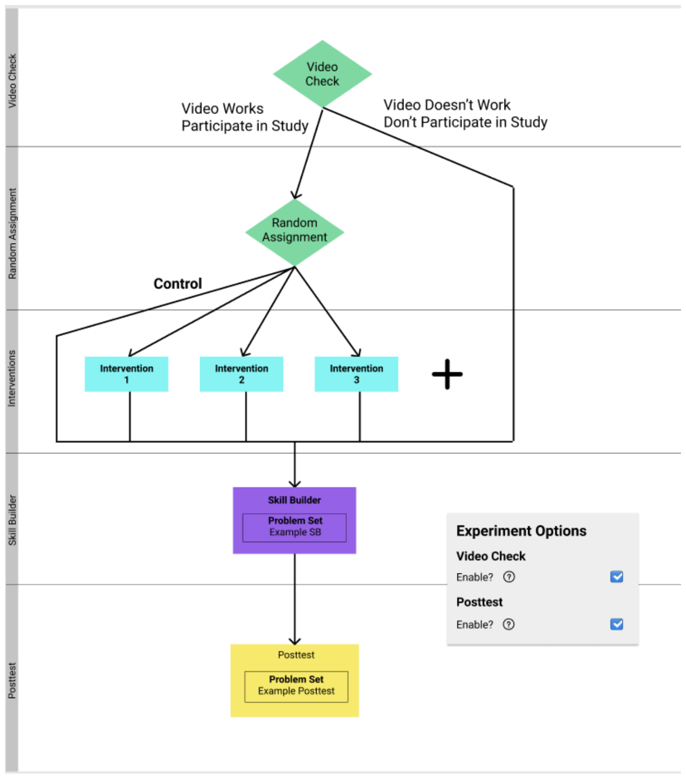


Figure 10: The first iteration of a flow chart concept integrated into the ASSISTments builder

This prototype included checkboxes to enable and disable parts of the flow chart that may not be useful to the researcher. This directly solved the problems that we had run into with the Template Tool by promoting template flexibility.

Allowing modularity for customization ended up being a major part of most of our prototypes. The flow chart design was understandable by researchers as shown by a series of user research interviews, and the toggleable components provided the customization that we had desired when working on our own studies.

3.3 Interactive Mockups

Over a two week period, the authors and two of the advisors on this paper (Dr. Ostrow and Mr. Emberling) worked in parallel to produce what they felt would best serve the needs of researchers based on what we had learned in prior stages of exploration. Using the prototyping tools Figma and Balsamiq, each participant built interactive prototypes that provided a guided study building experience. These prototypes were then given to potential users through a design thinking based user research protocol to see if they could figure out how to build a simple study. What follows is a description of each of the prototypes.

In all cases, the new problem editor created by Mr. Ashish Gurung, a graduate student on the ASSISTments project, was mocked up to accept user input.

3.3.1 Prototype A: Modular Studies

When developing this prototype, Mr. Krichevsky wanted to find a way to make the builder structure more understandable, while also ensuring that researchers would be able to create studies just as complex as those built using the 1.0 Builder, if desired. After looking at several studies, he noticed that many studies could be divided into modular components.

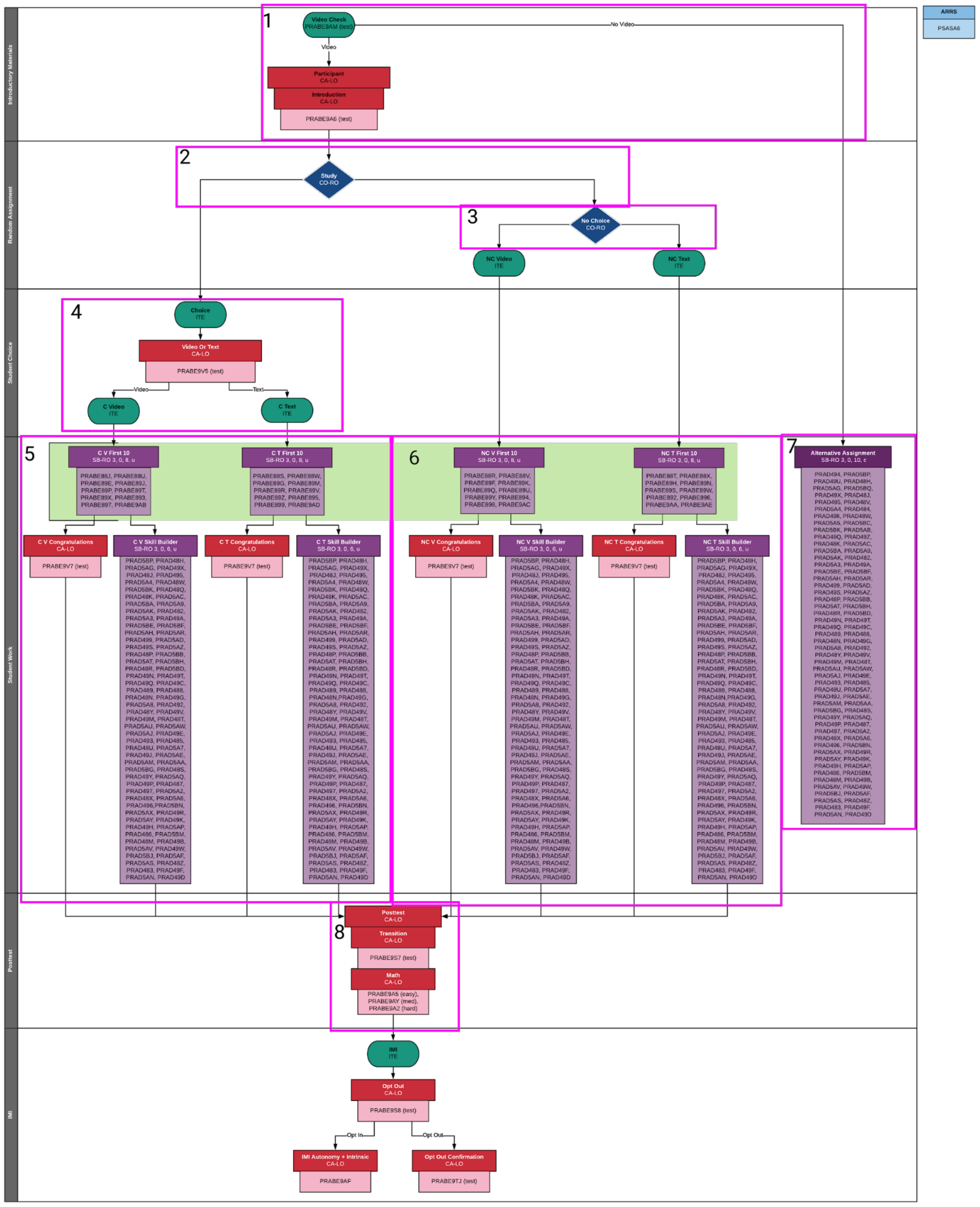


Figure 11: A complex study divided into modular components

In Figure 11, modules 1 and 4 allow for a user to make a choice, modules 2 and 3 involve random assignment, modules 5 and 6 represent identical trees involving two separate Skill Builders for different conditions, and module 8 represents a post-test. Using these components, a researcher could theoretically build a wide range of studies. This inspired a prototype that would allow a researcher to piece together several such modules to build a study.

In this prototype, when a researcher begins the study building process, they are presented with a blank canvas and a library of modules. Each of these modules can be dragged from the library to the canvas to build the desired study. In the prototype, as shown in Figure 12, three modules are provided:

1. Two Conditions - allows a researcher to choose two sets of problems. Each set of problems forms a Skill Builder to which students may be randomly assigned.

2. One Skill Builder - allows a researcher to pick a set of problems from which to build a single Skill Builder.

3. User Choice - allows a researcher to insert choice elements that are controlled via student selection rather than random assignment. For instance, this could be used for video checks, allowing students to log whether or not they can access YouTube videos.

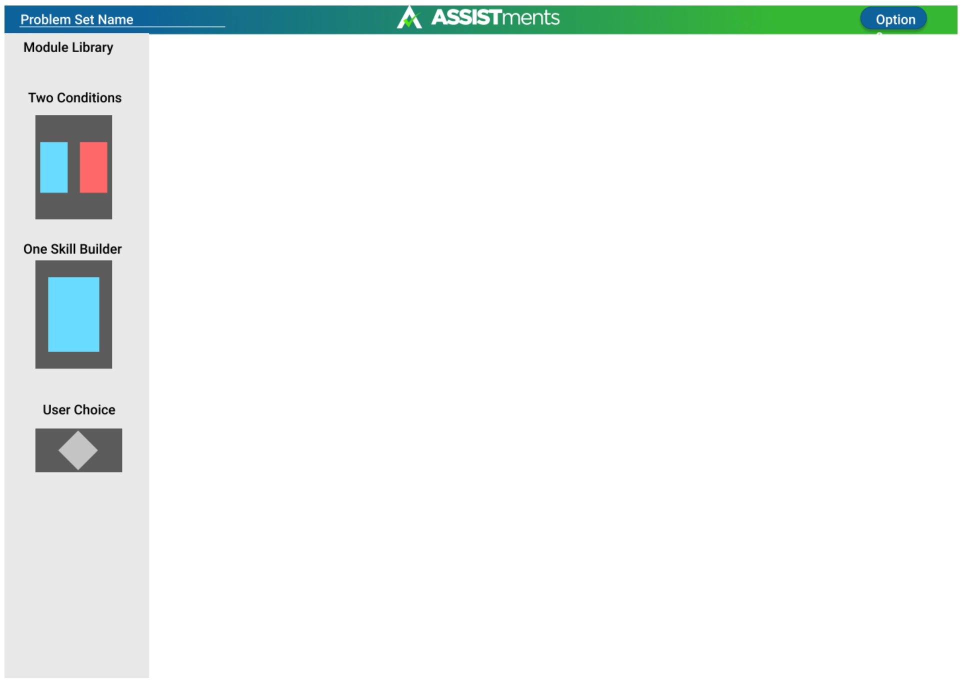


Figure 12: The blank canvas a user is presented in Prototype A

Once a module has been dragged into the canvas, it can be connected with other modules by dragging from the gray rectangle on the edge of a module, as shown in Figure 13. Each connection represents a possible path that students in the study may take, taking cues from both the flowchart and Scratch visualizations from our exploration work.

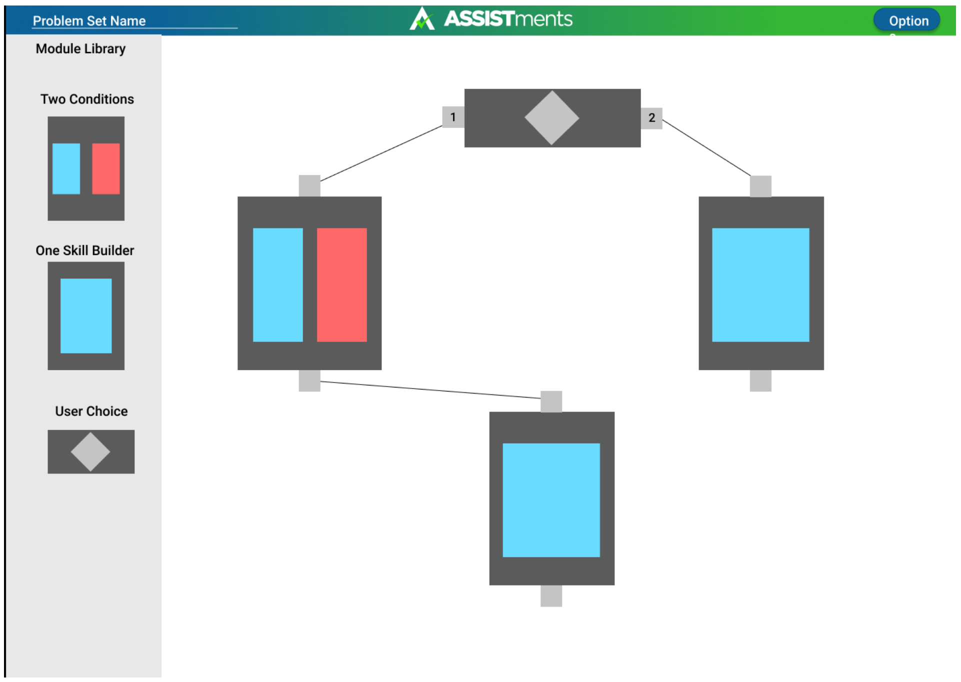
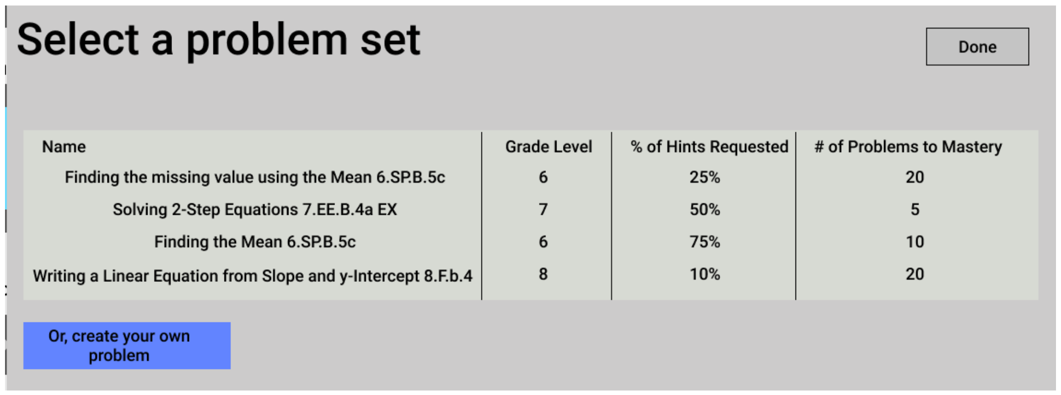


Figure 13: A simple study in Prototype A after a researcher has connected the modules they have selected

A researcher can populate each of these modules with problem content by clicking on them. For instance, when clicking on a One Skill Builder module, they are presented with a choice of problem sets to choose content from, in addition to statistics about each problem set, as shown in Figure 14.



Select a problem set

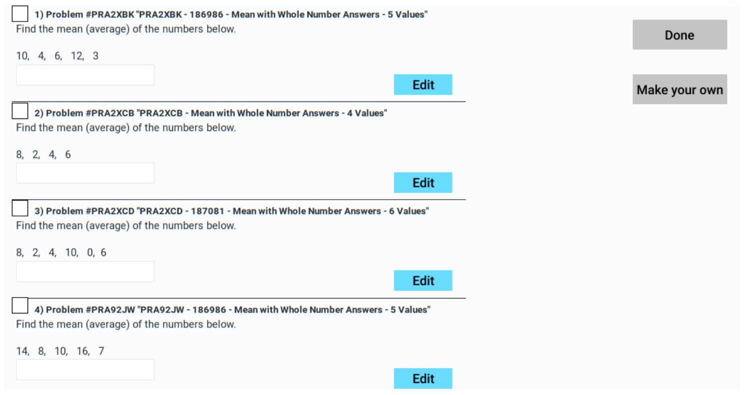
Done

Name	Grade Level	% of Hints Requested	# of Problems to Mastery
Finding the missing value using the Mean 6.SP.B.5c	6	25%	20
Solving 2-Step Equations 7.EE.B.4a EX	7	50%	5
Finding the Mean 6.SP.B.5c	6	75%	10
Writing a Linear Equation from Slope and y-Intercept 8.F.b.4	8	10%	20

Or, create your own problem

Figure 14: Selecting a problem set in Prototype A

Once a researcher has chosen a problem set, they can then select and edit existing problems or create new problems of their own, through the interface shown in Figure 15.


1) Problem #PRA2XBK "PRA2XBK - 186986 - Mean with Whole Number Answers - 5 Values"
Find the mean (average) of the numbers below.

10, 4, 6, 12, 3

2) Problem #PRA2XCB "PRA2XCB - Mean with Whole Number Answers - 4 Values"
Find the mean (average) of the numbers below.

8, 2, 4, 6

3) Problem #PRA2XCD "PRA2XCD - 187081 - Mean with Whole Number Answers - 6 Values"
Find the mean (average) of the numbers below.

8, 2, 4, 10, 0, 6

4) Problem #PRA92JW "PRA92JW - 186986 - Mean with Whole Number Answers - 5 Values"
Find the mean (average) of the numbers below.

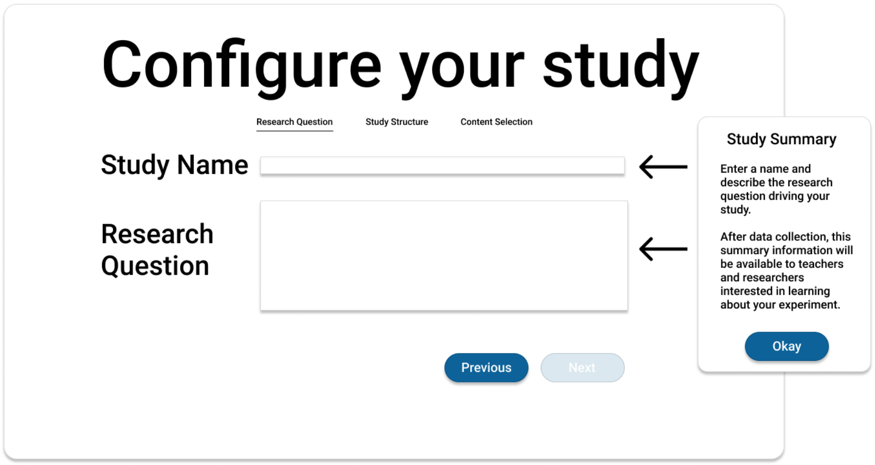
14, 8, 10, 16, 7

Figure 15: Selecting problems after selecting a problem set in Prototype A

This prototype provides certain benefits. The most notable benefit is that the researcher is welcome to make their structure as simple or complex as they want. However, this flexibility comes with a price: researchers are not guided towards their next step, and they may be left unsure of what to do next.

3.3.2 Prototype B: Condition-Based Building

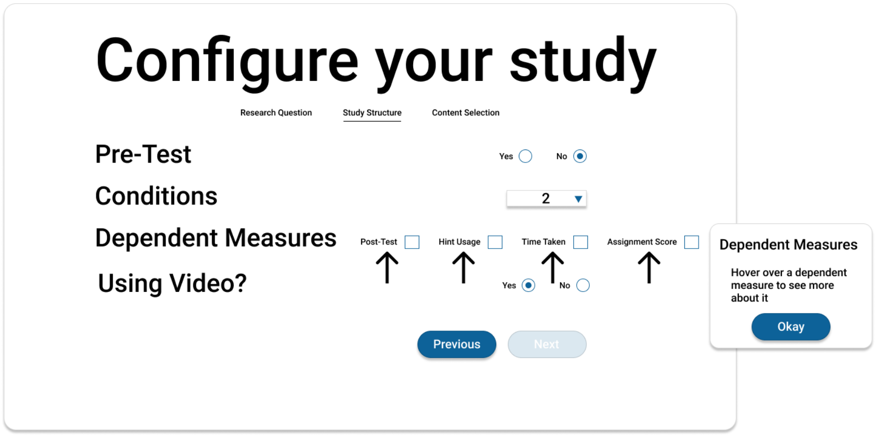
This prototype, designed by Mr. Emberling, aimed to provide an interface that gave researchers as much up-front configuration as possible. This included things as simple as naming their study and as complex as choosing what fields would represent dependent measures in the study.



Enter a name and describe the research question driving your study.

After data collection, this summary information will be available to teachers and researchers interested in learning about your experiment.

(a) Setting the study name and research question



Pre-Test Yes [✓ No]

Conditions 2

Using Video? [✓ Yes] No

Dependent Measures

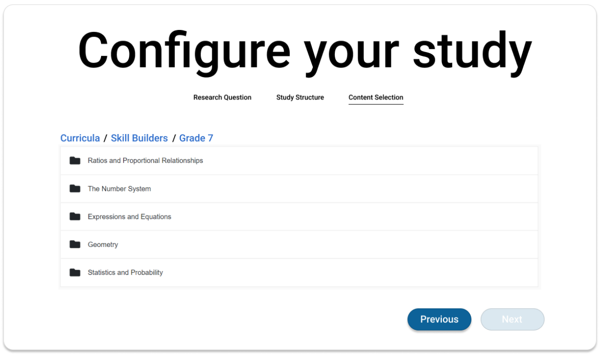
Hover over a dependent measure to see more about it

(b) Setting the structure and dependent measures

Figure 16: Various phases of study setup in Prototype B.

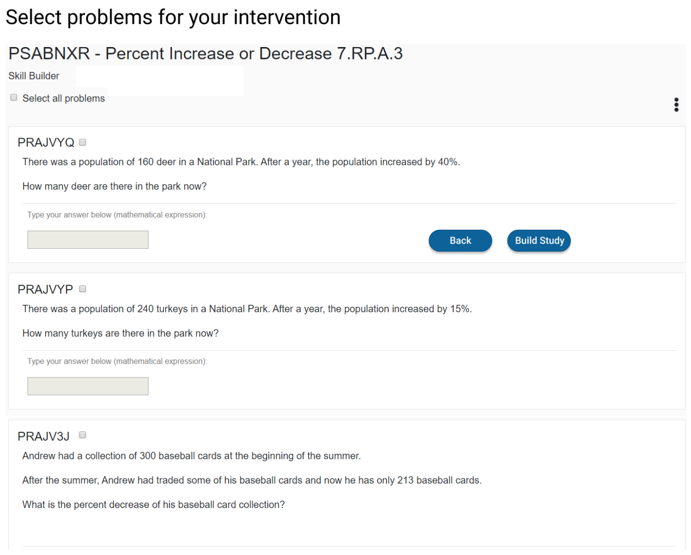
The concept of pre-configuration forces the researcher to make decisions about certain things up front. While this would be desirable for basic things like the study name and research questions, it was less clear what else should take place in such a process. One grey area was content selection. Prototype B chose to allow the researcher to select which set of problems they wished to use ahead of time.[4] Consider a case where a researcher wishes to develop a study that explores the difference between hints provided as text or as video. This requires creating at least one hint video for each problem, which can be quite labor intensive. As such, many researchers have historically chosen to only create content for a subset of problems, front-loading the study with this content and then filling the rest of the assignment with unaltered problems. Unaltered problems are those selected at this stage.

[4]Whether or not this was a good decision is addressed in Section 4.3.


Configure your study

Research Question
Study Structure
Content Selection
Curricula / Skill Builders / Grade 7
Ratios and Proportional Relationships
The Number System
Expressions and Equations
Geometry
Statistics and Probability
Previous
Next

(a) Selecting the problem set to use in the study


Select problems for your intervention

PSABNXR - Percent Increase or Decrease 7.RP.A.3

Skill Builder

PRAJVYQ

There was a population of 160 deer in a National Park. After a year, the population increased by 40%.

How many deer are there in the park now?

PRAJVYP

There was a population of 240 turkeys in a National Park. After a year, the population increased by 15%.

How many turkeys are there in the park now?

PRAJV3J

Andrew had a collection of 300 baseball cards at the beginning of the summer.

After the summer, Andrew had traded some of his baseball cards and now he has only 213 baseball cards.

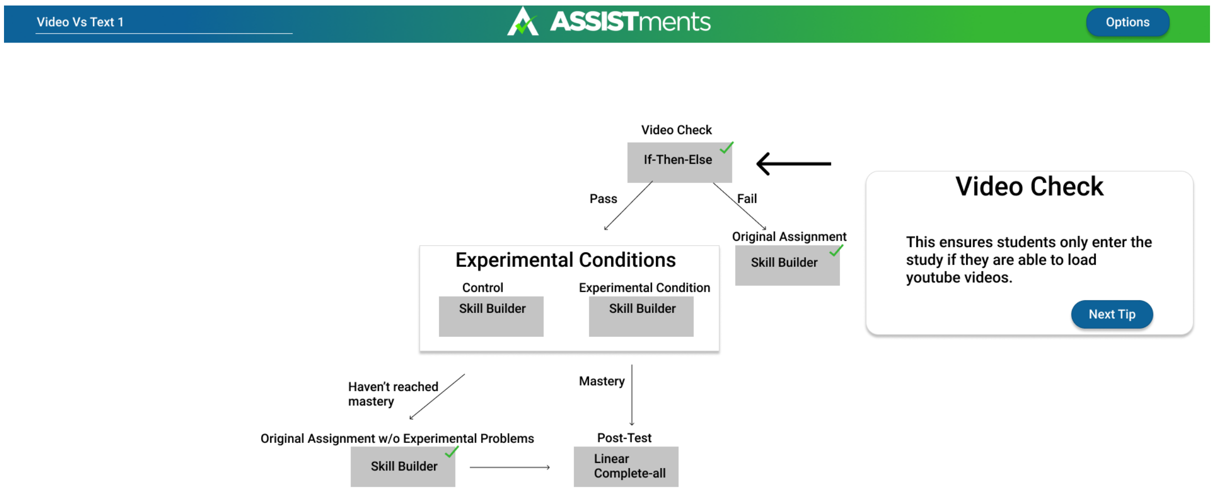
What is the percent decrease of his baseball card collection?

(b) Selecting which problems to include in the study

Figure 17: Upfront content selection in Prototype B

The other significant addition of this prototype was tooltips, which represent the embedded documentation mentioned in Section 2.2. Not only does this give researchers a rapid understanding of what they are building, but it ensures that documentation will not become stale, as it will be developed alongside the tool itself. See Figure 16 for an example of these tooltips.

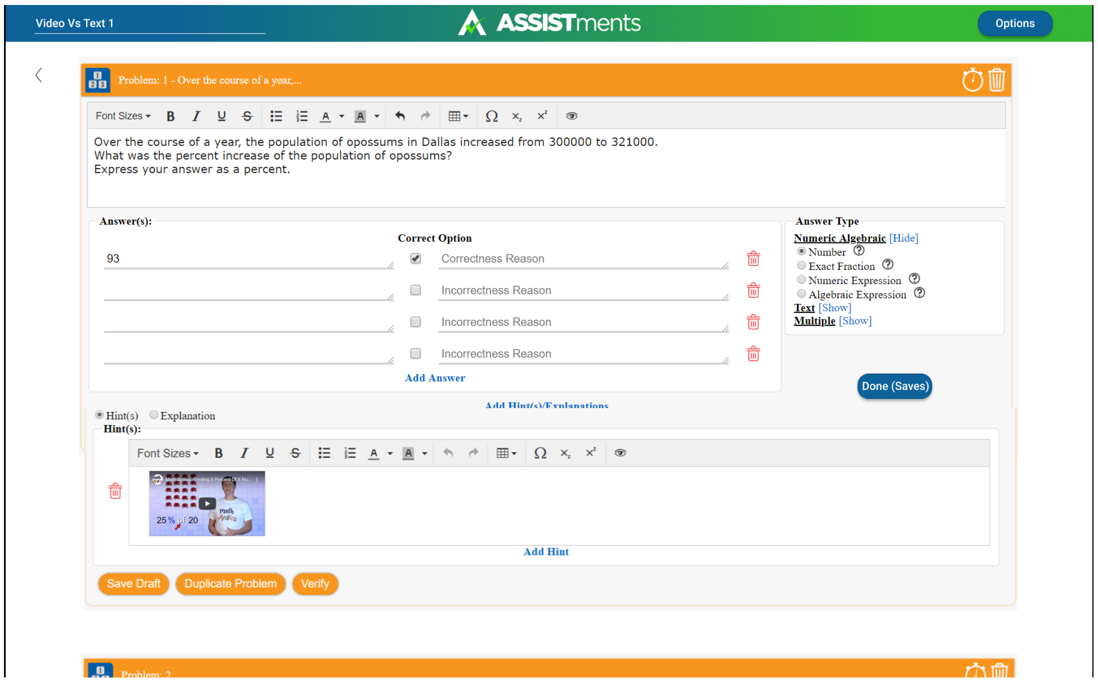
Once initial setup is complete, the researcher is shown a flowchart representation of their study, indicating which sections have been successfully configured. Each edge connecting modules in the flowchart clearly outlines potential cases in which student subjects will move through the study. Researchers are also shown more tooltips that provide an explanation of the resulting study structure; one such tooltip is shown in Figure 18.



This ensures students only enter the study if they are able to load youtube videos.

Figure 18: A flowchart produced after navigating through the setup phase of Prototype B

Clicking on one of the flowchart boxes immediately drops the researcher into a content editor for every problem contained within the condition. This allows the researcher to make modifications to the problems they have added as necessary (e.g. adding video hints to problems).



Over the course of a year, the population of opossums in Dallas increased from 300000 to 321000.
What was the percent increase of the population of opossums?
Express your answer as a percent.

Answer(s):

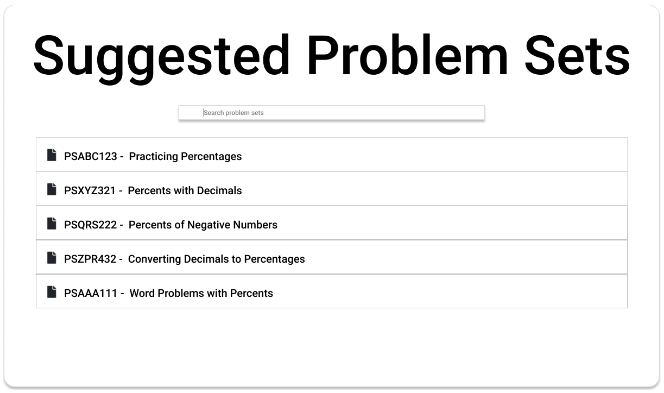
Answer	Correct Option	Reason
93	✓	Correctness Reason
		Incorrectness Reason
		Incorrectness Reason
		Incorrectness Reason

Answer Type
Numeric Algebraic [Hide]
● Number
○ Exact Fraction
○ Numeric Expression
○ Algebraic Expression
Text [Show]
Multiple [Show]

● Hint(s) ○ Explanation
Hint(s):

Figure 19: Editing the existing content for a study created in Prototype B

When modifying the post-test specifically, the researcher is instructed to select a related problem set (which is often considered good practice when building a study[5]).

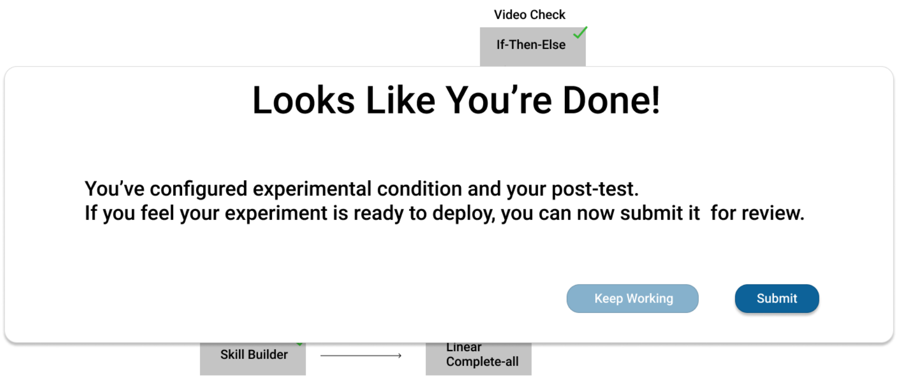


Suggested Problem Sets

- PSABC123 - Practicing Percentages
- PSXYZ321 - Percents with Decimals
- PSQRS222 - Percents of Negative Numbers
- PSZPR432 - Converting Decimals to Percentages
- PSAAA111 - Word Problems with Percents

Figure 20: Selection of related problem sets for the post test in Prototype B

When a researcher has completed all phases of setup, they are notified that the process is complete. From here, they may either make further customizations, or they may submit their study to be reviewed by the ASSISTments research team.



Looks Like You're Done!

You've configured experimental condition and your post-test.
If you feel your experiment is ready to deploy, you can now submit it for review.

Figure 21: An indication that a researcher has completed their study in Prototype B

3.3.3 Prototype C: Customizable Templates

Mr. Spinelli opted for a template-driven builder prototype that rested on a middle ground between flexibility (such as Prototype A) and an on-rails experience (such as Prototype B). This was a continuation of the same idea behind the already-functional Template Tool. The main idea of this design was to use a template-based system where each template had configurable options to allow for customization. The available options could differ from template to template.

The initial state of Prototype C was the "My Experiments" page shown in Figure 22. This was a dashboard for all studies created by a researcher; from here, a researcher could conceivably view the status of a running study or make edits to a non-running study. This idea was not the main focus of the prototype and was not fully fleshed out, but it was unique among the four prototypes.

[5]To be specific, it is a good demonstration of mastery when a student can apply their knowledge in the form of a near or far transfer skill, or a related extension to the assigned domain skill. For instance, if an assignment involves adding exponents, performing operations with scientific notation may be a viable choice for a transfer based post-test.

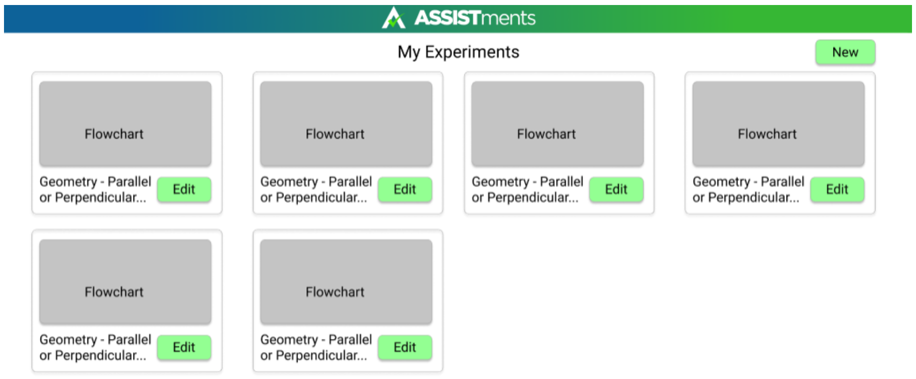
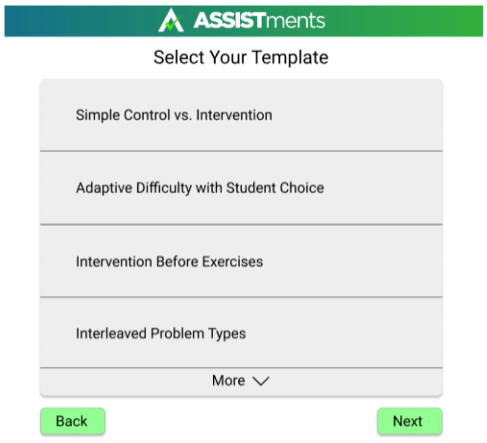
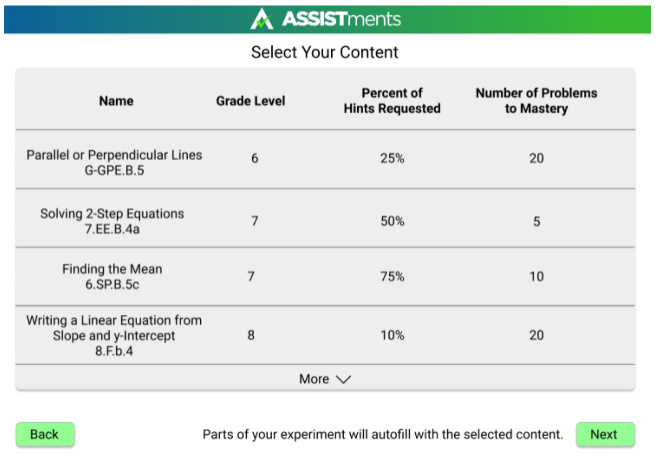


Figure 22: The “My Experiments” page in Prototype C

When the researcher clicks the “New” button, they must select a template and a problem set to use in their study. Content selection in this prototype is simpler than in Prototype B: the researcher only selects a full problem set, not individual problems.


ASSISTments
Select Your Template
Simple Control vs. Intervention
Adaptive Difficulty with Student Choice
Intervention Before Exercises
Interleaved Problem Types
More
Back
Next

(a) Template selection in Prototype C



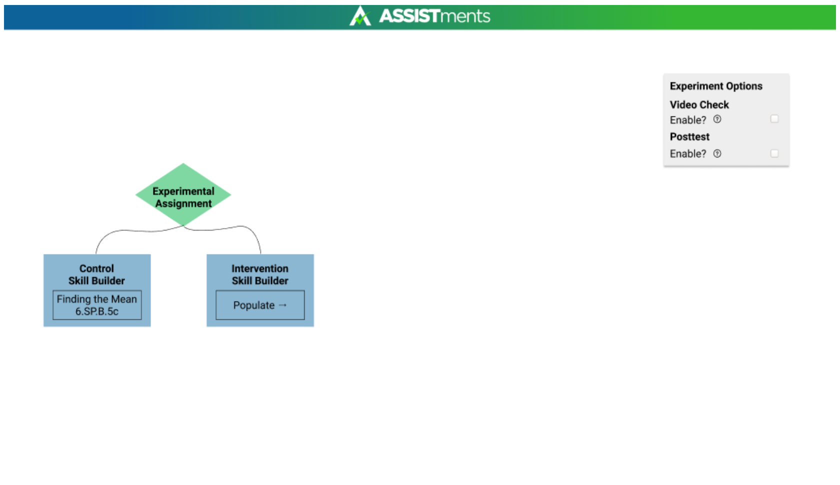
Name	Grade Level	Percent of Hints Requested	Number of Problems to Mastery
Parallel or Perpendicular Lines G-GPE.B.5	6	25%	20
Solving 2-Step Equations 7.EE.B.4a	7	50%	5
Finding the Mean 6.SP.B.5c	7	75%	10
Writing a Linear Equation from Slope and y-Intercept 8.F.b.4	8	10%	20

Parts of your experiment will autofill with the selected content.

(b) Content selection in Prototype C

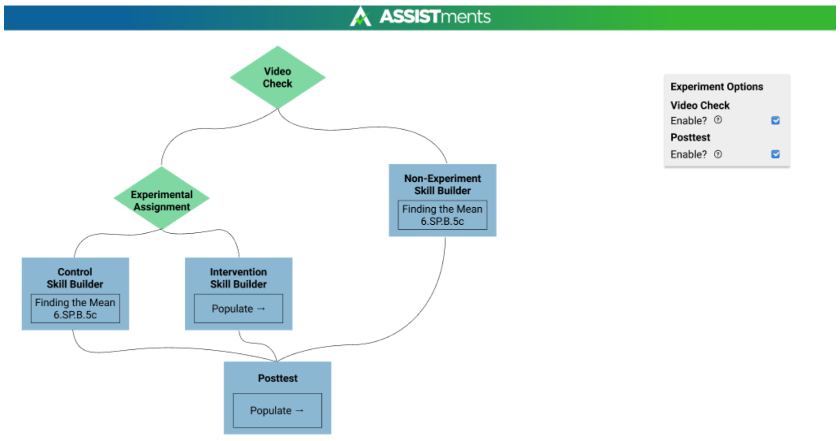
Figure 23: Introductory experiment configuration in Prototype C

After this, researchers enter the main overview section where they can view their template structure and configure certain details in the “Experiment Options” box. The experiment structure is represented as a top-to-bottom flowchart, and components are color-coded according to their type. In the example in Figure 24, the researcher can enable and disable the video check and the post-test independently and the structure will update accordingly. This prototype also features tooltips that explain the functionality of each toggle.



Experiment Options
Video Check
Enable? ☐
Posttest
Enable? ☐

(a) Toggleable options disabled

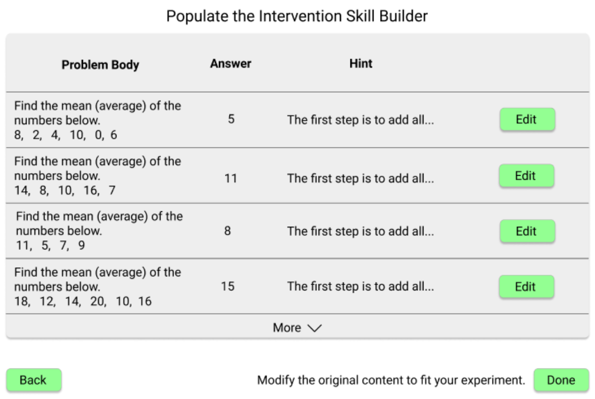

ASSISTments
Video Check
Experimental Assignment
Non-Experiment Skill Builder
Finding the Mean 6.SP.B.5c
Control Skill Builder
Finding the Mean 6.SP.B.5c
Intervention Skill Builder
Populate →
Posttest
Populate →

Experiment Options
Video Check
Enable? ☑
Posttest
Enable? ☑

(b) Toggleable options enabled

Figure 24: Toggleable options in Prototype C

When the researcher clicks "Populate →" on the Intervention Skill Builder or the Posttest, they can enter content for those components. Since the problems in the Intervention Skill Builder should be a derivative of the content in the Control Skill Builder, the researcher is encouraged to edit the original problems to populate the Intervention Skill Builder. Here, and elsewhere in the design, short instructional taglines are included near the bottom of the viewport to guide the researcher.



Populate the Intervention Skill Builder

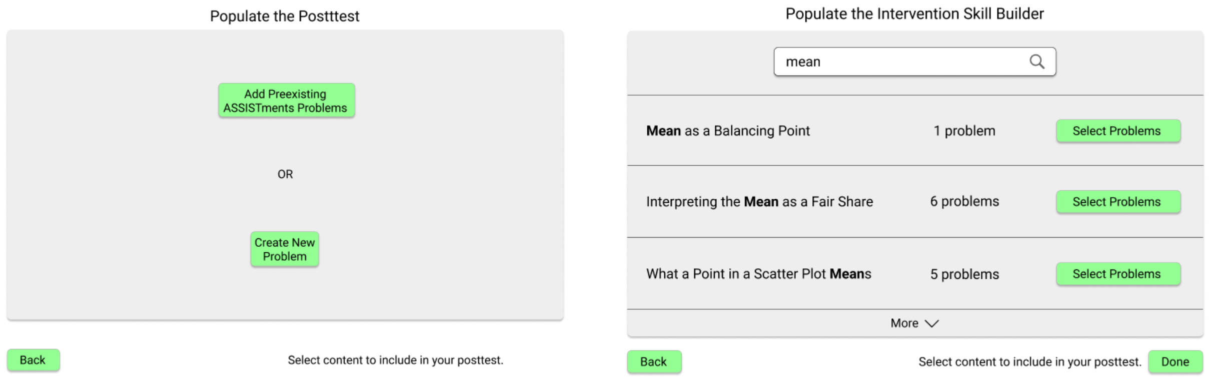
Problem Body	Answer	Hint	
Find the mean (average) of the numbers below. 8, 2, 4, 10, 0, 6	5	The first step is to add all...	Edit
Find the mean (average) of the numbers below. 14, 8, 10, 16, 7	11	The first step is to add all...	Edit
Find the mean (average) of the numbers below. 11, 5, 7, 9	8	The first step is to add all...	Edit
Find the mean (average) of the numbers below. 18, 12, 14, 20, 10, 16	15	The first step is to add all...	Edit

Modify the original content to fit your experiment.

Figure 25: Populating the Intervention Skill Builder in Prototype C

In this prototype, researchers are encouraged to populate the post-test with either problems from a related problem set, or to create new problems entirely, as shown in Figures 28 and 29.

The flowchart view updates automatically after populating a component to show that the new content has been added, as shown in Figure 28.



Select content to include in your posttest.

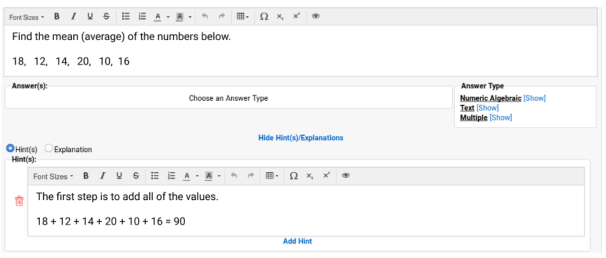
Populate the Intervention Skill Builder

Mean as a Balancing Point	1 problem	Select Problems
Interpreting the **Mean** as a Fair Share	6 problems	Select Problems
What a Point in a Scatter Plot **Mean**s	5 problems	Select Problems

Select content to include in your posttest.

(a) The two options for post-test population in Prototype C (b) Searching for related problem sets in the post-test of Prototype C

Figure 26: The process of populating a post-test in Prototype C



Find the mean (average) of the numbers below.

18, 12, 14, 20, 10, 16

Answer(s):

Choose an Answer Type

Answer Type

Numeric Algebraic [Show]
Text [Show]
Multiple [Show]

[✓ Hint(s)] Explanation

Hint(s):

The first step is to add all of the values.

18 + 12 + 14 + 20 + 10 + 16 = 90

Figure 27: Adding a new problem in Prototype C

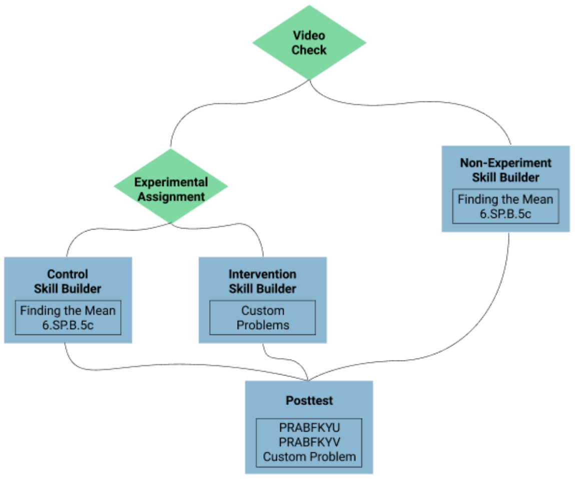


Figure 28: The flowchart in Prototype C after both the Intervention Skill Builder and the Post-test have been populated

The template-driven control flow enables the possibility of detecting when a user is finished building, similar to that observed in Prototype B.

3.3.4 Prototype D: Interactive Structure Building

While other prototypes place experimental structure and content selection on equal footing, Dr. Ostrow's design was driven primarily by structure. This prototype had researchers fully specify their experimental flow up front before moving on to content selection once the structure had been defined. This likely parallels the way a researcher would decide on experimental structure and material in general; our user research revealed that most researchers are looking to test the efficacy of a certain intervention rather than work with specific subject matter.

This prototype hinted at the existence of a control panel from which a researcher could access all of their experiments, as observed in Prototype C. ALI reports also appear to be integrated into the software, eliminating the previously-existing fragmentation between experiment building and data retrieval.

Figure 29: Links to the "My Experiments" control panel and "My Data Reports" ALI area in Prototype D

In Prototype D, a live preview is updated whenever the user configures an option for their experiment, much like that observed in Prototype C. However, Prototype D employed a different design language than that used by the other prototypes, opting instead to portray options and preview as equally-important panels within the software, as shown in Figure 31.

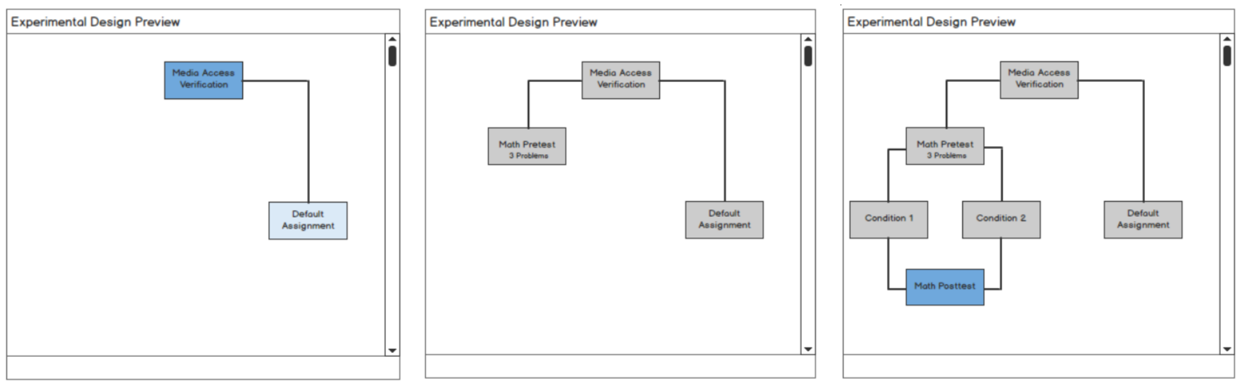
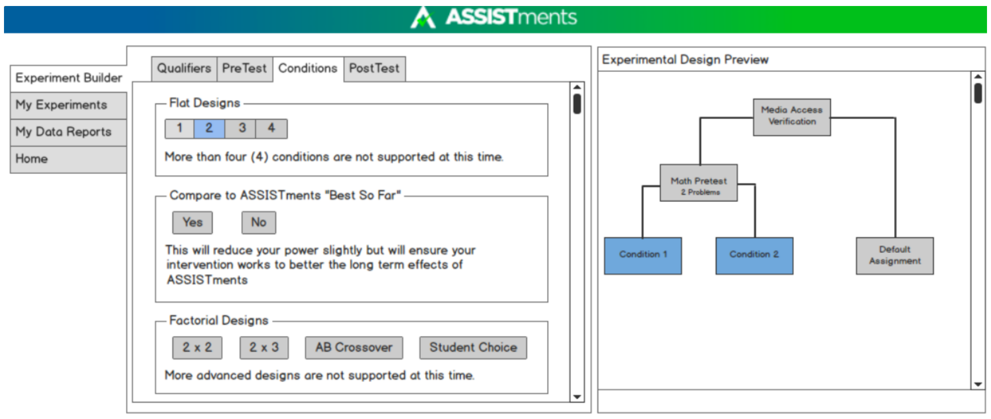

Experimental Design Preview
Media Access Verification
Default Assignment
Experimental Design Preview
Media Access Verification
Math Pretest 3 Problems
Default Assignment
Experimental Design Preview
Media Access Verification
Math Pretest 3 Problems
Condition 1
Condition 2
Default Assignment
Math Posttest

Figure 30: Samples of the live structure preview in Prototype D



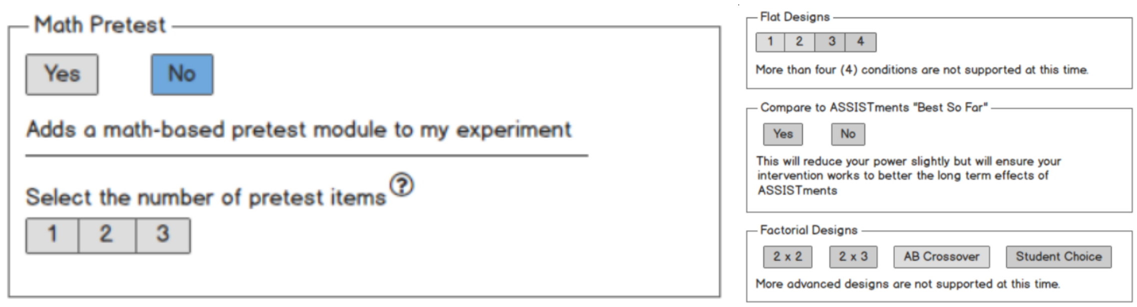
More than four (4) conditions are not supported at this time.

This will reduce your power slightly but will ensure your intervention works to better the long term effects of ASSISTments

More advanced designs are not supported at this time.

Figure 31: The coexisting configuration and preview panels in Prototype D

Prototype D also kept analysis as a first-class concern; the use of a pretest is encouraged, which can be used to normalize post-test scores, and the configuration of experimental conditions allows for complex yet statistically insightful study construction, as shown in Figure 32.



Adds a math-based pretest module to my experiment

More than four (4) conditions are not supported at this time.

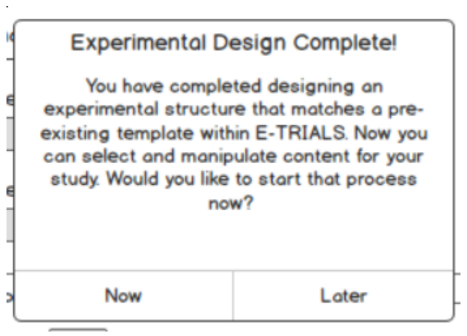
This will reduce your power slightly but will ensure your intervention works to better the long term effects of ASSISTments

More advanced designs are not supported at this time.

Figure 32: Examples of data analysis-relevant features in Prototype D

As observed in Prototypes B and C, the linearized control flow allows the software to detect when a researcher appears to be "done" creating an experimental structure. When this state is detected, the software encourages the user to move on to content selection.

Although Prototype D hints at content selection as part of the envisioned study builder, it did not include any concrete details on the process at the time of writing.



Experimental Design Complete!

You have completed designing an experimental structure that matches a pre-existing template within E-TRIALS. Now you can select and manipulate content for your study. Would you like to start that process now?

Figure 33: The "done" message in Prototype D

The usage of brief descriptions and tooltips throughout the wizard, together with the linear process suggested by the tabs or pages for configuration, help to hold the user's hand and consistently guide them toward a completed experiment. Dr. Ostrow also attempted to abandon ASSISTments' jargon. For example, the ubiquitous term "video check" is supplanted by a more descriptive term, "media access verification."

Figure 34: The navigation tabs in Prototype D

3.4 User Testing

We conducted two user testing sessions internal to the ASSISTments organization: one among the members of our IQP team and one with Mr. Thanaporn "March" Patikorn. In each case, the test subject completed a task in all four prototypes. This aided us in identifying the strengths and weaknesses of each design, and in enumerating characteristics and features of the 2.0 builder. In each user test, the subject was given the following instructions on the type of experiment they should build using the design prototype:

> Create an experiment to answer the following research question: Are video hints or text hints better for learning? Design your experiment to satisfy the following requirements:
>
> a. Use a problem set that has a high percentage of hint usage and for which students (on average) require fewer than 15 problems to reach mastery as the basis for your experiment.
> b. Use a post-test with three questions in it. Select the first two questions from a related problem set and create the third question from scratch.

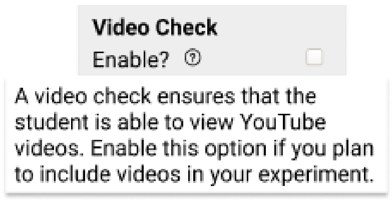
These instructions ensured that the test subject pays attention to several facets of the study building process: navigation of the prototype, structuring, content selection, content creation, and data analysis.

After each test subject completed the task using each prototype, we asked them for comments on the designs. In particular, we encouraged the test subjects to articulate any shortcomings, confusions, or frustrations they experienced. While test subjects may be hesitant to criticize, working with familiar colleagues in early iterations can help to mitigate such inhibitions. The findings from this testing is discussed in Section 4.

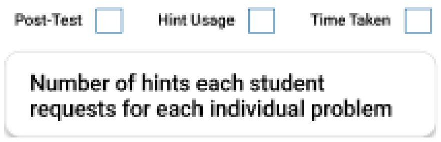
4 Proposals

4.1 Documentation

User testing indicated the efficacy of embedded documentation, in direct contrast to the fragmented external documentation currently available for the 1.0 builder. However, the exact form this should take is not well-defined, and is superfluous to the goal of simply incorporating some type of embedded documentation. Some of the prototypes described herein included various form of tooltips, as shown in Figure 35. One prototype included brief descriptions alongside many of the experiment settings, as shown in Figure 36.


Video Check
Enable?
A video check ensures that the student is able to view YouTube videos. Enable this option if you plan to include videos in your experiment.

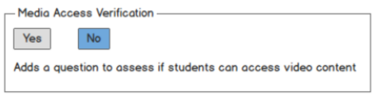
(a) A tooltip in Prototype C



Number of hints each student requests for each individual problem

(b) A tooltip in Prototype B.

Figure 35: Samples of tooltips in prototypes



Adds a question to assess if students can access video content

Figure 36: An option description in Prototype D

With that said, external documentation should be avoided if possible. Much of our confusion during the onboarding process was the result of a lack of unified documentation. As the ASSISTments universe grows and evolves, documentation will need to be updated. In the past, maintaining several repositories of information has proven ineffective and we do not want to recreate the same issue.

We also recognized the utility of a built-in tutorial system. Mr. Patikorn said during his testing session, "personally I prefer a lot of handholding when it comes to building experiments." We agree: part of the difficulty in using the 1.0 builder is its imbalanced ratio of power to intuitiveness. A built-in tutorial would help new users get going quicker and could support returning users in learning more advanced techniques in the ASSISTments ecosystem.

4.2 Visualization

As previously mentioned, a flowchart representation is the most intuitive and natural visualization method we found for visualizing experiments. A flowchart mimics the way that a researcher thinks about a study's structure, and when working inside the framework of a template-based builder (see section 4.4) it is possible to abstract away the internal software representation of experimental flow using such a visualization. Using a flowchart-based visualization scheme also helps to hold new users' hands and put them on familiar ground as they work.

4.3 Content Selection

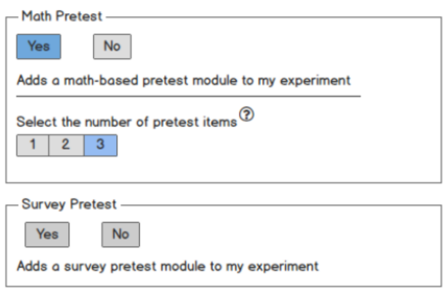
Content selection was a challenge for us throughout our prototyping phases. In all cases though, we feel that moving away from problem and problem set IDs is incredibly important. These details pertain strictly to the backend of how ASSISTments implements content, and don't need to be exposed to researchers. In addition, all details pertaining to the problem set should be embedded directly into the tool, just as with the documentation; researchers should not have to go to an external tool to find information such as average accuracy rates. With that said, the specific interface for finding problems was not explored fully in our prototypes. We were able to determine that some kind of searching and sorting of problem sets would be necessary, but we did not find a design that all team members approved of during prototyping.

The question was also raised where a researcher should select their problems. On one hand, being able to choose the questions before the researcher starts building allows them to hit the ground running without having to worry about padding their study with extra problems. For instance, if a researcher made a study that involved implementing video content within each question and could only produce a small number of videos, they would not have to worry about adding extra problems to make the assignment long enough for struggling students to reach mastery. On the other hand, a researcher may be more concerned with the structure of their study than picking a specific domain or topic, as their research may not be dependent on the content assigned. This gives researchers more control, but they may not pad the problem sets as necessary. The team has no specific proposal regarding this at this time, but it is worth considering the pros and cons of each choice.

4.4 Structure and Descriptive Power

A template-based builder strikes a good balance between flexibility and foolproofing. The key is to curate a large number of supported templates that are applicable to many scenarios and flexible enough to accommodate most needs. If these templates are well-designed and clear to the user, most researchers should need less help from ASSISTments staff than before.

One way to increase the flexibility of templates is to include toggleable options inside of a template. Some prototypes implemented these options, which elicited positive feedback in user tests.


Math Pretest

[✓ Yes] No

Adds a math-based pretest module to my experiment

Select the number of pretest items

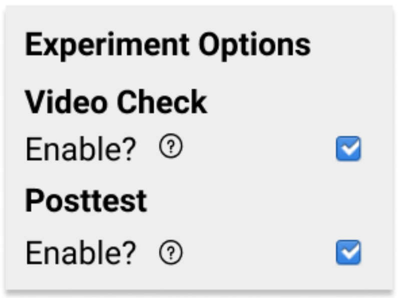
1 2 [✓ 3]

Survey Pretest

Yes No

Adds a survey pretest module to my experiment

(a) Toggleable pretest settings in Prototype D


Experiment Options

Video Check

Enable? ✓

Posttest

Enable? ✓

(b) Video check and posttest settings in Prototype C

Figure 37: Samples of toggleable options in prototypes

In user testing, we found that this level of flexibility was much more accessible than a model where researchers had to build the full study themselves, such as that applied by Prototype A. Allowing extensive control often left the user feeling overwhelmed and unsure of what to do next, while small toggleable options permitted testers to make informed decisions of what changes to make in their study.

4.5 External Support and Troubleshooting

The argument to prioritize reducing the need for external support is primarily based on efficiency. Making the need for one-on-one support largely obsolete will save the ASSISTments organization time and money. This allows employees to focus their efforts on improving the platform and studying best practices in online learning, which generates lasting value, in contrast to the meager benefits of helping researchers build individual studies.

The inclusion of a possible tutorial system contributes to creating a community of well-informed researchers who come in with the basic knowledge of study-building. From that point onward, embedded documentation empowers researchers to independently resolve any issues they may experience.

A template-based system also makes it more difficult for a researcher to sabotage themselves than it was in the 1.0 builder, simply by virtue of a smaller space of possible studies that can be built. By constraining researchers to a well-defined set of studies (namely those that align with templates) we can foresee and plan for possible errors.

There is also value in knowing when a researcher is "done" creating their study because experiment deployment could conceivably be integrated into the builder. An optimal scenario would entirely remove the need for communication between a researcher and the ASSISTments research team during the deployment process. Researchers could submit their experiment for review via the builder, which would automatically notify an ASSISTments administrator. The administrator would log into their dashboard and then either approve the study and publish it, or deny the request and send it back to the researcher for revision.

4.6 Data Analysis

Though data analysis was not our focus, it was important for our prototypes to recognize that in order for automated analysis to occur, a dependent measure must be identified. Unless ALI has some way of identifying what should be measured, the researcher is going to be forced to dive into the raw data, which is the same situation we found ourselves in before. In order to determine a path forward, the team engaged in conversations with Ms. Taylor Stefovic, a graduate student on the ASSISTments research team, Dr. Anthony Botelho, a Research Scientist at ASSISTments, and Mr. Joseph St. Pierre, a software engineer at ASSISTments. Through these conversations, we solidified a plan for how this analysis should be integrated into the front-end design of the application, as reflected in the final prototype in Section 5.

4.7 ASSISTments Team Intervention

In some cases, a study may be too complex for a template-based study builder. In these cases, we highly suggest ASSISTments have some kind of back door to fix complexities in studies. This could be accomplished with some of the more advanced features of the content builder put together by Mr. Gurung, which would allow for manual adjustmentment of a study's structure.

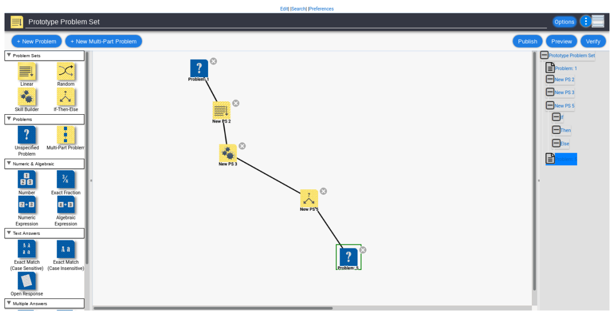
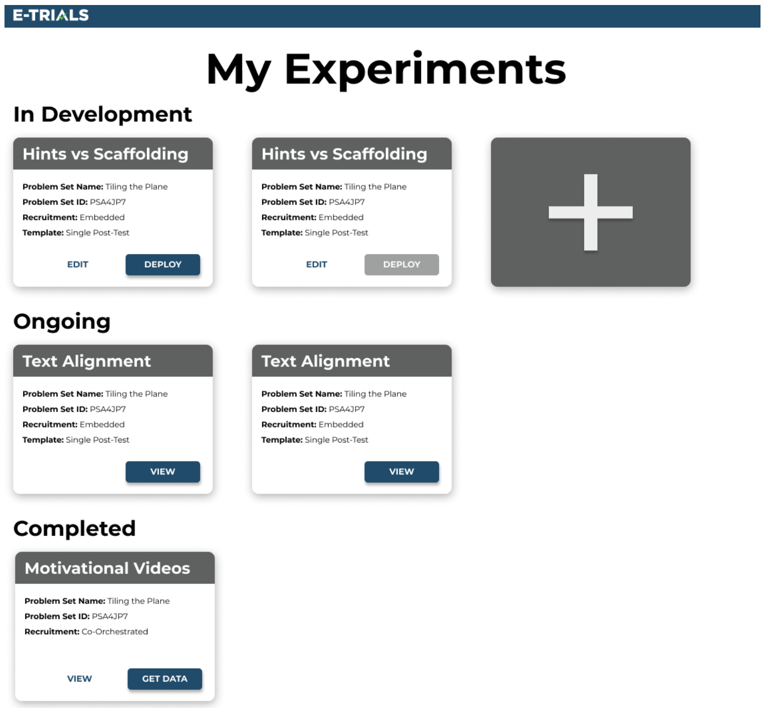


Figure 38: An example of utilizing the content builder to manually configure a study

5 Unifying the Prototypes

This IQP included a gap term for scheduling reasons. During this period, Dr. Ostrow and Mr. Emberling combined the four prototypes presented herein into a final unified prototype, following a final joint exercise before our reprieve that mapped the similarities and differences of the four designs, as shown in Appendix B. This prototype then became the basis for the first iteration of the coded E-TRIALS application.

When a researcher first loads the prototype, they are presented with the page shown in Figure 39. From this page, users can create a new study, edit existing work, or deploy a prepared study to student users of ASSISTments.


E-TRIALS

My Experiments

In Development

Hints vs Scaffolding
Problem Set Name: Tiling the Plane
Problem Set ID: PSA4JP7
Recruitment: Embedded
Template: Single Post-Test
EDIT DEPLOY

Hints vs Scaffolding
Problem Set Name: Tiling the Plane
Problem Set ID: PSA4JP7
Recruitment: Embedded
Template: Single Post-Test
EDIT DEPLOY

+

Ongoing

Text Alignment
Problem Set Name: Tiling the Plane
Problem Set ID: PSA4JP7
Recruitment: Embedded
Template: Single Post-Test
VIEW

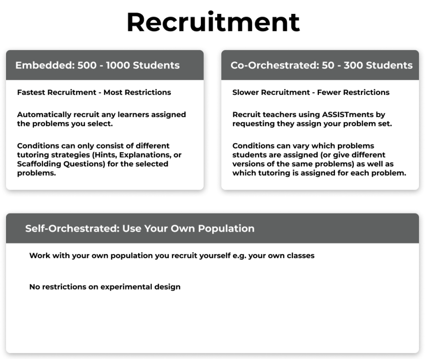
Text Alignment
Problem Set Name: Tiling the Plane
Problem Set ID: PSA4JP7
Recruitment: Embedded
Template: Single Post-Test
VIEW

Completed

Motivational Videos
Problem Set Name: Tiling the Plane
Problem Set ID: PSA4JP7
Recruitment: Co-Orchestrated
VIEW GET DATA

Figure 39: The front page of the final prototype

Upon beginning the setup process, researchers are presented with options to choose their study paradigm and how they wish to recruit students, as seen in Figure 40. This allows the researcher to explore available options and moves a previously human-communicated (and thereby time consuming) aspect of study development into the application.

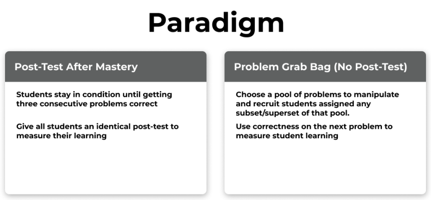

Recruitment

Embedded: 500 - 1000 Students
Fastest Recruitment - Most Restrictions
Automatically recruit any learners assigned the problems you select.
Conditions can only consist of different tutoring strategies (Hints, Explanations, or Scaffolding Questions) for the selected problems.

Co-Orchestrated: 50 - 300 Students
Slower Recruitment - Fewer Restrictions
Recruit teachers using ASSISTments by requesting they assign your problem set.
Conditions can vary which problems students are assigned (or give different versions of the same problems) as well as which tutoring is assigned for each problem.

Self-Orchestrated: Use Your Own Population
Work with your own population you recruit yourself e.g. your own classes
No restrictions on experimental design

(a) The recruitment page within the final prototype

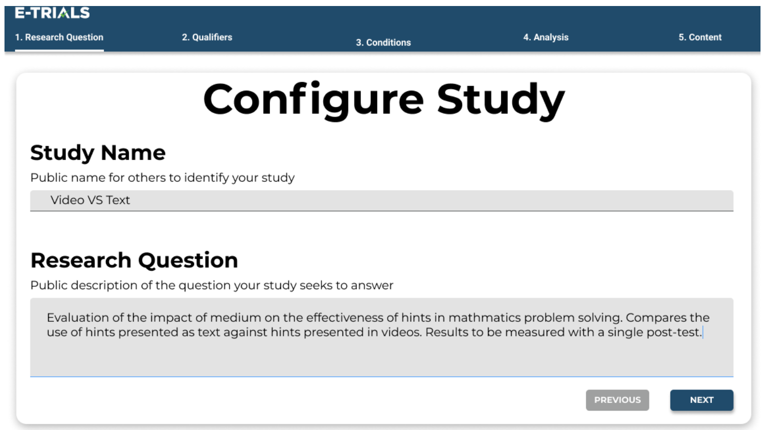

Paradigm

Post-Test After Mastery
Students stay in condition until getting three consecutive problems correct
Give all students an identical post-test to measure their learning

Problem Grab Bag (No Post-Test)
Choose a pool of problems to manipulate and recruit students assigned any subset/superset of that pool.
Use correctness on the next problem to measure student learning

(b) The paradigm page within the final prototype

Figure 40: Selection of study recruitment and paradigm pages within the final prototype



Configure Study

Study Name

Public name for others to identify your study

Video VS Text

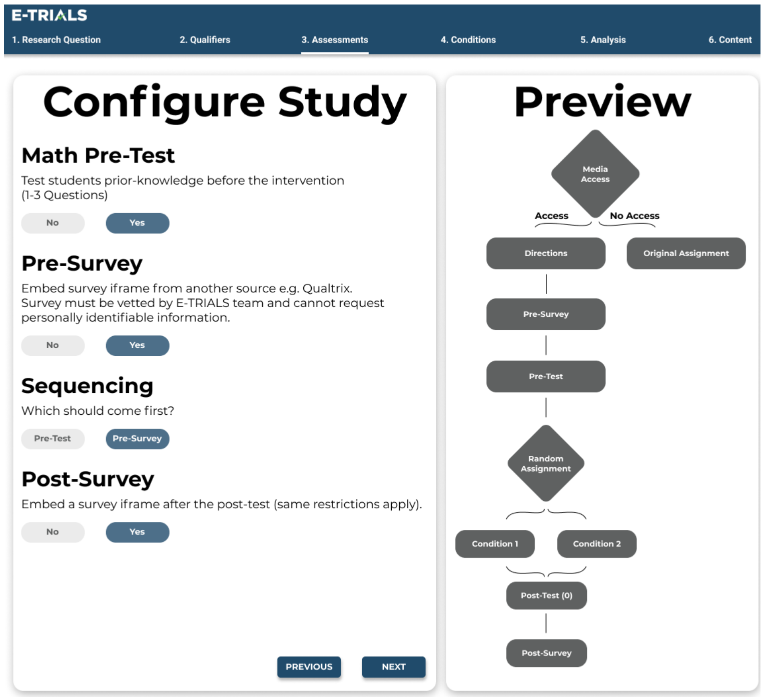
Research Question

Public description of the question your study seeks to answer

Evaluation of the impact of medium on the effectiveness of hints in mathmatics problem solving. Compares the use of hints presented as text against hints presented in videos. Results to be measured with a single post-test.

Figure 41: Setup of basic study details in the final prototype

After configuring the basic details of a study, such as the name and research question (see Figure 41), the researcher then has the ability to create their conditions interactively (see Figure 42), as previously observed in Prototypes B and D. This design was used in lieu of the template-oriented approach of Prototype B; this should give researchers enough flexibility to easily and intuitively recreate the complex structures afforded by each template.



Configure Study

Math Pre-Test

Test students prior-knowledge before the intervention (1-3 Questions)

No [✓ Yes]

Pre-Survey

Embed survey iframe from another source e.g. Qualtrix. Survey must be vetted by E-TRIALS team and cannot request personally identifiable information.

No [✓ Yes]

Sequencing

Which should come first?

Pre-Test [✓ Pre-Survey]

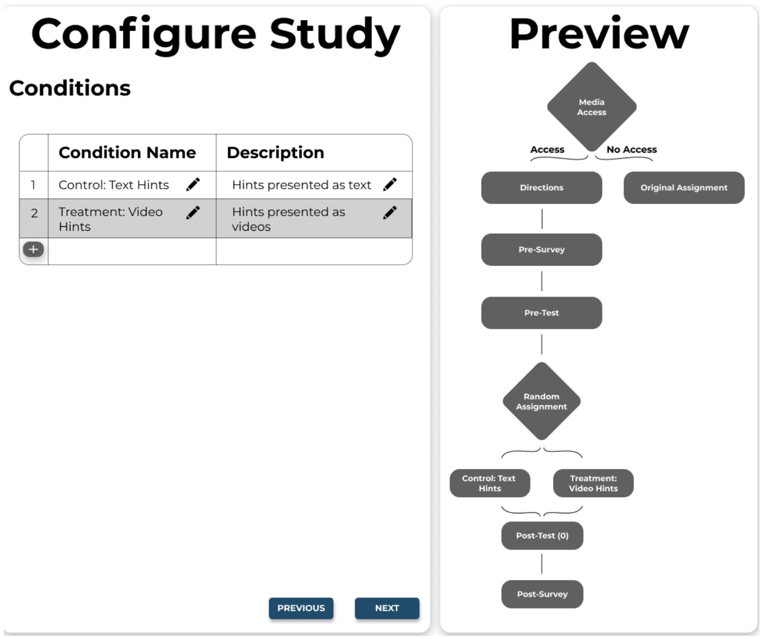
Post-Survey

Embed a survey iframe after the post-test (same restrictions apply).

No [✓ Yes]

Preview

(a) Setup of study components in the final prototype



Configure Study

Conditions

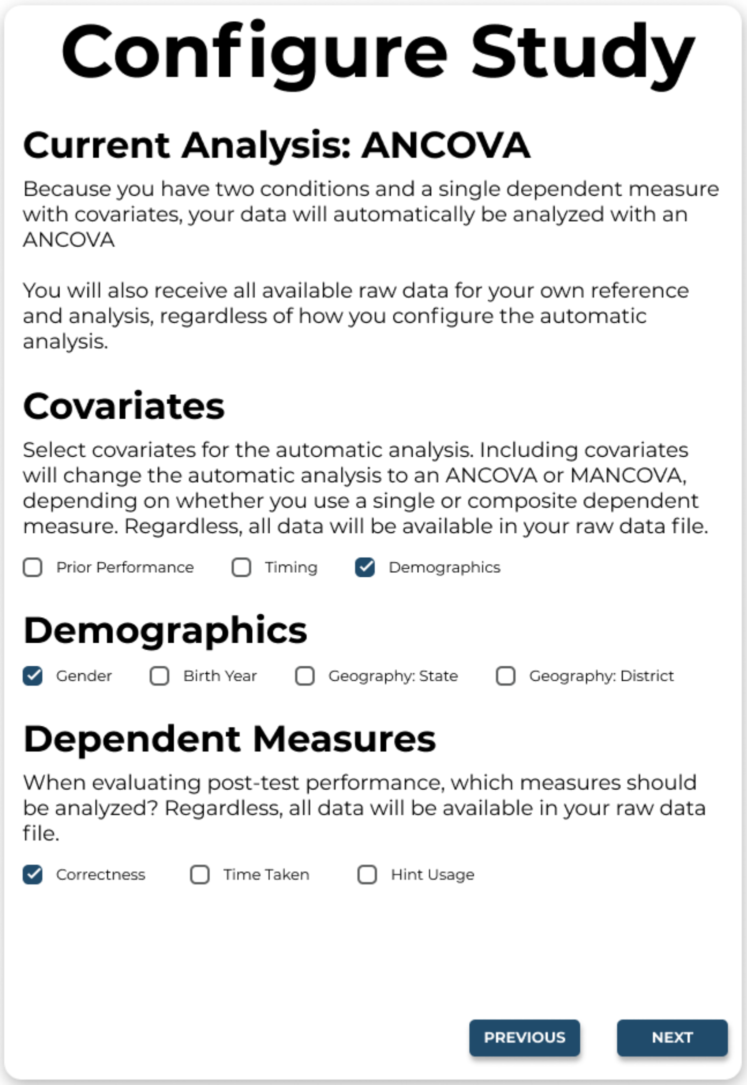
	Condition Name	Description
1	Control: Text Hints	Hints presented as text
2	Treatment: Video Hints	Hints presented as videos
+		

Preview

(b) Adding conditions to the study and subsequent flowchart preview in the final prototype

Figure 42: Configuration of study structure in the final prototype

Before adding content, the researcher must configure their desired analytic settings. This not only includes selecting the dependent measure, but also any covariates that many influence study results, such as demographics or prior performance on assignments. This step is intended to help guide researchers in preparing preregistration for their study while simultaneously providing E-TRIALS with information for backend analytics that will drive content improvement in ASSISTments.



Configure Study

Current Analysis: ANCOVA

Because you have two conditions and a single dependent measure with covariates, your data will automatically be analyzed with an ANCOVA

You will also receive all available raw data for your own reference and analysis, regardless of how you configure the automatic analysis.

Covariates

Select covariates for the automatic analysis. Including covariates will change the automatic analysis to an ANCOVA or MANCOVA, depending on whether you use a single or composite dependent measure. Regardless, all data will be available in your raw data file.

☐ Prior Performance ☐ Timing ☑ Demographics

Demographics

☑ Gender ☐ Birth Year ☐ Geography: State ☐ Geography: District

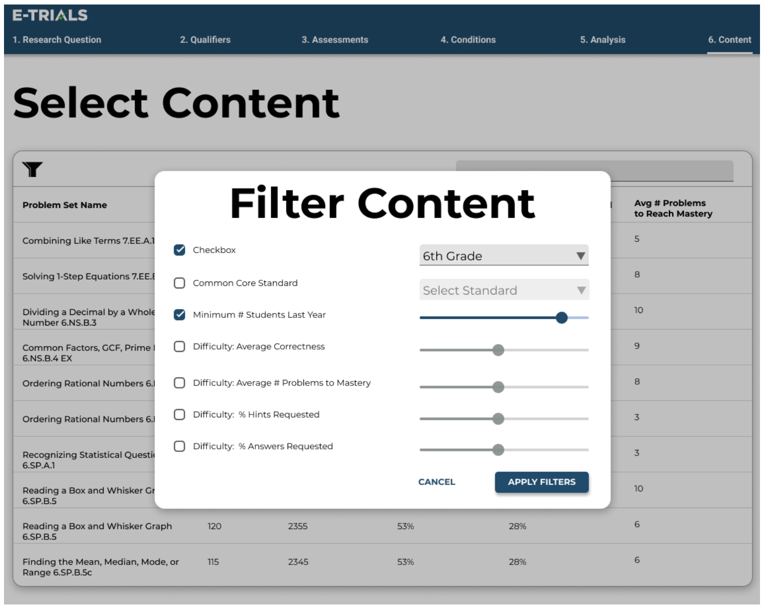
Dependent Measures

When evaluating post-test performance, which measures should be analyzed? Regardless, all data will be available in your raw data file.

☑ Correctness ☐ Time Taken ☐ Hint Usage

Figure 43: Selecting analysis techniques in the final prototype

Finally, the researcher can begin selecting content. In order to avoid having to refer to large spreadsheets, such as those mentioned in Section 2.2, the prototype integrates the list of problem sets and allows the user to filter the selection to a more manageable subset. This can include searching by grade level, average problem correctness, or average problems to mastery.



Select Content

Filter Content

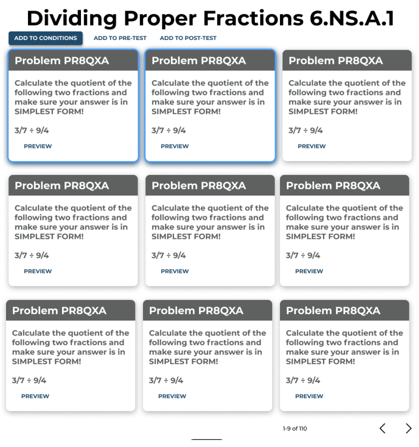
- [✓] Checkbox — 6th Grade
- [ ] Common Core Standard — Select Standard
- [✓] Minimum # Students Last Year
- [ ] Difficulty: Average Correctness
- [ ] Difficulty: Average # Problems to Mastery
- [ ] Difficulty: % Hints Requested
- [ ] Difficulty: % Answers Requested

CANCEL APPLY FILTERS

Problem Set Name					Avg # Problems to Reach Mastery
Combining Like Terms 7.EE.A.1					5
Solving 1-Step Equations 7.EE.E					8
Dividing a Decimal by a Whole Number 6.NS.B.3					10
Common Factors, GCF, Prime I 6.NS.B.4 EX					9
Ordering Rational Numbers 6.I					8
Ordering Rational Numbers 6.I					3
Recognizing Statistical Questi 6.SP.A.1					3
Reading a Box and Whisker Gr 6.SP.B.5					10
Reading a Box and Whisker Graph 6.SP.B.5	120	2355	53%	28%	6
Finding the Mean, Median, Mode, or Range 6.SP.B.5c	115	2345	53%	28%	6

Figure 44: Filtering content for the study in the final prototype

Once a problem set is selected, the researcher can then pick which problems to include in their study, and make any necessary modifications. By clicking through the graph view, the user can select which problems are used for each component of the study.



Dividing Proper Fractions 6.NS.A.1

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

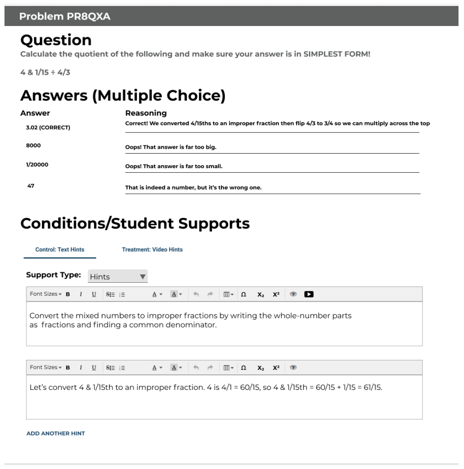
Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

Problem PR8QXA
Calculate the quotient of the following two fractions and make sure your answer is in SIMPLEST FORM!
3/7 ÷ 9/4

(a) Selecting individual problems in the final prototype



Problem PR8QXA

Question

Calculate the quotient of the following and make sure your answer is in SIMPLEST FORM!

4 & 1/15 ÷ 4/3

Answers (Multiple Choice)

Answer	Reasoning
3.02 (CORRECT)	Correct! We converted 4/15ths to an improper fraction then flip 4/3 to 3/4 so we can multiply across the top
8000	Oops! That answer is far too big.
1/20000	Oops! That answer is far too small.
47	That is indeed a number, but it's the wrong one.

Conditions/Student Supports

Control: Text Hints | Treatment: Video Hints

Support Type: Hints

Convert the mixed numbers to improper fractions by writing the whole-number parts as fractions and finding a common denominator.

Let's convert 4 & 1/15th to an improper fraction. 4 is 4/1 = 60/15, so 4 & 1/15th = 60/15 + 1/15 = 61/15.

(b) Editing individual problems in the final prototype

Figure 45: The content selection process in the final prototype

This final prototype provided a well-defined foundation on which we could model the resulting application. In particular, it helped describe the details of content selection and data analysis which were left largely unanswered at the end of B-term. Its design language – which refined that of the earlier four prototypes – was well-received by each team member and came to be emulated in the final application.

6 Web Application

6.1 Application Description

The application is built on Vue, a reactive JavaScript framework for webapps (Vue.js Team, 2020). Vue simplifies data flow and component management, which allowed us to more efficiently develop the application. While other reactive JS frameworks exist, Vue seemed like a good choice because both Mr. Emberling and Mr. Spinelli had some previous experience with the framework.

Shortly into the development process, we added Vuetify as an additional dependency. Vuetify is a Material design framework for Vue, containing a wide variety of prebuilt components (Vuetify Contributors, n.d.). The two main advantages to including Vuetify were 1) expedited development, as we were able to utilize many of the included components rather than create and test our own, and 2) consistency with the official Material Design specification. However, these advantages came at the cost of significant bloat in the number of dependencies included in the project. The decision to include Vuetify was weighed heavily by the team due to this concern. After consideration, it was decided that Vuetify would bring more benefits than drawbacks, especially since it obviated the need to include additional dependencies at a later time.

6.2 Workflow and Collaboration Techniques

Collaboration on the project's code was facilitated via a Git repository on the ASSISTments GitHub. Each new feature or fix was developed in a separate branch and merged into the master branch after being reviewed by each developer. In this way we were able to keep our contributions from interfering with each other while still developing simultaneously.

Mr. Emberling and Mr. Krichevsky had previous experience with Git, but Mr. Spinelli's experience was limited. With that in mind, Mr. Emberling and Mr. Krichevsky graciously assisted Mr. Spinelli in the basics of collaborative development via Git.

Code quality and durability were primary concerns at the beginning of the development phase of the project. With this in mind, we set up a Continuous Integration (CI) pipeline to validate both code style (by way of a strict ESLint ruleset) and correctness (by way of unit and end to end testing). Every time a member submitted a change to the codebase, it first had to pass through this pipeline to ensure that the change was working correctly and did not break any existing functionality.

During the final term of the project, we met four days per week. This included an hour-long comprehensive meeting each Monday for goal-setting and to map out tasks for the week, as well as 15-minute check-ins on Tuesdays, Wednesdays, and Thursdays. Frequent contact between team members facilitated collaboration on the project and gave us opportunities to ask each other for help and feedback. We also employed a Trello board to delegate tasks to each team member, which ensured that each person's responsibilities were clearly delineated and helped us to keep tabs on each other's progress asynchronously and communicated regularly via Slack.

6.3 Testing

Our approach to software testing was two-pronged: unit tests using the Jest framework (Facebook, 2020), and end-to-end (E2E) tests using the Cypress framework (Cypress.io, 2020). In short, unit tests were focused on small pieces (units) of code, while E2E tests worked on a larger scale and ensured compatibility between the different components. In general, the unit tests were aimed at single components; a common example was to test methods and event handling inside a component. Unit tests were also used to test basic rendering, although in general more sophisticated rendering checks were delegated to E2E tests. E2E tests also largely targeted a middle ground between unit testing and hands-on user testing. The framework we chose, Cypress, runs tests by opening a web browser and simulating user interaction with the webpage. E2E testing also enabled us to check larger portions of the application to make sure that pieces of the software worked together as a whole. While we still had to write the tests programmatically, they gave more assurances than unit tests that our application worked as expected in the hands of a user.

While the software is not yet mature enough to warrant user testing, a primitive form of hands-on testing was performed during the code review process. Each developer test-drove each branch before it was merged into master, meant to safe guard against glaring issues with the code. In addition, Dr. Ostrow reviewed the look and feel of new branches on a weekly basis to provide direction. As the application moves toward maturity, it would be prudent for the E-TRIALS team to coordinate with learning sciences researchers for proper hands-on testing.

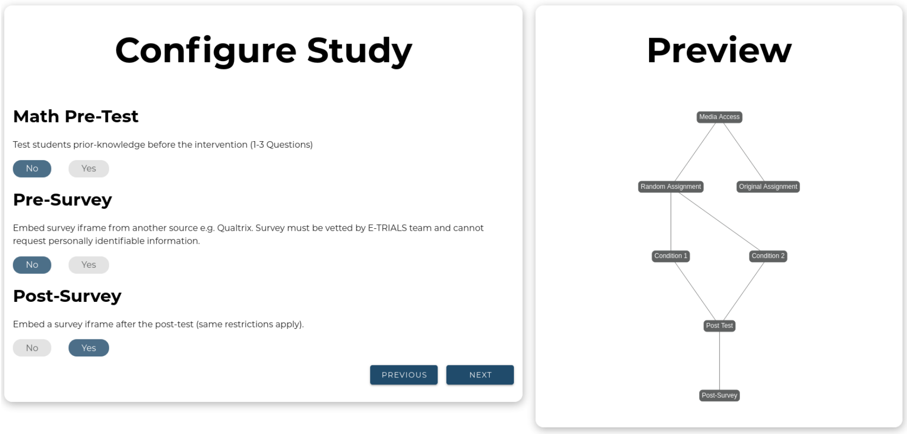
6.4 Development Progress

6.4.1 Reusable Components

One of the main advantages of using a reactive framework such as Vue is that one can leverage reusable components. These components ranged from specialized (e.g. the navigation bar) to general (e.g. a confirmation modal). Mr. Krichevsky developed modals for study creation and general confirmation, while Mr. Spinelli developed a reusable navigation stepper, built a reusable two-panel layout scheme, and iterated upon reusable experiment cards for the homepage.

6.4.2 Flowchart Visualization

One focus of Mr. Krichevsky's work was the flowchart visualization of the study that changes flexibly as the study is constructed. To ease the design of this component, Mr. Krichevsky selected vis.js (Vis.js Contributors, 2019), a graph visualization library. This, combined with the reactive nature of Vue, allows for instantaneous updates to the flowchart as different options are selected. A sample graph is shown in Figure 46.


Configure Study

Math Pre-Test

Test students prior-knowledge before the intervention (1-3 Questions)

[✓ No] Yes

Pre-Survey

Embed survey iframe from another source e.g. Qualtrix. Survey must be vetted by E-TRIALS team and cannot request personally identifiable information.

[✓ No] Yes

Post-Survey

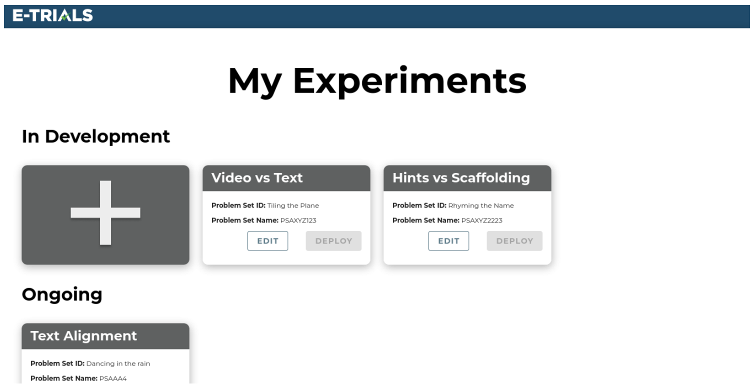
Embed a survey iframe after the post-test (same restrictions apply).

No [✓ Yes]

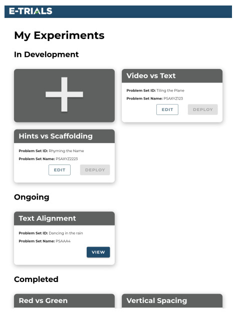
Figure 46: An example flowchart produced by the final application

6.4.3 Responsiveness

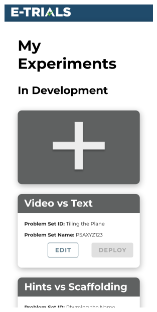
When we began working on the application at the beginning of D-term, the application was designed predominantly for desktop computers. While we do not know if researchers will have any reason to create studies on tablets or smartphones, we decided it was best to make the application fully responsive on all devices. Mr. Spinelli refactored some of the application templating and styling in order to achieve this goal. In particular, the Vuetify grid layout system proved invaluable for the homepage layout and for the reusable two-panel layout because it can automatically reflow content based on screen size.


E-TRIALS
My Experiments
In Development
Video vs Text
Problem Set ID: Tiling the Plane
Problem Set Name: PSAXYZ123
EDIT
DEPLOY
Hints vs Scaffolding
Problem Set ID: Rhyming the Name
Problem Set Name: PSAXYZ2223
EDIT
DEPLOY
Ongoing
Text Alignment
Problem Set ID: Dancing in the rain
Problem Set Name: PSAAA4

(a) The homepage layout on a PC



(b) The homepage layout on a simulated tablet



(c) The homepage layout on a simulated smartphone

Figure 47: The homepage layout on various device sizes

For large displays, gutters were added which prevented content from spreading out too far horizontally. The end result is a consistent implementation of our application's design language and a usable experience across all devices.

6.4.4 Data Table and Content Page

Mr. Spinelli began development on the content selection page, which includes a data table and filtering settings akin to those in the final prototype (see figure 44). Here, the Vuetify data-table and slider components were utilized in order to make the filtering truly reactive. This is an example of how the decision to include Vuetify as a dependency increased the team's efficiency; developing our own data table and sliders would have taken days or weeks of development and testing time.

Currently the content data table uses mock data built around a data store designed by Mr. Emberling. Lazily-loading the problem set information will be critical when transitioning to the real database due to the large volume of ASSISTments problem sets. Without a backend in place, the exact mechanics of this interaction are impossible to describe, and as such this must be put off until the backend is implemented.

The decision to use the Vuetify data-table component was a natural one because of the rich filtering, sorting, and reactiveness options. Indeed, there is built-in support for server-side filtering and sorting, which will be useful once a backend service is up and running.

6.5 Road Forward and Unanswered Questions

With the components we've built in hand, the ASSISTments research team should be able to complete a working application. There are still some unanswered questions that we feel should be addressed before final deployment. These are outlined below.

6.5.1 Backend

Currently the largest obstacle to completing the application is the lack of a backend. As of right now, all the data that a user sees in the application is mocked, and there is no server-side interaction. The backend will bring a plethora of design and development questions: what language(s) should be used? How much should it rely on frameworks or libraries? How will integration into the ASSISTments database work, and what modifications will the database need to support our application? These questions are outside the scope of this IQP but they must be answered by ASSISTments staff and developers in order to bring the application to market. A backend engineer was hired as we completed our project, with a start date of June 1, 2020, so the project is slated to move forward.

6.5.2 Documentation

Currently no documentation has been built for the webapp. While it makes sense to postpone adding documentation until the application is mostly finalized (to avoid having to rewrite significant portions if changes are made), it is still important to keep in mind during development. The E-TRIALS team should collaborate with the communications and teacher experience teams at ASSISTments to write and maintain this documentation.

6.5.3 Collaboration and Concurrency

It is common for researchers to collaborate on projects. With this in mind, allowing multiple accounts to access and edit the same study at the same time would be a useful feature. An approach similar to Google Docs could work well from a user-experience perspective, but setting this up – and testing it to reasonable standards – will be very time-intensive and could require significantly more effort on the backend. A more primitive approach is a "locking" style of collaboration, in which researchers cannot work concurrently but instead only one account can access a study at any particular time. While this is less effective for user experience, it would greatly simplify the infrastructure compared to a fully-concurrent model.

6.5.4 Data Delivery and Analysis

Most of the development so far has focused on study creation, and has ignored the rest of a study's life cycle. In particular, data handling for a finished study is completely unaddressed. While it is clear that some integration with ALI will be necessary in order to deliver data to researchers, the extent of this integration is not clear. However, it is certain that in order to integrate ALI's data reporting into the webapp, the ALI data flow will need to change in some way (in particular, the data should not be tied to Google Drive). The introduction of our webapp also represents an opportunity to upgrade ALI and introduce new features, if any are planned.

Integrating data reporting into the webapp allows us to enhance the automatic analysis features of ALI in a more flexible way than was possible with the current Google Drive-based ALI reporting. For example, a researcher may want to select new demographic covariates for a study after data collection has finished. Since that data is already in the database, the researcher could potentially update their study's analytic settings to include that covariate and generate a new data report.

With that said, the visual manner in which this data is presented and controlled is still completely unclear and will likely require extensive user testing in order to optimize it for researchers.

6.5.5 User Testing and QA

On the topic of user testing, we suggest that ASSISTments utilize its wide network of teachers, researchers, and scientists across the country to test-drive the application before it goes live. Putting the application in the hands of the people who will use it will allow the E-TRIALS team to tailor and tweak the software specifically for the needs of its users. This type of holistic testing is not possible with automated unit or end-to-end tests and is vital to ensuring that the application works as intended.

Even after the application is ready for release, it is certain that bugs will be discovered. To that end, we recommend that ASSISTments set up a dedicated channel through which researchers can submit bug reports or feature requests. Furthermore, ASSISTments should make an effort to *educate* researchers about this channel and *encourage* them to use it.

7 Contributions of This IQP, and Lessons Learned

At the outset of this IQP, the goals were open-ended; we were met with a problem to solve and little pre-determined direction on our ultimate goals. However, after significant effort alongside our advisors, we landed on a solidified path toward a real, usable tool for researchers. This process has taught us much about the virtues of project management, designing carefully for a use-case, and most of all, making powerful tooling not only available, but accessible.

As we have come to discover, ASSISTments' research tooling has a very powerful foundation. Researchers can build complex studies that answer difficult educational questions at little to no cost to them. However, over the years, the evolution of this tooling has made it a victim to a phenomenon that is not uncommon in large projects with high turnover: the Truck Factor (Bowler, 2005). Simply, the Truck Factor is a measure of how a project would continue if one member were to disappear (or were hit by a truck, as it were). In this case, the understanding of key elements of the platform are only understood by a small subset of individuals, such as the nuances of individual problem sets being locked away in spreadsheets, various processes being documented in a scattered fashion, and unintuitive study builder behavior. If certain members of the ASSISTments research team were hit by a truck, it is unlikely that research development would continue. With this in mind, building a tool that is as understandable and complete as possible can help to prevent such a scenario.

Our team's contributions represent a fresh set of eyes on an aging project. It is true that ASSISTments is undergoing a major architectural transition, and those changes will certainly have an impact. However, by putting our minds together to think through each detail of the current study design and deployment process, we believe that we have provided a strong path forward that will enable as much research as possible.

This project is concluding in the midst of the COVID-19 pandemic. With students around the country now learning from home, there has been a significant spike of interest in online homework and course delivery. ASSISTments is uniquely posed to step in as a premier platform for online classwork, and has accordingly seen a drastic increase in teacher and student traffic since the pandemic started. We hope that this leads to sustained growth in the pool of students for researchers to draw from once the E-TRIALS application is complete.

The team has full faith in both Dr. Ostrow and Mr. Emberling to lead the completion of this project, and we look forward to hearing about the research that is built using this tooling.

Appendices

A Study Building Time Tracking Spreadsheets

A.1 Mr. Krichevsky

Study	**Task**	**Minutes spent**	**Notes**
n/a, Derivation vs Formula	Picking Problem Set/Topic	30 minutes	A bit of time was spent finding something that would be derivable, which is more of a consequence of the experiment chosen than anything.
n/a, Difficulty Survey	Picking Problem Set/Topic	15 minutes	Finalize Problem Set/Topic in meeting 60 minutes
n/a, Derivation vs Formula	Determining what type of template was needed (including documentation of why this was the wrong type)	25 minutes	
PRABMVT6, Derivation vs Formula	Writing explanation intervention	10 minutes	
PSABDAFY, Derivation vs Formula	Template Setup	15 minutes	
PRABMVUD, Derivation vs Formula	Writing derivation intervention	15 minutes	
PSABDAFY, Derivation vs Formula	Attempting to modify study to use proper intervention model	30 minutes	Because this is not a simple video experiment, we had to make two different types of intervention, which the "intervention before" template does not solve
PSABDAFY, Derivation vs Formula	Having to rewrite the study to match intervention model	45 minutes	After realizing we cannot move sections around, my attempt to rework the template created version failed miserably once I went to test drive it
n/a, Derivation vs Formula	Flowchart	30 minutes	
n/a, Difficulty Survey	Picking "Hard" Problems	20 minutes	Used common wrong answer sheet
PSABDAVQ, Difficulty Survey	Setting up Structure	90 minutes	Still is not complete - waiting to hear back wrt debugging on IFT problems
PSABDAVQ, Difficulty Survey	Responding to feedback	5 minutes	Removed problems from post-test
PSABDAVQ, Difficulty Survey	Flowchart	20 minutes	

A.2 Mr. Spinelli

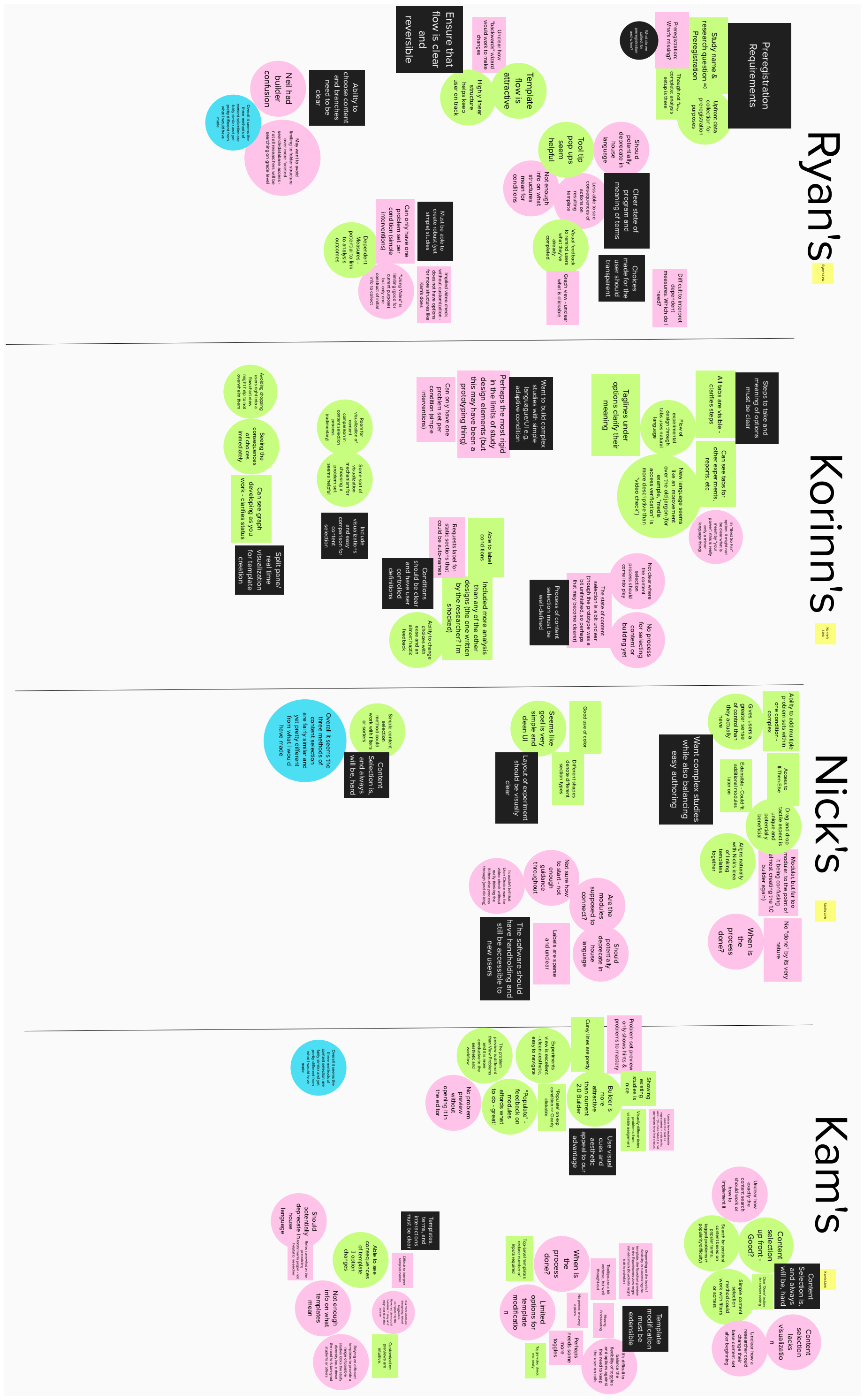
Study	Task	Minutes spent	Notes
N/A	Picking problem sets	60 minutes	
N/A	Finalize Problem Set/Topic in Meeting	60 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Select video (Khan Academy)	14 minutes	Was lucky that the first video I watched was suitable
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Turning problem set into comma-separated list of problem IDs (by hand)	12 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Creating ”intervention” (video) problem	5 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Finalizing information in the template builder	2 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Finding text instructions for the topic and creating problem with text instructions	14 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Restructuring control section to instead contain text instructions and skill builder	11 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Modifying video intervention problem to match style of text intervention problem	2 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Test drive	6 minutes	Did 2 test drives to see both types of instructions
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Deleting initial (top-level) message	1 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Re-adding control branch	4 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Re-adding [Control], [Experiment] designators	1 minutes	
PSAVUMN, Geometry - Rotations 8.G.A.3 / Video vs. text instructions	Creating Lucidchart	15 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Figuring out how to change font in problems	50 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Trimming selected problem set down to 20 problems	10 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Copying problem set and changing fonts to monospace manually	11 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Copying problem set and changing fonts to serif manually	10 minutes	

PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Finalizing information in the template builder	4 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Restructuring sections to include monospace and serif SBs	5 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Test drive	3 minutes	Did several test drives to ensure that all font possibilities were reachable
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Deleting initial (top-level) message	1 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Moving choose-condition to top level	6 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Creating Lucidchart	10 minutes	
PSABHZM, Algebra - Factoring Difference of Squares A-SSE.B.3b / Varying fonts in problem text	Changing hint fonts	69 minutes	Turned all text black in hint editor, then used my code editor's find/replace function on the source to change font
N/A	ALI doc request	18 minutes	

A.3 Mr. Emberling

Research Phase	Step	Time Spent	Good Things	Bad Things	General Notes
0 - Getting Familiar	Reading & Watching Step 1 page on TestBed	20 minutes	Why use skill builders? -> Access to students Examples give feel for research possibilities	General issues: Choose condition is not well explained Dependent measures are not well explained in examples Navigation is tricky (prog disclosure in nav is unintuitive) Vid 1 issues: Uses old version of testbed website Doesn't clarify techniques & indications (just examples) Vid 2 issues: Boredom example is vague - what was this study? Unexplained acronyms for problem sections	These docs should be reworked to follow a single worked example all the way through Vid 2 explains ARRS as useful BUT researchers can't control it. How to manage value prop? PS ID's are nice BUT no links given
0 - Getting Familiar	Reading & Watching Step 2 page on TestBed	30 minutes	Explains that researchers must build content Compells them to use skill builders Explains embedded vs orchestrated studies	"teacher example page" dead link Slide shows are overwhelming and difficult to parse Test mode referenced but unexplained Teacher Settings: rambling, unclear IRB coverage unclear. When a researcher finds their own teachers, do they need their own IRB? Examples of iframe embedding are broken or confusing If - Then - Else section explanation is confusing & has typos ALI section tags missing explanation Much about variablized templates, but they aren't used	I didn't watch all the builder vidoes b/c I have some experience w/ basic building Overall, the sequencing and segmentation of this page need work.
0 - Getting Familiar	Reading & Watching Step 3 page:	10 minutes	Explains how to create embedded studies vs using ASSISTments Direct Nice and short		
0 - Getting Familiar	Reading & Watching Step 4 pages	10 minutes	Gives chance to see reports directly Reasonably short	Sweet jesus the sheer volume of columns is intimidating	Need to abstract away some irrelevancies
1 - Research Idea	Drafting research inquiry	5 minutes			"Does the use of video feedback improve learning gains when compared with text?" This takes place outside of the platform. Difficult to measure how long this takes researchers, as this is a fluid process.
2 - Creating Problem Sets	Choosing Problem set as basis for experiment using tiny. cc/aproblems	10 minutes	Good to have centralized overview of available problem sets	Hard to track how many problems in a set in builder Need to flip back and forth between builder and overview	Not very sure of what's "good" or "bad" in the available stats. Want to avoid ceiling effect, so what % correct is appropriate? Same goes for other stats like problems to mastery
2 - Creating Problem Sets	Planning/Diagramming the study outline (1st draft)	30 minutes	Visualization simplifies thinking about structure & flow	Hard to specify hierarchy and flow	
2 - Creating Problem Sets	Revising study outline (2nd draft, with help)	15 minutes			Korinn explained problems with my structure and we fixed them together
2 - Creating Problem Sets	Creating problem set structure in ASSISTments	60 minutes		**Wasted 30 minutes creating structure in wrong order, had to throw out work. I initially tried making the innermost sections in the hopes of referencing them by ID's to nest them inside the outer sections, but this doesn't work in 1.0.** **It took 30 minutes to get my structure right once I knew what I had to do**	**Opportunity: Experiment builder can reduce this time**
2 - Creating Problem Sets	Adding problems to the problem sets	20 minutes		**Arduous copy pasting to duplicate an existing problem set by manually adding all its problems one at a time**	**Opportunity: Using problem set ID's to create sections in experiment will save time**
2 - Creating Problem Sets	Updating problems to apply intervention and serve as control	40 minutes		Added scaffolding questions to 5 problems and disabled hints on those problems. Also updated hints on 5 control problems to better align with the experimental problems. Navigation between problems was clunky (once you finish editing a problem, it's hard to return to where you were in the problem set to edit the next one)	
4 - Data Analysis	Deleting garbage columns	15 minutes			**Removed Peer explanation, work is sharable, templates, path info, and prerequisite and postrequisite skills**
4 - Data Analysis	Filtering out irrelevant students	5 minutes			Removed students who failed video check
4 - Data Analysis	Separate the conditions	60 minutes			**Opportunity: Doing this automatically will make this irrelevant (saving 60 minutes)** **Needed to create lists of problem IDs for each condition and check the first problem they 'really' did against lists**
4 - Data Analysis	Calculate post-test scores	15 minutes			**Search the problems each student did for the IDs of post-test questions**
4 - Data Analysis	Run T-Test	5 minutes			**Have to copy data into different format to suit t-testing (separate conditions into different columns instead of being interleaved with a column to denote condition**

B Design Synthesis Mural Board


Ryan's
Ryan's Link
Preregistration Requirements
Study name & research question => Preregistration
Preregistration: What's missing?
What do we collect for preregistration and when?
Upfront data collection for preregistration purposes
Though not fully complete, analysis setup is there
Ensure that flow is clear and reversible
Unclear how "backwards" wizard would work to make changes
Template flow is attractive
Highly linear structure helps keep user on track
Ability to choose content and branches need to be clear
Neil had builder confusion
Overall it seems the three methods of content selection are fairly similar and yet pretty different from what I would have made
May want to avoid limiting to folder structure over more faceted search/database access - not all researchers will be searching on grade level
Should potentially deprecate in house language
Tool tip pop ups seem helpful
Not enough info on what structures mean for conditions
Less able to see consequences of actions on resulting template
Clear state of program and meaning of terms
Visual feedback to remind users what they've already completed
Choices made for the user should transparent
Graph view - unclear what is clickable
Difficult to interpret dependent measures. Which do I need?
Must be able to create robust (yet simple) studies
Can only have one problem set per condition (simple interventions)
Dependent Measures - potential to link to analysis outcomes
"Using Video" is limiting (good for current purpose) but only one construct of initial info to collect
Implied video check without customization - does not have options for more structures like Kam's does
Korinn's
Korinn's Link
Steps to take and meaning of options must be clear
All tabs are visible - clarifies steps
Flow of experimental design through tabs uses natural language
Taglines under options clarify their meaning
Can see tabs for other experiments, reports, etc
New language seems like an improvement over the old jargon (for example, "media access verification" is more descriptive than "video check")
In "Best So Far" option: it might not be clear what is meant by "your power", (this is really only a minor language thing)
Want to build complex studies with simple language/UI e.g. adaptive condition
Perhaps the most rigid in the limits of study design elements (but this may have been a prototyping thing)
Can only have one problem set per condition (simple interventions)
Avoiding dropping users right into a flowchart view might help to not overwhelm them
Seeing the consequences of choices immediately
Can see graph developing as you work - clarifies status
Split pane/ real time visualization for template creation
Room for visualization of content comparison in content selection process (rudimentary)
Some sort of visualization mechanism for choosing a problem set seems helpful
Include visualizations and easy comparison for content selection
Requests label for static sections that could be auto-names
Able to label conditions
Conditions should be clear and have user controlled definitions
Included more analysis than any of the other designs (the one written by the researcher? I'm shocked)
Ability to change choices with ease and an almost haptic feedback
Not clear where the content selection process should come into play
The state of content selection is a bit unclear (though the prototype was a bit unfinished, so perhaps that may become clearer)
Process of content selection must be well-defined
No process for selecting content or building yet
Nick's
Nick's Link
Ability to add multiple problem sets within one condition - complex
Gives users a greater sense of control than they actually have
Want complex studies while also balancing easy authoring
Access to If-Then-Else
Extensible - Could fit additional modules later on
Drag and drop tactile aspect is unique and potentially beneficial
Aligns naturally with Nick's idea of linking templates together
Modular, but far too modular, to the point of it being confusing (almost creating the 1.0 builder again)
No "done" by its very nature
When is the process done?
Good use of color
Seems like goal is very simple and clean UI
Different shapes denote different section types
Layout of experiment should be visually clear
Simple content selection method could work with filters or sorters
Content Selection is, and always will be, hard
Overall it seems the three methods of content selection are fairly similar and yet pretty different from what I would have made
Not sure how to start - not enough guidance throughout
I couldn't tell that User Choice was for video check without really thinking the if-then-else process through (and clicking)
Are the modules supposed to connect?
Should potentially deprecate in house language
Labels are sparse and unclear
The software should have handholding and still be accessible to new users
Kam's
Kam's Link
Problem set preview only shows hints & problems to mastery
Curvy lines are pretty
Showing existing studies is nice
Experiments view is excellent - clean aesthetic, easy to navigate
The problem preview is different than View Problems and it is more conducive to the aesthetic and workflow
Builder is more attractive than current 2.0 Builder
"Populate" on exp condition => Clearly clickable
"Populate" - feedback on modules affords what to do - great!
No problem preview without opening it in the editor
Visually differentiates problems from variable assignment
Use visual cues and aesthetic appeal to our advantage
Overall it seems the three methods of content selection are fairly similar and yet pretty different from what I would have made
Unclear how exactly the content search should work or how to implement it
Content selection up front - Good?
Search for posttest content based on popular terms, tagged problems (+ popularity/difficulty)
Content Selection is, and always will be, hard
Clear "Done" button for content editing
Simple content selection method could work with filters or sorters
Content selection lacks visualization
Unclear how a researcher could change their base content set after beginning
Template modification must be extensible
Depending on the level of flexibility in customizing the template, the flowchart preview in the Experiments view might not add much (flowcharts might look too similar)
Tooltips are a bit verbose, but well thought-out
Missing Frontloading
No pretest or survey options
Perhaps needs some more toggles
It's difficult to balance the flexibility of toggles and options against the need to keep the user on rails
When is the process done?
Limited options for template modification
Top-Level templates reduce number of inputs required
Toggle video check etc. easily
Templates, terms, and interactions must be clear
Difficult to interpret template names
Should potentially deprecate in house language
Relies somewhat on the pre-existing ASSISTments jargon - not helpful to researcher
Able to see consequences of template option changes
Not enough info on what templates mean
Customization previews are intuitive.
Relying on different templates to provide a range of possible studies kicks the study diversity issue down the road to future grad students or others

References

ASSISTments. (2015). *Assistments testbed resource guide.* Retreived, May 9, 2020, from `http://www.assistmentstestbed.org/`.

ASSISTments. (2017, March 29). *Assistments advanced builder instructions site.* Retrieved May 9, 2020, from `https://sites.google.com/site/assistmentsadvancedbuilder/home`.

ASSISTments. (2019a, August 29). *ALI's raw data.* Retrieved January 6, 2020, from `https://www.etrialstestbed.org/research/research-process/4-analyze-data/ali-s-raw-data`.

ASSISTments. (2019b). *Assistments – free education tool for teachers & students.* Retrieved May 9, 2020, from `https://new.assistments.org/`.

ASSISTments. (2019c). *Efficacy research.* Retrieved May 9, 2020, from `https://new.assistments.org/research`.

ASSISTments. (2019d, August 29). *Research process.* Retrieved, May 9, 2020, from `https://www.etrialstestbed.org/research/completed-studies`.

Bowler, M. (2005, May 15). *Truck factor.* Retrieved May 9, 2020, from `http://www.agileadvice.com/2005/05/15/agilemanagement/truck-factor/`.

Cypress.io. (2020, May 5). *Why cypress?* Retrieved May 9, 2020, from `https://docs.cypress.io/guides/overview/why-cypress.html`.

Facebook. (2020). *Jest – delightful javascript testing.* Retrieved May 9, 2020, from `https://jestjs.io/en/`.

Heffernan, N. (2014, October 30). *Video instructions for creating a study in assistments.* Retreived May 9, 2020, from `https://www.youtube.com/playlist?list=PLAYFxbnTKguxjvFKBPbew-dKt6Q2G_doV`.

Massachusetts Institute of Technology. (n.d.). *Scratch - imagine, program, share.* Retrieved May 9, 2020, from `https://scratch.mit.edu/about`.

Vis.js Contributors. (2019, December 29). *vis.js.* Retreived May 9, 2020 from `https://visjs.org/`.

Vue.js Team. (2020, February 24). *Introduction – Vue.js.*

Vuetify Contributors. (n.d.). *Why vuetify?* Retrieved May 9, 2020, from `https://vuetifyjs.com/en/introduction/why-vuetify/`.