Leveraging User Testing to Address Learnability Issues for Teachers Using ASSISTments

by

Joshua Bodah

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

May 2013

APPROVED:

Professor Neil Heffernan, Major Thesis Advisor

Professor David Brown, Thesis Reader

Professor Craig Wills, Head of Department

Abstract

The goal of this thesis is to demonstrate how user testing can be used to identify and remediate learnability issues of a web application. Experimentation revolved around ASSISTments (www.assistments.org), an intelligent tutoring web application in which teachers create virtual classrooms where they can assign problem sets to their students and gain valuable data which can be used to make informed decisions. Recent log analysis uncovered very low task completion rates for new users on tasks that were intended to be trivial. Suspecting that this could be due to poor user interface design, user tests were conducted to help identify usability problems. Sessions were analyzed, and changes were made between each user test to address issues found.

Feedback from user testing led to the implementation of an embedded support system. This support system consisted of a splash page which gave an overview of how the system should be used and a collection of context-sensitive tooltips which tried to give the user instructions on what to do as well as explain various parts of the interface. A randomized control trial was performed to measure the effectiveness of the embedded support. 69 participants were shown one of two interfaces: one with embedded support and one without. Task completion rates were analyzed for each of the groups. We found that the support system was able to influence which links a user clicked. However, although the support system was intended to address poor task completion rates, users in the conditions had similar task completion rates regardless of whether the support system was enabled.

Acknowledgements

I would like to thank Neil Heffernan for providing me with the opportunity to be part of the ASSISTments lab.

I would also like to thank David Brown for reading my thesis and giving me useful advice and direction.

Last, but not least, I'd like to thank the members of the ASSISTments lab for making it all worth it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Formal user interface development methods have contributed to significant usability gains in a number of application areas. In this paper we discuss how formal usability methods were applied to ASSISTments, an intelligent tutoring web application. Our focus is on teachers who are new to the system. We discuss how several user tests were conducted to assess the user interface and uncover usability issues. New interfaces were developed between user tests to try an address usability issues found. Additionally, we ran a controlled experiment to analyze the effectiveness of an embedded support system which was designed to give users a better understanding of the system.

The following sections will give a general overview of what ASSISTments is and how it can be used. We will discuss why we felt the application had usability issues.

1.1 ASSISTments

ASSISTments is an intelligent tutoring web application developed by the Tutor Research Group (TRG) of WPI [RMN$^+$07]. In ASSISTments, teachers can create a virtual classroom, have students enroll in their class, assign problem sets to their class for their students to work on, view reports of how their students did, and provide remediation using a variety of interesting features. Teachers are also allowed to build, edit, and share content. The system provides users with a lot of flexibility, freedom, and depth.

Teachers in ASSISTments are generally considered to be proficient at the most basic level if they know how to complete three major tasks: create a class, assign content, and view a report. While there is no data to support it, there is an intuition within the TRG that the best way for a new teacher to learn how to complete these major tasks is by following a specific sequence of steps (which we will henceforth refer to as the "TRG-proposed learning sequence"). It is believed that new teachers should, in order:

1. Create a class.
2. Assign pre-built content to their class.
3. Enrolling in their own class as a student.
4. Complete their assignment as a student.
5. View the data they generated in their Item Report as a teacher.

The validity of whether or not this sequence is the best way for a teacher to learn the system is up for debate, but it certainly seems reasonable. Teachers who do this will have gained experience handling most of the major structural components of the system.

1.2 Problem Background

Research in ASSISTments up until now has mostly targeted students, usually in some attempt to show an increase in student learning [PGBH12][GBH10][BOM+12][BWH+08].

Little research has been conducted on teachers using the system, and none has been done that addresses usability. However, recent analysis has shown that teachers, in general, have very low task completion rates. We analyzed teachers who created an account during the 2011-2012 school year and found the following statistics:

- 53% created a class;
- 17% enrolled 5+ students;
- 17% enrolled in their own class as a student;
- 33% assigned any content;
- 22% assigned WPI certified content;
- 18% had students complete at least 1 assignment;
- 12% had students complete at least 5 assignments;
- 50% built a problem set.

As one can see, only half of new users even complete the first step of the "TRG-proposed learning sequence". These numbers don't come as a big surprise. It doesn't take much effort to realize that the site is complicated and confusing.

In-depth personal training has been the TRG's primary way of teaching the system to new teachers. As the system's popularity grows and spreads, the requirements for training have been stretched beyond the TRG's capabilities. In the Fall of the 2012-2013 school year, a controlled experiment was performed to measure the effectiveness of an online class. Half of the users who signed up for ASSISTments were enrolled in an online class which lasted several weeks. In this class user's were provided with assignments which consisted of using ASSISTments to do various things (such as create a class and build content). Users were also provided with a forum to ask questions and communicate with other users. The control group did not have any class and had to learn ASSISTments from the interface and documentation alone. Users in the online class were found to have no significant advantage over users who simply tried using the site with existing documentation. The only difference was that the online class took a lot more time, money, and effort.

At that point, the TRG decided to test if the problem was inherently a user interface problem. Usability, learning theory, and user interface design techniques were researched and applied in an effort to address this learnability problem. It was decided that a series of user tests would be performed to identify usability issues and drive development decisions aimed at addressing them. The development of these tests was then analyzed further using a controlled experiment to assess the effectiveness of the changes.

1.3 Problem Significance

Our research has two significant components. First, it is a case study which shows how a formal user interface design process can be used to make a site easier to use. It also examines the effectiveness of embedded support as a tool to address learnability issues.

1.4 Usability

Before we talk about the various literature that exists, it makes sense to first define *usability*. The International Organization for Standardization (ISO) defines usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”

[fS98]. Additionally, Nielsen explains that usability is composed of five principles [Nie12]:

1. **Learnability** How easy is it for users to accomplish basic tasks the first time they encounter the design?

2. **Efficiency** How quickly users can perform tasks once they've learned the design?

3. **Memorability** How easily can users regain proficiency with a design after a period of not using it?

4. **Errors** How many errors do users make, how severe are those errors, and how easily can they recover from them?

5. **Satisfaction** How much do users like the design?

In our research we are particularly interested in learnability. We want to make it easier for new teachers to use ASSISTments. This mean that we will be interested in knowing how they reason about the user interface. We will want to identify the parts of the interface make it difficult to understand which tasks the system can be used for as well as how the system can be used to perform those tasks.

Chapter 2

Literature Review & Background

2.1 User Research

It makes sense to start by trying to understand the users who use the ASSISTments interface. The target user base for the interface we are concerned with consists primarily of middle school math teachers. From interacting with users, we know that most of these teachers were born roughly between 1950 and 1990. Note that this population stretches the technology divide caused by the introduction of the Internet, so users have a wide range of previous experience with computers. Some teachers are extremely well-versed in or have come to teaching from technology-centric fields such as engineering, while others have very little experience using the Internet.

Bridging this gap between users is a serious challenge. Understanding how individuals use technology has provided us with a deeper insight into fundamental issues relevant to the existing learnability issues in ASSISTments.

2.1.1 Teachers and Technology

Technology is a rapidly growing force in education, and many schools are approaching a one-to-one student-to-computer mapping. Maine, for example, recently started a program requiring middle schools to provide laptops for all of its students [SL04]. Despite this rapid growth in technology, survey results show that only 20% of teachers feel well prepared to use technology in the classroom [CFCM02].

While teachers' feelings about technology improve greatly with professional development [CFCM02], training is costly and doesn't scale well to a large number of users. As an application developer, it is crucial to understand that many users already have trouble using technology to begin with. This implies that many teachers may not be familiar with certain paradigms that those of us who use computers every day take for-granted (e.g. the concept of copying and pasting text). An effective interface for teachers would be usable even by teachers with minimal experience with computers.

2.1.2 Learning Theory of Computer Applications

Carroll and Rosson published an influential paper in 1987 which addresses issues relative to the learnability of computer applications [CR87]. In particular they note that people have difficulty learning how to use computers, and peoples' skills with an application tend to peak before mastering a system. Carroll and Rosson claim that these problems are due to "conflicting motivational and cognitive strategies" as opposed to design.

They introduce the concept of the active user. In the case of an inexperienced computer user, this could refer to someone who is overwhelmed by advertisements, instructions, graphics, and links, among other things. Such users do their best to make sense of all of the information they are given and make an educated decision, but it could be the case that some information is ignored. These users often just try something and see what happens; they “simply strike out into the unknown”. Experienced users are only different in the sense that they can build on their previous knowledge. In both cases, users rely on adaptive learning strategies to solve issues of learnability. However, these adaptive strategies may not be adequate for a given situation, and this leads to learning problems.

Carroll and Rosson continue to describe two paradoxes. The first is the production paradox. Users only want to learn new tools so that they can use them to get work done, but getting work done can be an issue if users lack the skills necessary to use a tool. They do not typically want to simply learn a tool; they want to learn that tool to do something with it. This explains situations where users may get frustrated with long-winded tutorials and wizards.

The second paradox Carroll and Rosson discuss is one they call the assimilation paradox. Often the new concepts that people learn are built on previously mastered knowledge. However, it must also be true that people learn other ways or else they would never learn new things. This previous knowledge can sometimes influence users such that they are unreceptive to new ideas and they can get stuck by relating existing knowledge to their current context incorrectly. This can lead to a negative transfer of learning, in which a user’s learning and performance are impaired by some previous experience or behavior.

Carroll and Rosson suggest a number of ways to attack, mitigate, and design for these two paradoxes. For the production paradox, they suggest that a system make learning rewarding and easy, and that it utilizes the user’s desire to learn the system. For the assimilation paradox, they suggest that designers either try to avoid any potentially confusing connections caused by previous knowledge or to embrace the user's previous knowledge by describing the system as something familiar. Additionally, designers should anticipate situations in which previous experience may fail the user while exploiting the situations in which it is beneficial.

2.1.3 Eye Tracking

Knowing how users physically react to interfaces is also of importance to understanding effective user interface design. Nielsen reports that English-speaking web users spend 69% of their time viewing the left half of the page and only 30% of it viewing the right half [Nie01b]. He goes on to recommend that all of a site's most important information (like navigation) be all the way to the left so it may get the most attention.

In another article, Nielsen also relates that plain text, along with faces and sexually explicit images, is one of the most effective design elements for attracting a user's attention[Nie07]. He reports that regular internet users have been trained to ignore anything that resembles an advertisement, hence the adherence to plain text.

He also talks about the F-pattern that users use to scan pages. Users' eyes follow along the left side of the screen while scanning. When they find something interesting they read across. The pattern makes the shape similar to the letter 'F'. Nielsen also relates that scanning is more common than reading[Nie07].

2.2 User Interface Design

User interface design is sometimes written off as an intuitive matter in which the guiding principles are obvious. Gould and Lewis [GL85] suggest that the process of designing usable human-computer interfaces is guided by three principles:

1. have an early focus on your users and the tasks they perform

2. measure interface changes empirically

3. have an iterative development cycle

They conducted a survey of systems planners, programmers, designers, and developers in the 1980's asking them to list the steps they thought one should go through when designing a computer system for end users. From this experiment they found that only 62% of the population mentioned an early focus on users, only 40% mentioned doing any sort of empirical measurement, and only 20% mentioned some form of an iterative design process. While Gould and These results suggest that there is some level of confusion as to what steps one should take while developing an interface as to maximize its effectiveness, and that the process for effective user interface design can not be entirely intuitive.

Today user interfaces get much more attention than did in the days of Gould and Lewis, but their proposed process of iteratively getting feedback from users, creating changes, and testing those changes is the heart of modern user interface design.

2.2.1 General Guidelines

We'll be designing a user interface for users in an actual production environment so it makes sense to gain a pragmatic understanding of what makes an interface effective. Carroll and Rosson's ideas discussed previously are the theoretical backbone to a lot of the ideas we are about to present [CR87].

Krug's book *Don't Make Me Think* is a practical guide to interface design and is full of web usability insight [Kru06]. As the book's title suggests, one of the major ideas Krug pushes for is not making the user think. In other words, make the interface explicit and clear enough so that users don't need to spend time trying to deduce how it works. The site should be self evident. Using web conventions that have already been established is one way to achieve this and reduce the mental load on your users.

Krug gives examples of using wording that is easy to understand (e.g. "Jobs" vs. "Employment Opportunities") and making visual elements clear (e.g. a button that is clearly clickable vs. a string that may or may not be clickable). The idea of using familiar knowledge and making things self-evident are exactly what Carroll and Rosson recommended to deal with the paradoxes of the active user.

Krug goes on to explain some of the ideas presented by Carroll and Rosson[CR87] and Nielsen[Nie07]. He explains how to design pages for scanning suggesting readers to create clear visual hierarchies, use conventions, break pages into clearly defined areas, make clickable things obvious, and minimize visual noise (e.g. condensing large blocks of text, eliminating unnecessary visual elements, etc.).

Krug highlights two features which he considers to be the most important features of any web site: the navigation and the home page. Navigation is particularly important because computers do not give us the same intuitions as real life navigation does. He mentions that we have the same sense of sense of scale, direction, or location as we do when we are navigating a physical space, so having good navigation in a virtual space is a must. Navigation acts as a familiar fallback for when one is lost. It also tells users what the site provides by establishing a visual hierarchy and tells them how the site can be used.

Some recommendations that Krug gives include having a site logo in the top left of the screen, limiting navigation to two levels and breaking navigation down into clearly defined sections and subsections, separating utilities (e.g. Help) from other navigation elements, having a way to search as well as a way to return to the home page, and using breadcrumbs, text at the top of a page which details how the user got there (like a trail of breadcrumbs in a maze).

The home page has different objectives. Krug explains that the home page should show users what they are looking for, expose them to some new things that they might not have considered and might be interested in, and show users where to start using the site. A home page should convey the message of the site quickly to the user.

Between the navigation and the home page, users should have a clear idea of what the site can be used for and how they can get started doing what they want to do.

2.2.2 Paper Prototyping

Paper prototyping is a technique in which an interface design is rapidly produced on a piece of paper. It typically occurs early in an interface's design phase and has the advantage of abstracting many of the details that one might have to focus on during implementation. The key here is that paper prototypes are meant to be cheap and rough. Complex interactions can be simulated by using multiple pieces of paper.

Paper prototypes can be used during user tests to allow for quick iterations and are effective at testing layouts, wording, colors, and getting a general feel for how a user interface might work prior to implementation. They can reducing the overall cost of producing an effective interface since they can find usability problems before any costly development is done.

2.3 Forms of Instruction in Computer Applications

Most people would agree that having someone teach you something is the best way to learn about it. Unfortunately, that paradigm just isn't practical in computer applications. While the best-case scenario for any application is that user's know how to use it from the very start without any instruction, some applications require stand-alone forms of instruction to ensure that users understand how to use it.

Here we discuss some of the types of instruction or support that computer applications have used.

2.3.1 Documentation and Manuals

Many websites provide support to their users in the form of documentation. These are typically elaborate, heavy-weight documents that are thorough but wordy. Essentially, these are the digital equivalent of paper manuals. They often times reside in their own section of a website, forcing users who want to view them to experience some misdirection from their current goals.

Documentation is, in fact, so ineffective as a teaching tool that usability expert Jakob Nielsen has devised two "Laws of Computer Documentation": the first being that "people don't read it", a slight exaggeration to prove the point that users very rarely read documentation and the second being that "users read system documentation only when they are in trouble" [Nie01a].

Retting reiterates the point that people typically don't read documentation and goes into detail distinguishing different styles of documentation, commenting on their effectiveness and practical use [Ret91].

2.3.2 Help Menus

Current research supports the claim that users, in general, do not like to refer to external help when learning. Grayling conducted ten user tests (five with inexperienced users and five with experienced users) to study how users approached scenarios in a system with a documentation support base [Gra98]. He and his team took great care when designing the support base by performing usability tests throughout its design. Links to the support base were kept in a "Help" menu on the user interface. When it came to testing a working prototype, he and his team found that users initially approached scenarios by trying to learn on their own through trial and error. They reasoned about the information provided by menu titles, tried something, and scanned dialog boxes for anything that seemed relevant to the scenario they were working on.

While this behavior was expected, Grayling was surprised to find that users resisted clicking on the "Help" menu even after failing to learn on their own. He reports that some subjects showed explicit signs of frustration and still ignored the "Help" menu. Some subjects, after failing to learn the interface on their own, did refer to the documentation, but they read it quickly and did not pay much attention it. This just led to more errors. Grayling noticed that those users who – usually out of frustration – did take the time to read the documentation in a careful and methodical manner did perform better. However, users were still very adamant about not using the "Help" menu. In Grayling's own words: "their whole attitude seemed to be: 'I don't want to be here! I don't want to read all this information that you have so carefully designed and written!"'. Grayling sums up his findings:

- Users go to the help menu only as a last resort
- They bail out of help early if a "clue" is not seen in the first one or two screens

- They are reluctant to even scroll down a help screen
- General overviews are often ignored

Spool reports similar results after conducting over 2,000 user tests [Spo96][Spo97]. Grayling also claims that the users did seem to like the embedded help they received. He hypothesizes that there are several guiding principles for creating a good help system. In particular, he believes that a help system should be:

- Context-specific
- Easily available
- Obvious to invoke
- Useful
- Non-intrusive

2.3.3 Embedded Support

Embedded support refers to any help that is given to the user that does not require them to leave their current context in a computer application. Parush and Parush explain how they used embedded support to solve usability issues of users not knowing where to begin using a system and of users not knowing what buttons mean [PP01].

2.4 Usability Evaluation Methods

If science has taught humanity one thing, it's that our intuitions are often wrong. Due to this, testing is of utmost important when developing a user interface. Small changes to an interface can have very large results. In one case study involving an e-commerce site, Kohavi et al. studied the effects of adding an input box for a coupon code to a payment form in a checkout process [KLSH09]. One might intuitively believe that giving users a way to use coupons to save money would increase sales. However, upon testing the interface, they found that the interface resulted in a 90% loss of revenue. This could have been due to several reasons: the coupon code could have reminded users that they might be able to find the products for a cheaper price elsewhere; entering the coupon code required typing which is error-prone; and the user experience was disrupted because the user felt the need to go and search for a coupon code.

The following sections describe three of the major ways of evaluating an interface's usability: user testing, heuristic evaluation, and A/B testing.

2.4.1 User Testing

User testing is the most basic, and perhaps the most useful, form of usability testing. It simply consists of watching new users – preferably ones representative of those who would actually use your interface – perform tasks with an interface design. The main idea behind user testing is that those who have worked intimately with a project or interface are already experts on the system and have already made mistakes and learned from them. By working with new users, one gains tangible insight when the user make mistakes with the interface. Theses mistakes can usually be mapped back to specific interface design decisions and act as aids in identifying design flaws.

There are many variations in user testing that vary from case to case. One particular area for variation is in the choice, length, and frequency of the tasks for the user. In some cases it may be best to give the user one broad scenario, while in others it would be more revealing if the user was given many short, specific tasks.

The “think aloud” protocol, as described by Lewis[Lew82], is the backbone of many forms of user testing. Users are encouraged by the test administrator to speak about what they are trying to do, what they believe elements of the design will do, and what they think they need to do next to achieve their goals. This allows the observer to better understand the user’s thought process relative to the task and design. Observers may take notes, which are often in the form of audio-video recordings and screen captures, which they can compile later to draw inferences about how well the design worked and what improvements could be made. It is often the case that the testing is done in a dedicated testing room equipped with a one-way mirror so that others can observe the test as well.

It is important during the test that the test administrator let the user provide feedback which is uninfluenced by the tester. The user should be free to fail or succeed in using the design to complete tasks. With that said, it’s important to keep in mind the reason why a test is being performed. For example, if the purpose of the test is to identify new usability problems, then the test administrator should make sure that the test time is used effectively as to pursue this goal. There’s no sense in frustrating a confused user, especially if no new useful data is coming from it. Krug writes about how to conduct user testing inexpensively [Kru10]. He covers everything from scheduling sessions and finding participants to analyzing results.

2.4.2 Heuristic Evaluation

Nielsen and Molich developed heuristic evaluation in 1990 [NM90]. It is a usability testing method in which evaluators are given a list of heuristics which they will use to judge an interface design. Evaluators need to be trained, but they do not need to be representative of the users who will use the design. The purpose of heuristic evaluation is to identify problems with a design without having to go through the effort of testing it with users. It is cheap and can be done with any individuals capable of providing an unbiased evaluation of the design based on the heuristics chosen. Nielsen defines ten heuristics in his book *Usability Engineering*[Nie94]:

1. **Visibility of system status** The user should always know what the system is doing and receive responses in a reasonable amount of time.

2. **Match between system and the real world** The system should abide to the standard domain language used by the user and follow real-world conventions.

3. **User control and freedom** The system should provide clear ways to easily leave an unwanted state.

4. **Consistency and standards** The system should use the same language and protocols throughout its entirety so as to keep from confusing the user.

5. **Error prevention** Eliminate error-prone conditions or alert the user before they commit an action which may result in an error.

6. **Recognition rather than recall** Minimize the user's memory load by making objects, actions, and options visible.

7. **Flexibility and efficiency of use** Use accelerators to hasten the interaction for expert users, but hide them from novice users so that the system may cater to both.

8. **Aesthetic and minimalist design** Hide information that is irrelevant or rarely needed.

9. **Help users recognize, diagnose, and recover from errors** Express error messages in plain English which indicate the problem and propose a solution.

10. **Help and documentation** Provide help and documentation that is easy to search, focused on the user's task, and give clear and concise instructions.

2.4.3 A/B Testing

A/B Testing is a way of running a *randomized control trial* to measure the effects of interface changes on the web [KLSH09]. It is sometimes also referred to as split testing or multivariate testing. In this approach, two (or more) interfaces are designed for the same task and fully implemented. Users are randomly split into experimental groups (one for each interface) and each is shown a different interface. An overall evaluation criterion (OEC), which is also sometimes called a key performance indicator, is predetermined and measured throughout the experiment. The OEC's purpose is to monitor some goal objective. For example, an OEC that an e-commerce site might use is the number of sales, so the experiment may measure the difference in the number of sales between multiple interfaces. If the difference is found to be statistically significant, it can be said with some degree of accuracy that one interface is better than the other given the OEC chosen.

Kohavi and Longbotham mention several important topics regarding A/B testing such as the importance in choosing an OEC in advance, how to avoid web crawlers which may skew results, and the pros and cons of various implementations. Some interesting design pitfalls (such as picking a poor OEC like "time spent on a page"), as well as explanations of the analysis on some puzzling experiments, are also documented [KDF$^+$12][KCL09][CFKL09]. One particularly interesting finding that Kohavi mentions (with respect to a study done at Microsoft) is that only one third of the ideas that were designed to improve some OEC actually showed an improvement after testing [KCL09].

A/B testing is particularly good at detecting small changes. Sometimes these small changes can have very large impacts. Siroker writes about how simple A/B testing was used during Barack Obama's presidential campaign to try and gain more supporters [Sir10]. Splash pages with different text and images were compared, the best of which had an 11.6% supporter sign-up difference. The average supporter ended up donating around $21 which translated to an additional $60 million in donations. Burke provides a list of problems that are often approached by applying A/B testing which includes making decisions about page layouts, navigation schemes, button placement, category names, and much more [Bur05].

2.4.4 Comparison of Usability Evaluation Methods

The three previously mentioned evaluation methods are all very different; each excels at something different. User testing can be very insightful as it can help the testers understand why a particular error is occurring, leading to a potential for a larger benefit. It helps the test team in determining what is confusing. User testing also doesn't need full implementations and can work on paper prototypes. However, pinpointing small differences through user testing can be difficult. User testing also requires a significant amount of time and money to schedule and conduct. Lastly, the conditions of the test environment may not accurately reflect those of the real world [Nie05].

Heuristic evaluation is very cheap and does not require representative users at all. This makes it a very convenient choice for testing as it doesn't require finding and scheduling specific types of users like user testing does. It also works on paper prototypes. However, the biggest criticism for heuristic evaluation is that it does not involve the end-user at all. Some believe that heuristic evaluation loses credit since it does not measure actual behavior under real-world conditions.

A/B testing can be very cheap in terms of time and money, and it is easy to set-up once a testing framework is established. However, it can require a large number of samples to yield a statistically significant result. For this reason, it is particularly popular on the web because of the amount of traffic that a site can get. With enough traffic, it can be performed on a weekly basis and can be used to test almost anything. A/B testing can be used to detect very small differences with high statistical significance. Some drawbacks include a restriction to measuring a single OEC at a time, they only work with fully implemented designs, they often require a large sample size to get meaningful results, and they do not provide behavioral insight like user testing [KLSH09].

2.4.5 Final Word

It's important to remember that each of these usability methods have specific purposes. Heuristic evaluation, for example, is very good for getting meaningful feedback very quickly in the early design stages as well as with implemented systems. In an ideal world, it could help narrow down which interfaces to try with actual users. User testing is excellent for getting behavioral insight which can help identify large misconceptions. A/B testing excels where user testing is too unclear. It is best used for aiding in decisions about small changes like button placement or color.

Chapter 3

Preliminary Usability Analysis

3.1 Motivation

Prior to performing any user testing, a preliminary analysis of the interface was performed as preparation. The interface was analyzed with respect to the principles of design researched and discussed earlier in this paper. This process was informal and did not follow any specific heuristics. The purpose of this analysis was to see if poor usability seemed like reasonable cause for the issues that teachers were experiencing.

3.2 ASSISTments Object & Interaction Models

In this section we will define the vocabulary, structure, and interaction model used in ASSISTments. To make this section easier for the reader, we will denote **objects in bold** and *actions in italics*.

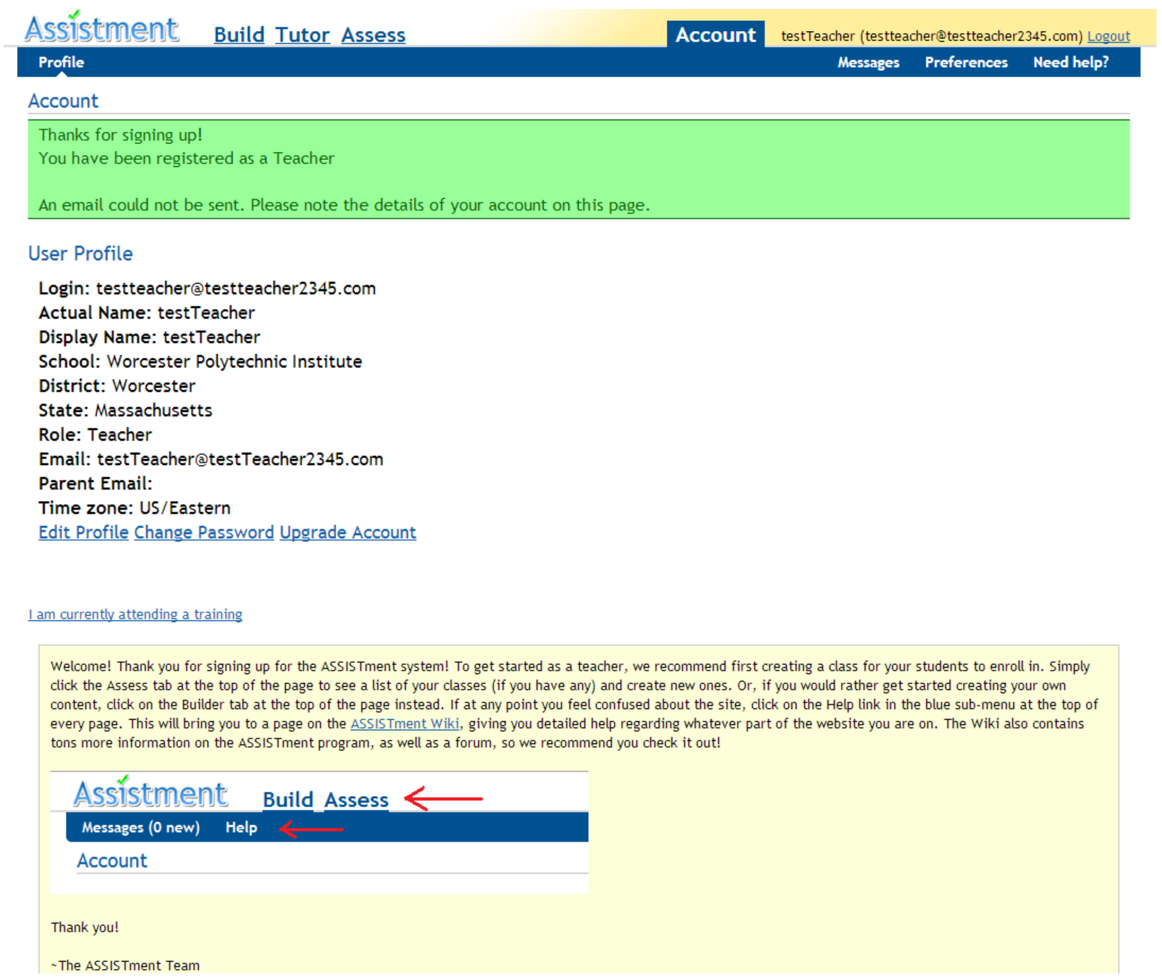
For all relevant purposes, there are two types of users in ASSISTments: **students** and **teachers**. Both types of users must first create an account. Teachers may create **classes**. Classes can contain students. A student must *enroll* in a class.

A **problem set** is a collection of questions. These questions are called **assistments**. There is a great deal of pre-built content that teachers have access to. Problem sets are organized in **folders**. Teachers may *assign* problem sets to their classes. These assigned problem sets are referred to as **assignments**. Students enrolled in a teacher's class may *open* any assignment in the **Tutor** and *complete* that assignment. After that, a teacher may view the student's data in the **Item Report**.

Teachers have several other special privileges. They may *build* content, whether it be a problem set or an assistment. They may also *organize* their assignments and built content into **folders**. Lastly, any teacher is allowed to do anything that a student can do. That is, every user that is a teachercan act as a student.

3.3 Navigation Layout

The general navigation of the site is broken down into a tab structure (see Figure 3.1). There are four major tabs: Build, Tutor, Assess, and Account. Students can see the Tutor and Account tabs, while teachers can see all four tabs. Each tab is associated with different tasks.


Assistment
Build Tutor Assess
Account
Profile
Messages Preferences Need help?
Edit Profile Change Password Upgrade Account
I am currently attending a training
Assistment
Build Assess
Messages (0 new) Help
Account

testTeacher (testteacher@testteacher2345.com) Logout

Account

Thanks for signing up!
You have been registered as a Teacher

An email could not be sent. Please note the details of your account on this page.

User Profile

Login: testteacher@testteacher2345.com
Actual Name: testTeacher
Display Name: testTeacher
School: Worcester Polytechnic Institute
District: Worcester
State: Massachusetts
Role: Teacher
Email: testTeacher@testTeacher2345.com
Parent Email:
Time zone: US/Eastern

Welcome! Thank you for signing up for the ASSISTment system! To get started as a teacher, we recommend first creating a class for your students to enroll in. Simply click the Assess tab at the top of the page to see a list of your classes (if you have any) and create new ones. Or, if you would rather get started creating your own content, click on the Builder tab at the top of the page instead. If at any point you feel confused about the site, click on the Help link in the blue sub-menu at the top of every page. This will bring you to a page on the ASSISTment Wiki, giving you detailed help regarding whatever part of the website you are on. The Wiki also contains tons more information on the ASSISTment program, as well as a forum, so we recommend you check it out!

Thank you!

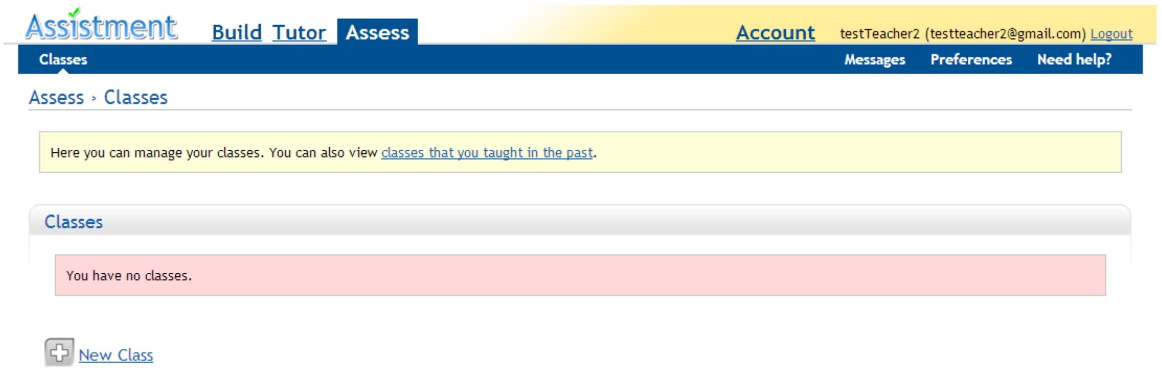
~The ASSISTment Team

Figure 3.1: (Original design) Account tab

The Build tab allows a user to create their own custom problem sets and assistments.

The Tutor tab allows a user to enroll in a class. It also allows the user to view a list of a class' assignments. Users can open up an assignment in the Tutor, where they can then answer assistments and complete the assignment.

The Assess tab is where a teacher can create a class (Figure 3.2). They can also disable a class and view classes that they might have taught in the past (a list of disabled classes). A teacher can view their class' assignments. They may also browse through folders there to assign a problem set. Once a student has completed an assignment, the teacher may then view the Item Report for that assignment.


Assistment
Build Tutor Assess
Account
testTeacher2 (testteacher2@gmail.com) Logout
Classes
Messages Preferences Need help?
Assess › Classes
Classes
New Class

Here you can manage your classes. You can also view classes that you taught in the past.

You have no classes.

Figure 3.2: (Original design) Assess tab

The Account tab displays a user's account information such as their login and e-mail (Figure 3.1). Users can edit their account information here.

In summary, the important parts of the navigation are structured as in Figure 3.3. The white blocks represent the four navigation tabs, each of which is always accessible. The blue blocks are more for navigation (e.g. viewing a list of assignments) while the green blocks are more task-oriented (e.g. can assigning a problem set).

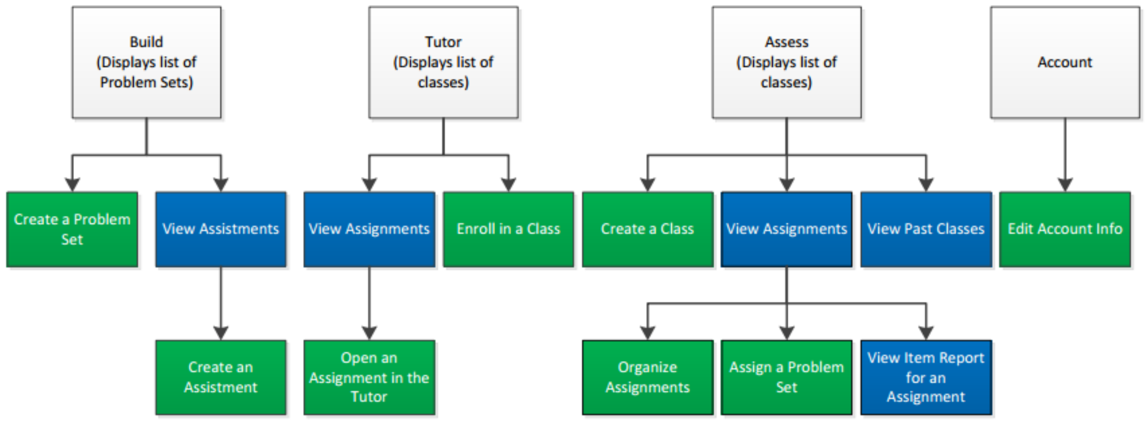
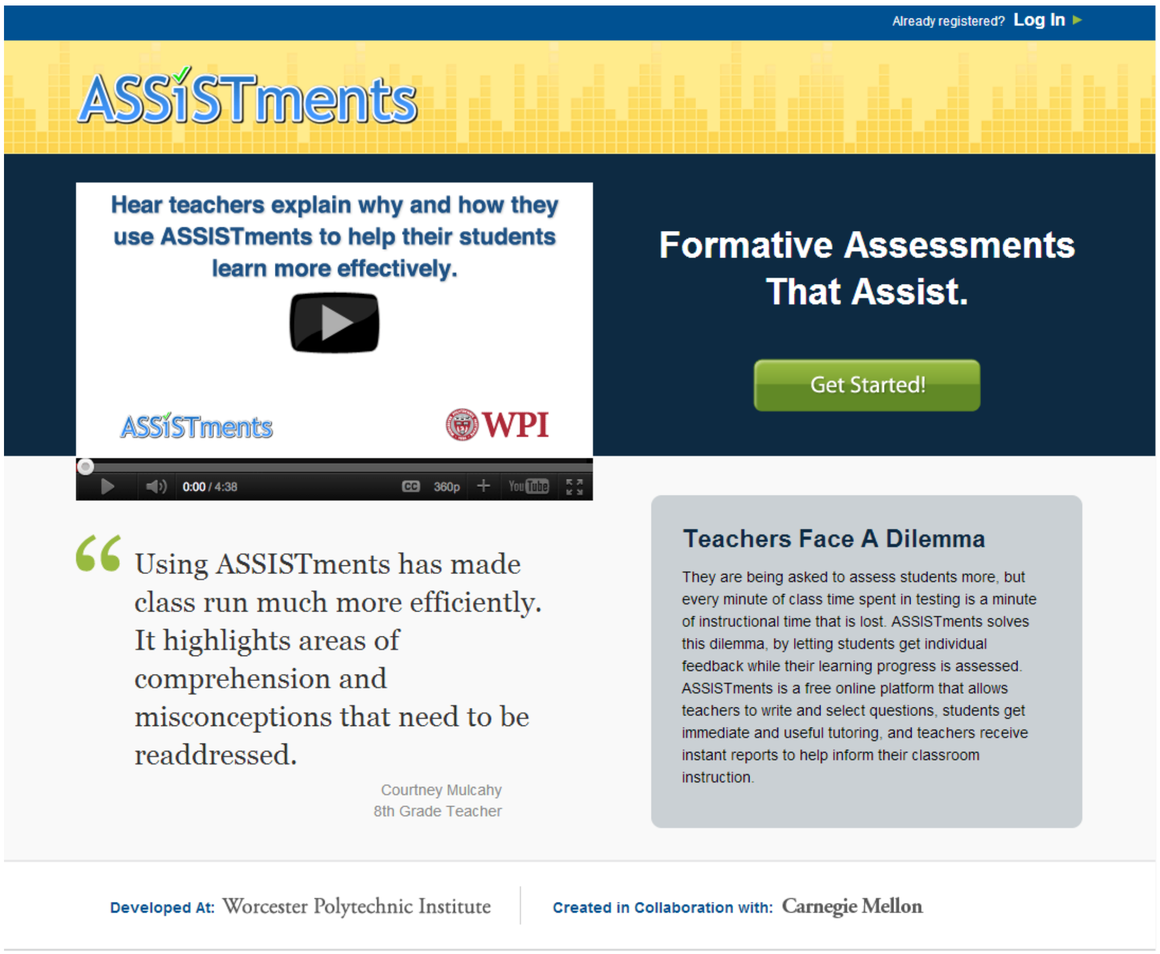


Figure 3.3: Outline of navigation

3.4 Interface Analysis

When a user visits ASSISTments, they are greeted with the home page, an advertising page designed to entice users into creating an account (Figure 3.4). Once a teacher has created an account, they are sent to the account tab (Figure 3.1). The reason for this is so that user's won't forgot their login information (such as their display name). There is a yellow box of text at bottom of the screen with an out-dated screenshot (it is missing the Tutor tab and the subtabs are incorrect) which is supposed to tell the user how to get started. It also references a Wiki-style documentation page. Other than that, there is no indication of what to do. The wording of the tabs feels unclear. Teachers have questioned this wording in the past.


Already registered? Log In

ASSISTments

Hear teachers explain why and how they use ASSISTments to help their students learn more effectively.

Formative Assessments That Assist.

Get Started!

"Using ASSISTments has made class run much more efficiently. It highlights areas of comprehension and misconceptions that need to be readdressed."

Courtney Mulcahy
8th Grade Teacher

Teachers Face A Dilemma

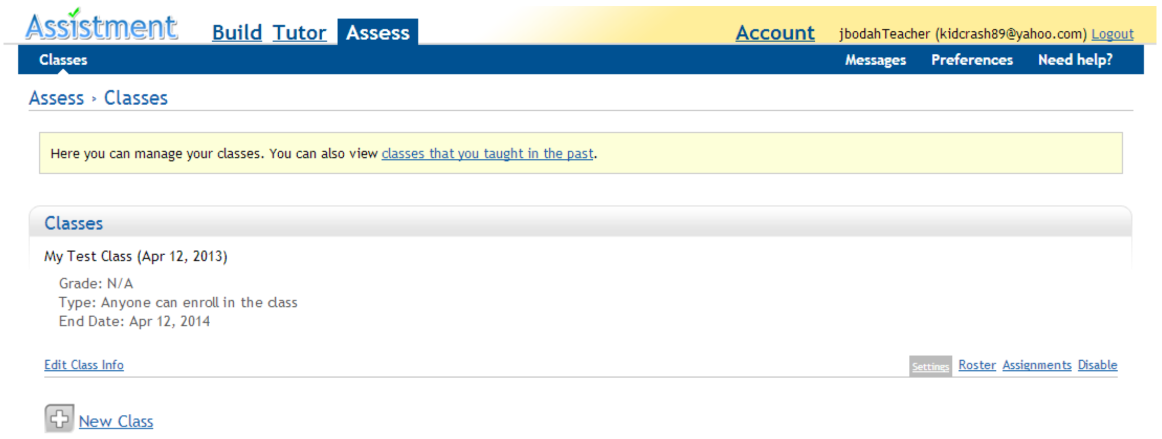
They are being asked to assess students more, but every minute of class time spent in testing is a minute of instructional time that is lost. ASSISTments solves this dilemma, by letting students get individual feedback while their learning progress is assessed. ASSISTments is a free online platform that allows teachers to write and select questions, students get immediate and useful tutoring, and teachers receive instant reports to help inform their classroom instruction.

Developed At: Worcester Polytechnic Institute | Created in Collaboration with: Carnegie Mellon

Figure 3.4: ASSISTments home page

The "TRG-proposed learning sequence" would have a user click the Assess tab. The Build tab is the left-most tab, and Nielsen claims that the left-most part of the screen gets the most attention from users [Nie01b]. This helps explains why 50% of users in 2011-2012 tried to build a problem set.

Supposing a teacher does click the Assess tab, they are sent to a page where they can create a class (Figure 3.2). There is some help dialogue and a link to the user's past classes (even though the user has no classes, past or otherwise). Once a user clicks to create a new class, they are given a form which asks for various class details. Once the form is submitted, the interface dynamically populates the class list with the new class (Figure 3.5). The user can then view their class' roster, view their class' assignment list, or disable the class.



jbodahTeacher (kidcrash89@yahoo.com) Logout

Assess › Classes

Here you can manage your classes. You can also view classes that you taught in the past.

Classes

My Test Class (Apr 12, 2013)

Grade: N/A
Type: Anyone can enroll in the class
End Date: Apr 12, 2014

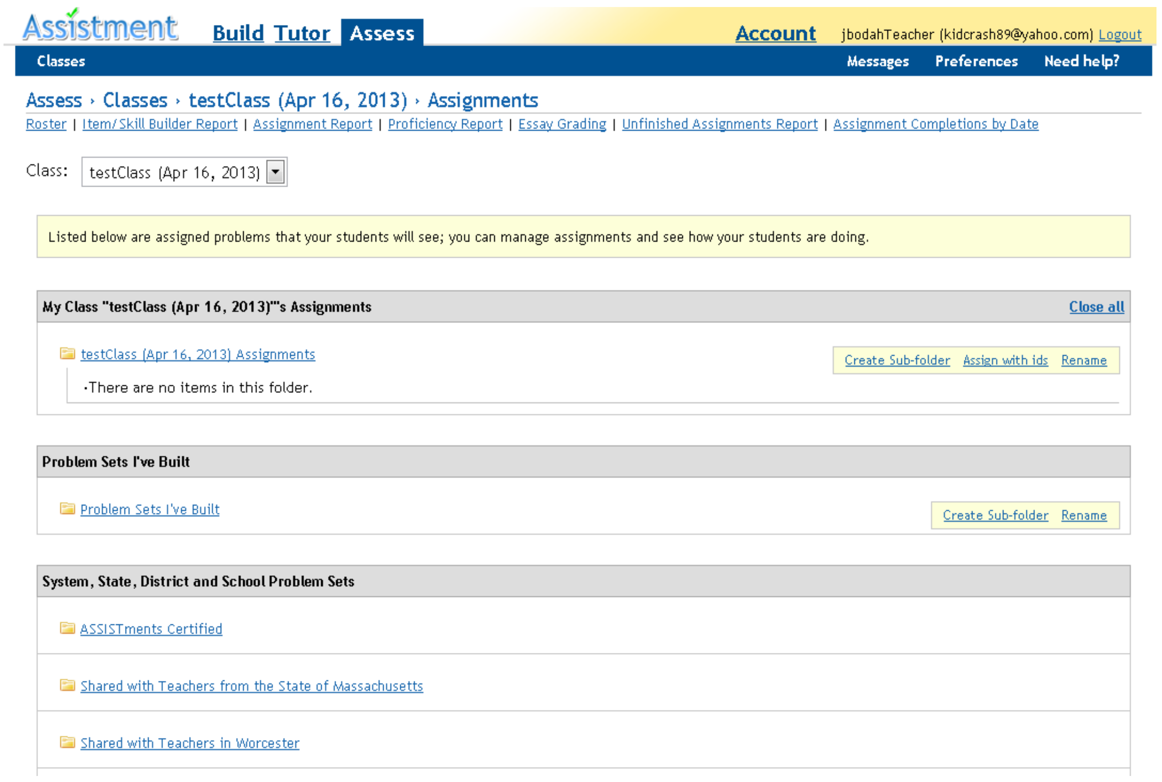
Edit Class Info   Settings Roster Assignments Disable

New Class

Figure 3.5: (Original design) Assess tab with one class

The class roster link is to the far left of the class-specific links. This suggests it will get the most attention [Nie01b]. Additionally, the link furthest to the left is not the link the class' assignment list (which is the next click for the "TRG-proposed learning sequence"). The teacher can also disable the class. One odd thing to note is that the "class [a teacher] taught in the past" are essentially disabled class, so the link to one's past/disabled classes does not fully explain what the page that is linked to will do.

No links seen so far have provided the user any sense that they can have the ability to view a report or assign something. To view links to these features one has to click the Assignments link on the class instance. This link is a crucial navigational component in the "TRG-proposed learning sequence", yet it is embedded in the page rather than being at the top left as is conventional for navigation. The user has no idea these features even exist. Clicking on Assignments link would bring the user to that class' assignments (Figure 3.6).


Assistment
Build Tutor Assess
Account
jbodahTeacher (kidcrash89@yahoo.com) Logout
Classes
Messages Preferences Need help?

Assess › Classes › testClass (Apr 16, 2013) › Assignments

Roster | Item/Skill Builder Report | Assignment Report | Proficiency Report | Essay Grading | Unfinished Assignments Report | Assignment Completions by Date

Class: testClass (Apr 16, 2013)

Listed below are assigned problems that your students will see; you can manage assignments and see how your students are doing.

My Class "testClass (Apr 16, 2013)"'s Assignments — Close all

testClass (Apr 16, 2013) Assignments — Create Sub-folder Assign with ids Rename

·There are no items in this folder.

Problem Sets I've Built

Problem Sets I've Built — Create Sub-folder Rename

System, State, District and School Problem Sets

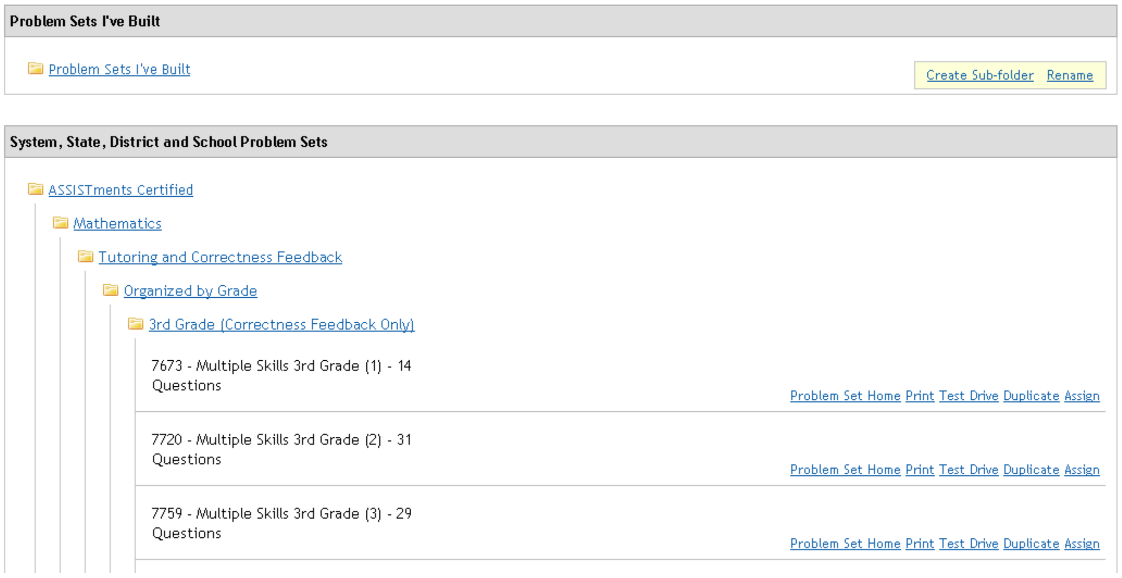
ASSISTments Certified

Shared with Teachers from the State of Massachusetts

Shared with Teachers in Worcester

Figure 3.6: (Original design) Class' assignments page

Once at the class' assignments page, the user is presented with a large number of links: to the roster, various reports, different folder actions, and access to some folders. There's a select box for choosing a class and a rather uninformative text box. The breadcrumbs at the top of the page add a sense of business as well. The added clutter does not seem worth the small amount of navigational support that they provide. Users can't click them, and there are no links that allow you to follow your breadcrumbs backwards. Folders are organized into various containers. Again, the place that users should be clicking based on the "TRG-proposed learning sequence" is not supported by the layout; they are expected to click on the ASSISTments Certified folder and to browse for content. When they find something they are interested in, they can click the Assign link (Figure 3.7). The links are very close together making it hard to distinguish them. The Assign link is all the way to the right, one of the worst spots it could be.



Problem Sets I've Built

Problem Sets I've Built — Create Sub-folder Rename

System, State, District and School Problem Sets

ASSISTments Certified
Mathematics
Tutoring and Correctness Feedback
Organized by Grade
3rd Grade (Correctness Feedback Only)

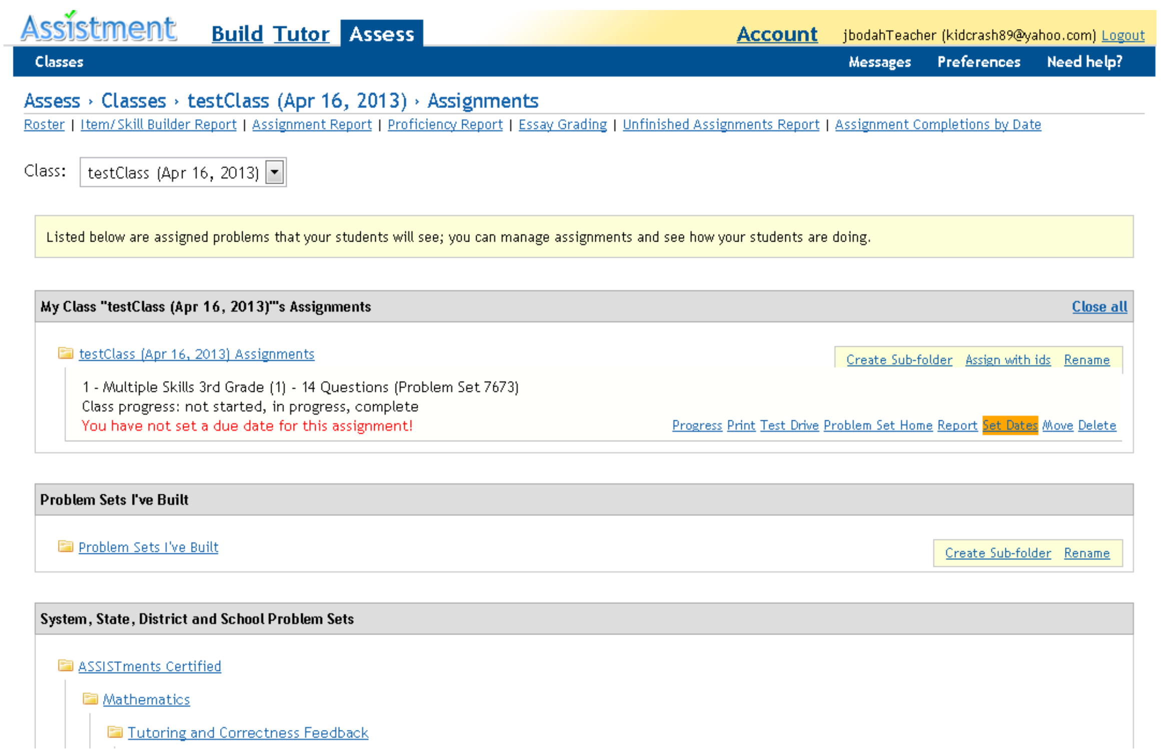
7673 - Multiple Skills 3rd Grade (1) - 14 Questions — Problem Set Home Print Test Drive Duplicate Assign

7720 - Multiple Skills 3rd Grade (2) - 31 Questions — Problem Set Home Print Test Drive Duplicate Assign

7759 - Multiple Skills 3rd Grade (3) - 29 Questions — Problem Set Home Print Test Drive Duplicate Assign

Figure 3.7: (Original design) Class' assignments page, browsing content

After the problem set is assigned, the assignment list at the top of the page gets updated dynamically (Figure 3.8). There is a visual indicator that something was assigned, but it is not very strong. It is very possible that a user might scroll too far and not notice that the assignment list changed. There are several other issues that are similar to those addressed previously regarding links. Additionally, the next thing the user should do according to the "TRG-proposed learning sequence" is click the Tutor tab.


Assistment Build Tutor Assess Account jbodahTeacher (kidcrash89@yahoo.com) Logout

Classes Messages Preferences Need help?

Assess › Classes › testClass (Apr 16, 2013) › Assignments

Roster | Item/Skill Builder Report | Assignment Report | Proficiency Report | Essay Grading | Unfinished Assignments Report | Assignment Completions by Date

Class: testClass (Apr 16, 2013)

Listed below are assigned problems that your students will see; you can manage assignments and see how your students are doing.

My Class "testClass (Apr 16, 2013)"'s Assignments Close all

testClass (Apr 16, 2013) Assignments Create Sub-folder Assign with ids Rename

1 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set 7673)
Class progress: not started, in progress, complete
You have not set a due date for this assignment!
Progress Print Test Drive Problem Set Home Report Set Dates Move Delete

Problem Sets I've Built

Problem Sets I've Built Create Sub-folder Rename

System, State, District and School Problem Sets

ASSISTments Certified
Mathematics
Tutoring and Correctness Feedback

Figure 3.8: (Original design) Class' assignments page, after assigning

The Tutor tab is actually fairly straightforward. The interface does not allow the user to make very many errors. Enrolling in a class and completing an assignment are fairly intuitive as they are the only things one can do. The hardest part of following the "TRG-proposed learning sequence" then is clicking the Assess tab and navigating to the Item Report.

3.5 Conclusions

The site clearly has some big usability issues. Below is a summary of some of the usability issues that were found:

- No landing page after sign-up, application redirects user to Account tab. The user's only visual instruction after signing up is a poorly placed, out-dated image surrounded by a large body of text. There is no overview.

- No explicit home page. Krug claims that the home page is essential for grounding the user and letting them know what their possible actions are and how they can start using the site [Kru06].
- Features seem hidden under layers of navigation.
- Poor wording choice for tabs (e.g. “Assess” vs. “Teacher”). Does not feel like it matches the names of the domain well.
- Link positioning does not support the “TRG-proposed learning sequence”. The “Build” tab, for example, is all the way to the left. Users need to click elements located all over the screen to progress.
- Many pages are very busy and wordy. There is a daunting amount of text at times.
- Links are exposed to the novice user that they don’t need to care about yet (e.g. link to past classes).

While a major refactoring or redesign may seem like the best solution, it would be both expensive to design and implement and problematic for existing users. Krug recommends trying to tweak a user interface rather than trying to perform a large redesign [Kru06]. He recommends trying to look for cheap changes that yield big gains (e.g. hiding things based on a user’s context, removing text, moving links, etc.).

While we could simply jump in and start making changes, we believe that there is a lot of value in waiting until we see users actually stumble on these problems. We will conduct a series of user tests to try and match user errors to usability issues. The issues found during this preliminary analysis will be treated as a bank of possible issues that we can look to if we are unable to determine why a user might be encountering a particular error.

Chapter 4

User Tests

4.1 Overview

Since several usability issues were identified during the preliminary analysis, it seemed reasonable to assume that a new interface design could produce an interface that was easier for teachers to learn. An iterative approach was used to design a new interface following the recommendations of many usability experts [Nie12][Kru10][GL85]. Four user tests were conducted to test a developing ASSISTments interface. That is, changes were made between each user test and tested with the next subject.

Each session consisted of an interview phase, an observation phase, and a post-test discussion. Data was analyzed by several members of the TRG to determine which usability issues were worth addressing, and solutions were proposed. Development occurred between each session and those changes were tested with the next user.

4.2 Methods

All tests were voluntary, and no compensation was provided to the users. In addition, all tests were conducted at the user's place of work (a quiet, after-hours classroom in all cases). Only the test administrator observed the session. He was the only other person in the room with the test subject. Tests took between 30 minutes and 1 hour. Each user was an experienced middle school or high school teacher. All users were regular internet users although some were more confident in their technical skills than others. Users were found from personal connections. Having worked with many teachers over the years, we were able to contact friends and relatives who did not use the system and were willing to sit down with us. All users tested were either high school or middle school teachers and felt comfortable using computers and the Internet.

User tests consisted of three parts. First a scripted introduction was read to the user which concluded with a short interview of the user to understand their background. The user was then observed for 15-30 minutes, all the time being encouraged to follow the "think aloud" protocol [Lew82]. Finally, an informal post-test discussion was conducted as a means of seeing how the user felt after the experience was over and to see if they could further describe what they were feeling during the test. Data consisted of server request logs which tracked where the user clicked on the site and written logs collected by the test administrator to record the user's verbal and emotional responses. User tests were analyzed and changes were made to address specific usability problems. Revisions were then tested in the following user test.

4.2.1 Interview

The interview component of the user test was scripted. Instructions were read aloud to users prior to the observation. The script was based on the one provided by Krug [Kru10]. It explained the purpose of the test, how the user should be honest in their feelings and reactions, and described the think aloud protocol to the user.

After the reading, the user was asked to answer several questions so a background for the user could be established. These questions focused on their experience as a teacher, their experience with the Internet, and their experience using educational software as a whole (whether it be something they use directly with their students or something that supports their teaching, like a grade book).

After these questions, the test moved into the observation phase.

4.2.2 Observation

During the observation phase of the user test, the user was asked to use the system, and the test administrator watched and encouraged the user to follow the think aloud protocol. The test administrator was not allowed to answer questions which might guide the user but was allowed to help the user if they were not uncovering any new usability problems.

The user was first asked to look at the ASSISTments home page. This page is essentially an advertising page which explains ASSISTments in a way to entice teachers to sign-up for an account. The user was asked to explain what they thought they knew about ASSISTments from that page. The user was then assisted in making an account. After an account was made, they were given a general, open-ended scenario to complete:

- Let's assume that you were thinking of using ASSISTments with your Algebra class next week. I'd like for you to setup ASSISTments such that you can do this.

This scenario was chosen to be intentionally vague to see how teachers' reasoning compared to that of the "TRG-proposed learning sequence". Users were then given 15-20 minutes to try to complete this scenario as they see fit. In general, the first 10 minutes were the most useful. Once this time was up and the test administrator felt there was nothing more to learn from the session, the test moved into the post-test discussion phase.

4.2.3 Post-Test Discussion

This phase of the test is where the user was allowed to ask any questions they wanted. The discussion was informal but insightful. The user was told what interface elements were being tested during the session, and the user's reaction upon hearing this was logged. The user was also asked if they could give a general overview of how ASSISTments worked. The purpose of this was to see whether or not they understood the major components of the system and how they were meant to react. It was used as a metric to see whether the user actually knew how ASSISTments could be used to get work done. They were told the "TRG-proposed learning sequence", and their reaction was recorded.

4.2.4 Analysis

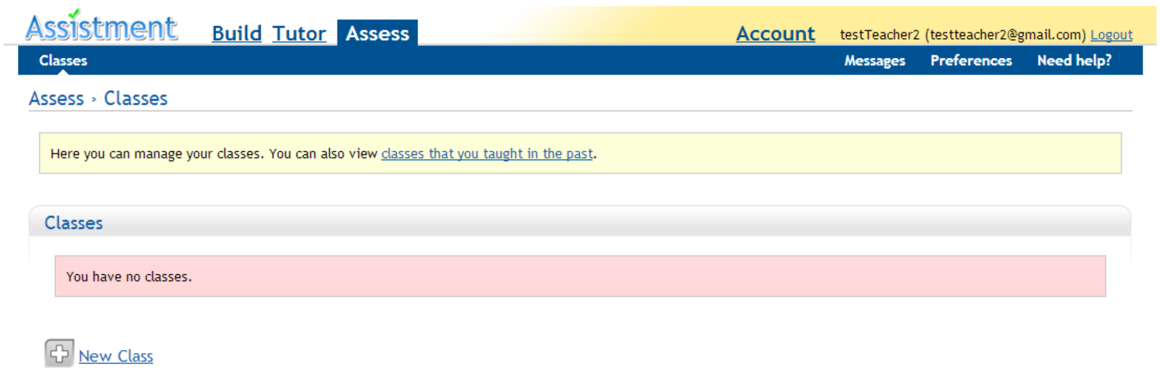
In the analysis phase, notes and server logs were compiled to identify usability issues. Issues were characterized by misunderstanding, frustration, navigation errors, or slow navigation. They were listed and prioritized by several TRG members. Possible solutions were proposed, and subset of the changes were pushed into development based on a balance of predicted effect and difficulty to implement.

4.2.5 Development

Changes in each iteration were limited to those chosen in the analysis phase. The changes were made and deployed on a test server for the next user test. In one case, a paper prototype was used due to time constraints.

4.3 Test 1: Initial Design

The original ASSISTments interface was tested first (As in Figure 4.1. See Figures A.3 to A.7 for all screenshots of the original design). No issues found in the preliminary usability analysis were addressed. The reason for this was to ensure that every usability change could be mapped to a specific observed usability bug.


Assistment
Build Tutor Assess
Account
testTeacher2 (testteacher2@gmail.com) Logout
Classes
Messages Preferences Need help?
Assess › Classes
Classes
New Class

Here you can manage your classes. You can also view classes that you taught in the past.

You have no classes.

Figure 4.1: (Original design) Assess tab

4.3.1 Session Brief

The user tested was a high school Physics teacher of 17 years. She was a regular internet user and had some experience using supportive educational software for comparing test scores.

Reading the home page, she was able to figure out that the site had something to do with giving students assessments. After signing up, the user was redirected to the Account tab. She expressed verbal signs of confusion as to what she should be doing and clicked back and forth between the Build, Assess, and Tutor tabs. After about two minutes of floundering, she decided to start building content. While building, she managed to view some pre-built content but couldn't figure out how to make use of it. After roughly ten minutes of exploring the Build tab, the user – still confused – tried exploring the Assess tab. There she began completing the tasks in the "TRG-proposed learning sequence". She created a class and then immediately accessed her roster. She was greeted with a text blurb which explained how students could enroll in her class and a link to a video which demonstrated how to use the roster page, each of which she glanced over at first and didn't pay much attention to. When she finally did, she figured out that students sign-up for an account on their own. She then went on to assign some pre-built content, and enroll in her own class. The observation was then terminated by the test administrator due to time.

During the post-test discussion, she related that she felt that the navigation was confusing.

4.3.2 Analysis

Analysis revealed several usability problems. First, the user didn't have a clear idea of what the intent of the site was after creating an account. It was clear that ASSISTments had something to do with assessment, but she didn't understand how to begin using the site. The first page she saw was the Account tab which didn't seem very helpful or meaningful. Naturally, her first click was on the Build tab (rather than the Assess tab) due to it's top-left position on the interface. Pages were fairly busy with a lot of text and visual content (horizontal rules, borders, etc). She seemed to have a hard time focusing on the elements that would progress her along the "TRG-proposed learning sequence". She scanned over much of the instructional text embedded in the site without reading it, relying on intuition first. Once she found that too difficult, she gave in and read the instructions. After being presented the scenario, she had little idea of where to start. Additionally, when she finally started exploring the Assess tab, she followed the "TRG-proposed learning sequence" fairly closely, stumbling upon only minor problems along the way.

4.3.3 Development

The goal of this development iteration was to try and make it so that creating a class was the first thing that users did. Once the user in the this test created a class, she progressed quickly. The majority of her time spent with problems was spent being lost before she tried to explore the Assess tab and create a class.

Four major changes were made during this iteration. First, instead of having teachers land in the Account tab after creating an account, they would land in the Assess tab. Relevant account information would be displayed in a flash notice at the top of the Assess tab. This was done to encourage teachers to explore the Assess tab first and to avoid misdirection caused by the Account tab.

Second, many of the pages were made cleaner. Instructional text was removed or shortened, redundant navigation elements were replaced with white space, and information specific to more advanced users was encapsulated and dynamically hidden for new users. Reducing the amount of visual noise on the page should help the important parts of the page (like the link for creating a new class) catch the user's attention.

Third, the navigation tabs were renamed. The original names of "Build", "Assess", and "Tutor" are non-standard to the domain of education. In an effort to make the site more self-explanatory, these were changed to the more standard "Builder", "Teacher", and "Student" respectively.

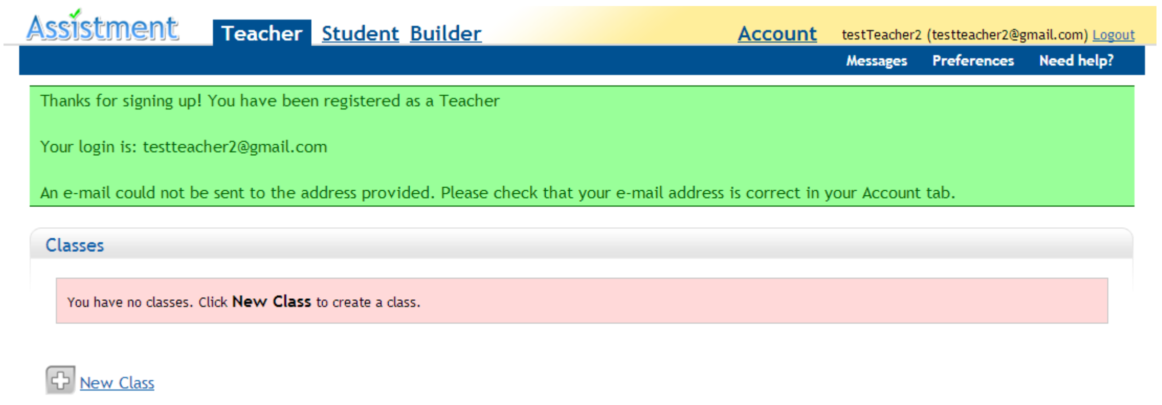
Lastly, the tabs were reordered – "Teacher", "Student", "Builder" – in an effort to place more emphasis on the Teacher tab [Nie01b].

4.3.4 Post-Development Analysis

The changes from this revision were released on the ASSISTments production servers shortly after they were developed. The TRG felt that the interface already seemed much better. The changes from this revision were monitored for one month. 119 teachers created accounts during this month, and 107 (90%) of those teachers created a class. In the previous month, 99 teachers created an account, 74 (75%) of which created a class. Running chi-squared on these values yields a chi-squared statistic of 8.82, or about a 0.3% chance that the change was by chance. It's worth noting that the populations were not sampled from the same distribution, given that some users came from one month and more from another, but these numbers provide at least some support that the development made was an improvement if we assume the populations similar. Note that these numbers are significantly better than the class creation rate of 56% over the 2011-2012 school year described earlier.

4.4 Test 2: Page with Reduced Noise, Changed Vocabulary, and Stronger Redirection

The development from the previous iteration was aimed at getting users to first create a class (Figures 5.1 and 5.2). Thus, one of the goals of this user test was to see whether the actions that users took with the new interface supported these goals.



Thanks for signing up! You have been registered as a Teacher

Your login is: testteacher2@gmail.com

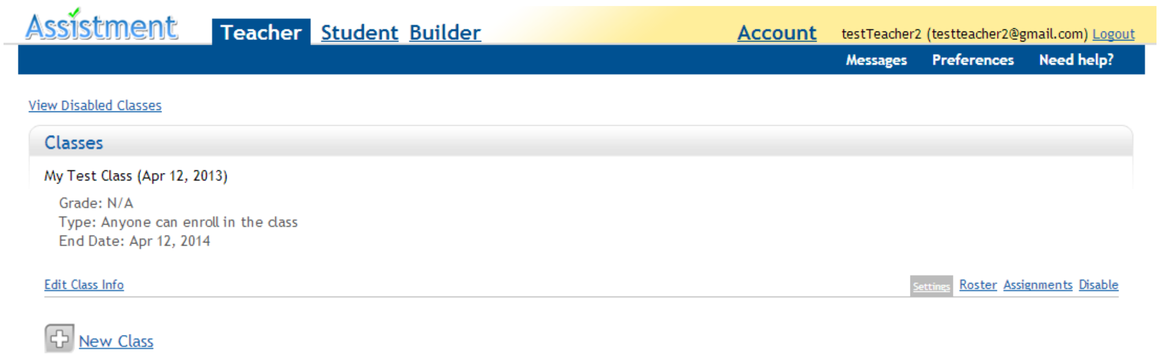
An e-mail could not be sent to the address provided. Please check that your e-mail address is correct in your Account tab.

Classes

You have no classes. Click **New Class** to create a class.

New Class

Figure 4.2: (Iteration 1) Teacher tab



Classes

My Test Class (Apr 12, 2013)

Grade: N/A
Type: Anyone can enroll in the class
End Date: Apr 12, 2014

Figure 4.3: (Iteration 1) Teacher tab with one class

4.4.1 Session Brief

The user had taught high school English for 11 years. She was a regular internet user who mostly used the web for social purposes (e-mail, social media, etc.). She had limited experience with software applications specific to education, but she did use the Internet to help her find content to use in her class. ASSISTments was presented to her school once a year or two ago, but she did not have a clear idea of what it was. She knew it had something to do with getting data on her students.

During the observation, she created an account and then was redirected to the Teacher tab. The scenario was presented, and the user immediately exclaimed that she was unsure of what to do. She began by creating a class and then examined her roster. She was curious how she got students into her class and was under the impression that she had to enter her students' names. While there was some instructional text and a link to a video, it was mostly ignored. To save time, the test administrator intervened and told her that she didn't need to worry about her roster and asked her to assume that her students were already in her class. She was instructed to go back to her class list. She then got very lost; she went into the Student tab and enrolled in her own class. After that, she went into the Builder tab and started creating a problem set. There was lots of terminology used on the site that the user was asking about. She created a problem set and then went into her Student tab to see her class's assignments. She was surprised that the problem set she created wasn't already assigned to her class. The user seemed very frustrated, so the test administrator decided to end the observation.

In the discussion after the test the user immediately related that she had no idea what she was doing and claimed that she would have given up had she tried this on her own. When asked to explain the main use of the site, her response was full of gaps of confusion and misunderstanding.

4.4.2 Analysis

There were a number of takeaways from this user test. So far the changes from the first user test seemed to be working; the first thing that the teacher did was create a class. Prior to this however, she explicitly noted that she didn't know what it was that she should be doing. This set the tone for the entire session. At no point did it seem like she had any idea of how she would complete the scenario. Some sort of lightweight navigational recommendations or a quick overview of the major components of the site could have gone a long way towards helping the user. Instead, she did her best to do what she thought was logical by creating content. She never realized that there was already a large bank of pre-built content which she could make use of.

Another thing worth mentioning is that she was confused by some of the terminology in the Builder tab. Since the Builder is not a part of the "TRG-proposed learning sequence", this issue was left to be dealt with at a later date.

The user also, like the user from the first session, was interested in knowing about how to fill her roster after creating a class. She was under the impression that she had to submit the names of her students in some kind of form when the actual answer was that her students needed to make accounts and enroll in her class.

4.4.3 Development

The biggest issue was that the user didn't know what to do. To address this issue, context-sensitive tooltips which showed the user where to click were developed and embedded on pages throughout ASSISTments. The tooltips were designed to walk a user through the "TRG-proposed learning sequence". Different tooltips would show up based on what the user had already done. The purpose was to give the user direction without restricting the their freedom to explore.

Several other usability problems were noted and solutions were proposed, but they were left out of this development cycle to isolate the changes.

4.5 Test 3: Context-sensitive Embedded Tooltips

The interface in this test had context-sensitive tooltips embedded in it (As in Figure 4.4. See Figures A.10 to A.17 to see all changes from Iteration 2). The tooltips pointed to the links that walked a user through the“TRG-proposed learning sequence”. The system made queries about the user’s state and displayed the appropriate support. Each tooltip had five to eight words explaining what the user was clicking the link for.

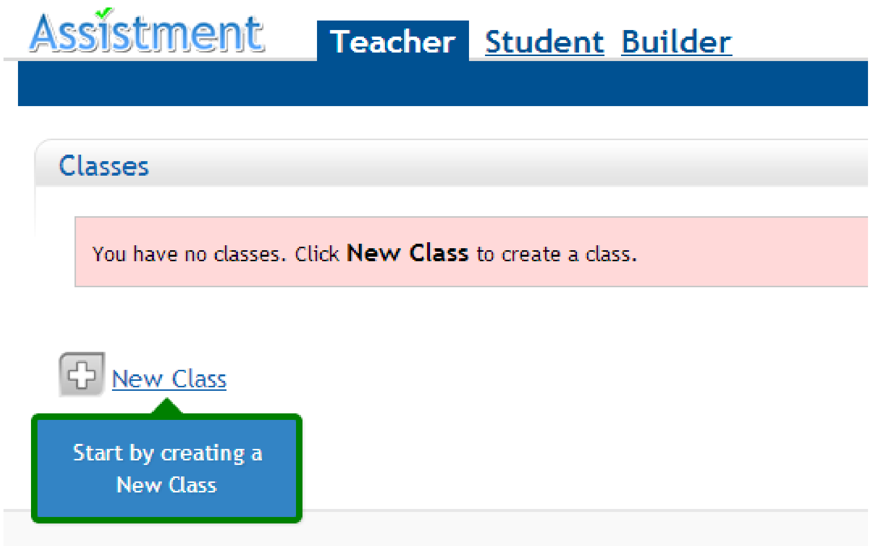

Assistment
Teacher
Student
Builder
Classes
You have no classes. Click New Class to create a class.
New Class
Start by creating a New Class

Figure 4.4: (Iteration 2) Create a class tooltip

4.5.1 Session Brief

The user in this test had 13 years of experience as a high school Math teacher. He was a very experienced internet user with a background working with computer systems. He used several computer applications which assisted him in outlining his lesson plans, but otherwise had limited involvement with educational software. ASSISTments was presented at his school once, but he never tried using it and wasn’t sure how it worked.

Upon viewing the home page, he figured that ASSISTments was a tool that could be used to show evidence of learning. The user created an account and was read the scenario. He created a class and then viewed his roster. Note that he did not follow the tooltip which asked him to assign some content and instead followed his own intuitions. The test administrator let the user explore the roster for a brief period of time and then asked him to assume that enrolling his students was not an issue. The user then continued to follow the tooltips by viewing his class's assignments. The user browsed the pre-built content and then assigned something. After assigning something, he wasn't sure of what to do next. He had scrolled down past the view of the tooltip which asked him to enroll in his own class as a student, so the test administrator asked him to scroll up. The user noticed that the problem set he assigned was now listed in his class's assignments. He also saw the tooltip which encouraged him to enroll in his own class, but he was very adverse to the idea of clicking it. He didn't understand why the system wanted him to be a student. He felt that he should not be doing this and likened the tooltips to "carrot dangling". Though reluctant, he followed the tooltips to their completion which eventually led him to his item report. After allowing the teacher to view his item report for a short time, the test administrator ended the observation and transitioned into the discussion phase of the test.

In the discussion the user was able to relate the major components of the system and stated that the system seemed to be a "substitute for paper-and-pencil". The test administrator encouraged the user to talk about his reluctance in following the tooltips regarding clicking the Student tab. In his response he explained that he didn't understand why he would act as a student when he is really a teacher. He was told that the intent was two-fold: so that he could experience what his students would experience and so he could generate data which he could then view in an item report. His reaction to the idea was still unfavorable, and alternatives such as an ability to generate fake student data were discussed.

4.5.2 Analysis

In general, this user test went much better than the others. Although the user did still need some guidance from the test administrator, the user did complete the major tasks which deemed a user proficient.

The user got started by creating a class as intended in the task sequence. Like other teachers, he was then immediately interested in how students got populated in his class. The site clearly needed to address to teachers how exactly students got enrolled in a class. Also, the tooltips didn't communicate to the user effectively when he scrolled too far after assigning something. Development was needed to address the issue of scrolling.

Still, the TRG felt that the biggest issue happened when the interface instructed the user to enroll in their own class as a student. The user felt the urge to resist the system's instruction. This could mean one of two things: 1) the "TRG-proposed learning sequence", which centered around teachers exploring the system through the eyes of a student to generate data and then viewing that data in a report, does not align with the actual goals of some teachers, or 2) the interface did not express the value in seeing the system as a student well enough to the teacher.

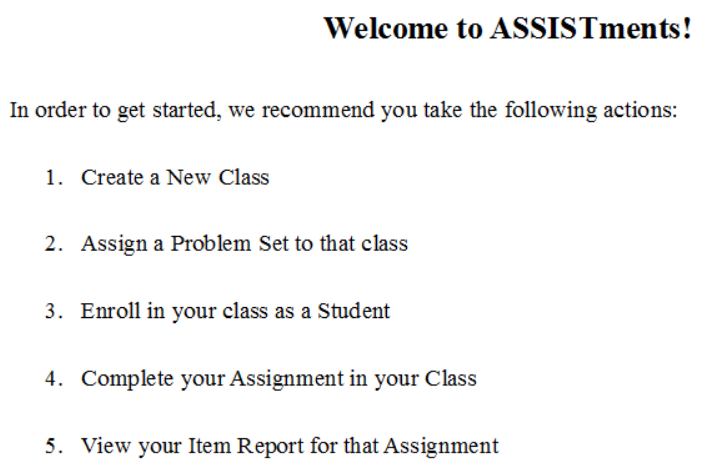
Whether or not teachers care enough about knowing what their students will see when they use the ASSISTments (that is, to the point that they would want to be instructed to act as a student in their own class) is an open question. The TRG still believes that showing teachers a report with data in it in their first few interactions with the system is crucial. However, this is an open question as well. If seeing a report in the first interactions is worthwhile, but encouraging teachers to be a student in their own class is not, then a different way for generating report data (such as having the user click a button to generate some sample data) may be required.

4.5.3 Development

Due to time constraints, the only development made during this iteration was the preparation of a paper prototype which gave an overview of the steps of the "TRG-proposed learning sequence". The hope was that this would help to explain the reasoning as to why the tooltips were directing the user the way they were. Rather than "carrot dangling" the user blindly through the interface, this change was intended to help give the user a better sense of how the system is supposed to work and to help establish a fundamental understanding of the interaction that a user should expect to have with the system when using it. This document would be handed to the next user subject after they created an account. It would simulate a splash screen which would appear after creating an account and before allowing the user to interact with ASSISTments.

4.6 Test 4: Paper Prototype Landing Page with Site Overview

In this test the user was given a paper prototype after creating an account. It briefly outlined the basic steps that the ASSISTments team expected the user to complete (Figure 4.5).


Welcome to ASSISTments!

In order to get started, we recommend you take the following actions:

1. Create a New Class
2. Assign a Problem Set to that class
3. Enroll in your class as a Student
4. Complete your Assignment in your Class
5. View your Item Report for that Assignment

Figure 4.5: (Iteration 3) Paper prototype of landing splash

4.6.1 Session Brief

This teacher had taught middle school Math for 6 years and had worked previously for an insurance company. She was a regular internet user and used the Internet for news, social media, and e-commerce. She also used it as a reference to find math worksheets and games. She had previous experience using tutoring applications other than ASSISTments. Another teacher in her school used ASSISTments regularly, so she was already familiar with the purpose of ASSISTments. She attended a short introduction class to ASSISTments a few years ago, but claimed she did not know anything about the system. Still, she was able to correctly answer specific questions regarding the organization of the system. Because of her experience, her input had to be taken with some skepticism.

The user was instructed to create an account and then was handed the paper prototype. She spent a minute talking out loud about the paper. She related that she did not understand what it meant when it said your “Enroll in your own class as a Student”. When she was finished, she handed the paper to the test administrator.

She did not ask for it again. She began by creating a class and then – ignoring the tooltip which tried to instruct her to view her assignments – navigated to view her roster. She spent some time looking for a way to enroll students in her class for about a minute. She was instructed to assume that her students were already enrolled. After that, the user returned to her class list and then continued to view her class's assignments as the tooltips said to do. She began looking through the pre-built content and assigned something. Then she began organizing her class assignments into folders. She spent some time doing this and then was asked to assume that she had enough content and organization. The test administrator asked her what she felt needed to do next to which she did not know. There were no visual cues as to how to proceed, so the user was asked to scroll up. The user assumed that the administrator wanted her to keep following the tooltips, which she did until the end of the test.

During the post-test discussion, the user was told what was being tested (the paper prototype). She seemed a little surprised by this. She was also asked to elaborate as to why she clicked into her roster after creating a class instead of following the tooltips. She claimed that she thought she could learn the system without them and didn't want to use them. She also commented that she found them a bit intrusive.

4.6.2 Analysis

Overall, the paper prototype seemed to have little effect. It could be that the paper was too disconnected from the digital user interface, but it would also seem reasonable to think that she had simply forgotten what she was supposed to be doing. There were not enough interface elements to point her where she needed to go every time she was expected to do something. She did not retaliate to the idea of being a student as much as the user in the previous session, but she did claim that she was a little bit confused as to why she was being a student. Having had an introduction to the system previously didn't seem to affect her task completion very much, but she did seem notably more comfortable than previous users.

4.6.3 Development

Following this session, the wording of many tooltips were changed, and several tooltips were added to explain how users could disable them, how students enrolled in the user's class on their own, and an overlay was added on the assignment page to tell the user to scroll up after assigning something. The colors of the tooltips were changed to further distinct them from the rest of the interface. A splash containing a reworded version of the paper prototype was designed and implemented to pop up after a user created an account.

Chapter 5

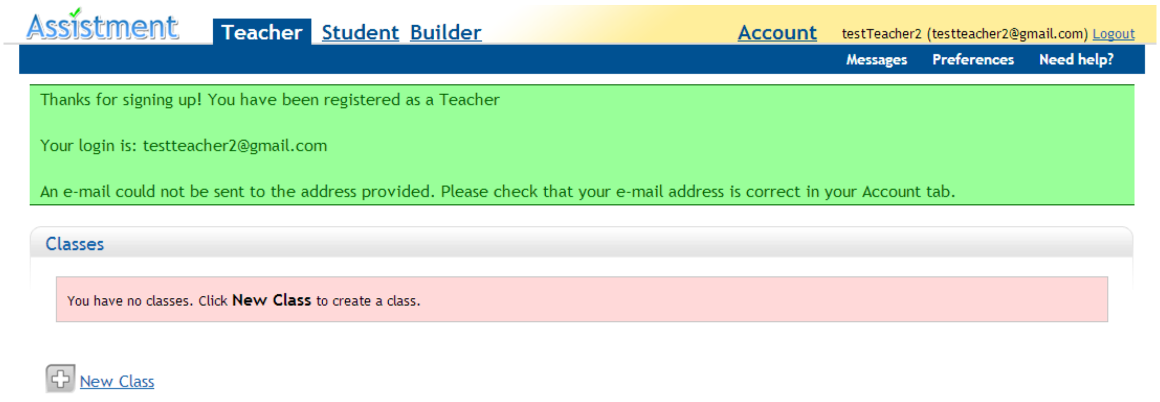
Controlled Experiment for Embedded Support

5.1 Overview

A controlled experiment was conducted on the ASSISTments production server to see whether or not having an embedded support system could have an effect on the task completion rates of new users.

5.2 Methods

There were 69 teachers who created new accounts in ASSISTments over a two week span that were exposed to one of two interfaces. These teachers were randomly assigned to one of two conditions. These teachers were exposed to either a control interface (Figures 5.1 and 5.2 from Iteration 1), which contained all of the development from the first user test, or a test interface (As in Figures 5.3 and 5.4 from Iteration 4. See Figures A.19 to A.30 for all changes from Iteration 4), which had all of the changes from each of the user tests. Teachers did not see both interfaces. In particular, the experiment was comparing the effectiveness of the embedded support system and landing splash page described previously.



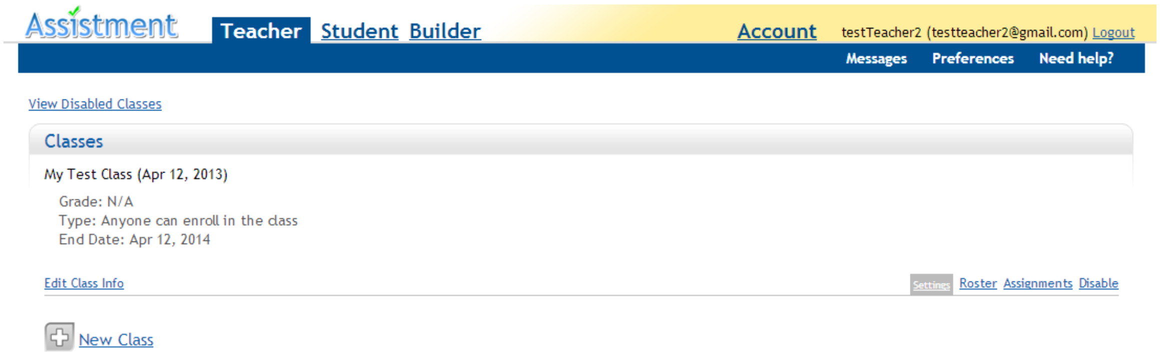
Thanks for signing up! You have been registered as a Teacher

Your login is: testteacher2@gmail.com

An e-mail could not be sent to the address provided. Please check that your e-mail address is correct in your Account tab.

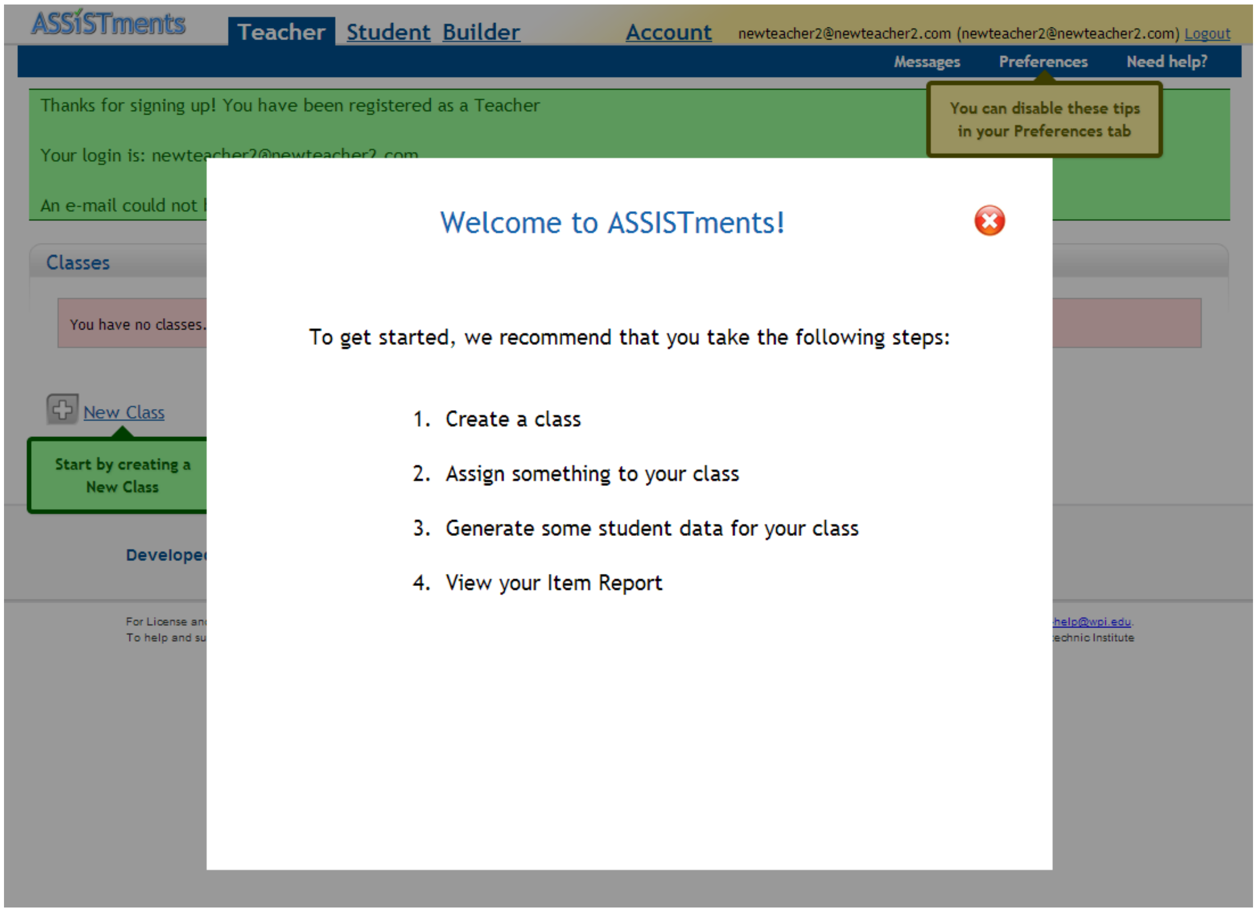
You have no classes. Click **New Class** to create a class.

Figure 5.1: (Iteration 1) Teacher tab



My Test Class (Apr 12, 2013)
Grade: N/A
Type: Anyone can enroll in the class
End Date: Apr 12, 2014

Figure 5.2: (Iteration 1) Teacher tab with one class



newteacher2@newteacher2.com (newteacher2@newteacher2.com)

Thanks for signing up! You have been registered as a Teacher

Your login is: newteacher2@newteacher2.com

An e-mail could not

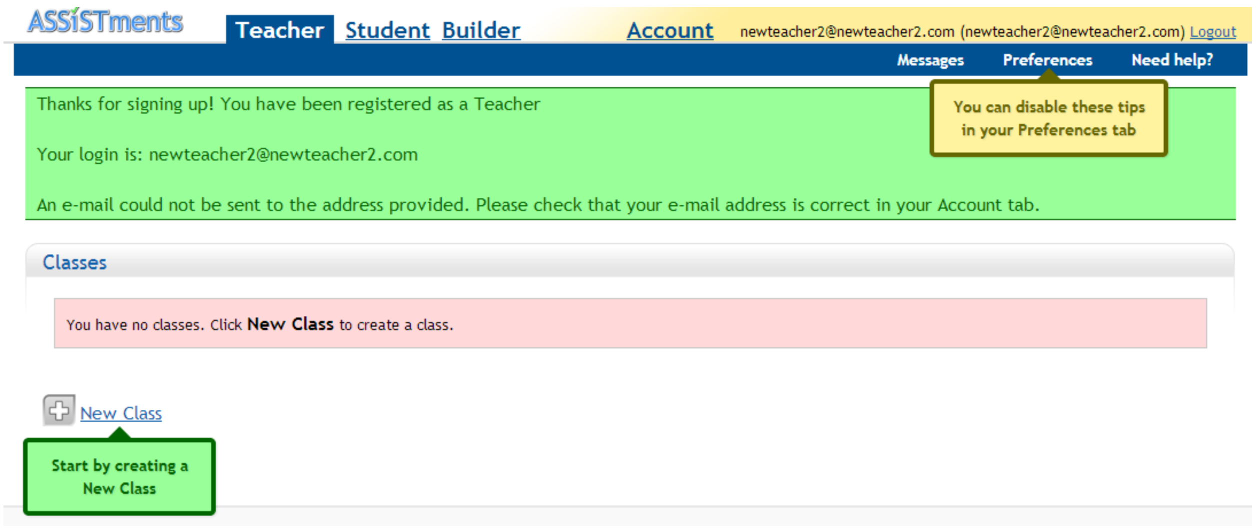
You have no classes.

Welcome to ASSISTments!

To get started, we recommend that you take the following steps:

1. Create a class
2. Assign something to your class
3. Generate some student data for your class
4. View your Item Report

Figure 5.3: (Iteration 4) Landing splash


ASSISTments
Teacher
Student
Builder
Account
newteacher2@newteacher2.com (newteacher2@newteacher2.com) Logout
Messages
Preferences
Need help?
You can disable these tips in your Preferences tab

Thanks for signing up! You have been registered as a Teacher

Your login is: newteacher2@newteacher2.com

An e-mail could not be sent to the address provided. Please check that your e-mail address is correct in your Account tab.

You have no classes. Click **New Class** to create a class.

New Class
Start by creating a New Class

Figure 5.4: (Iteration 4) Create a class and disable explanation tooltips

Task completion rates were the primary metric analyzed. Fourteen different teacher tasks were analyzed:

1. Creating a class
2. Enrolling 1+ students in a class (not including self)
3. Enrolling 5+ students in a class
4. Enrolling in own class
5. Assigning content
6. Assigning ASSISTments certified content
7. Having students complete 1+ problems
8. Having students complete 5+ problems
9. Having students complete 1+ assignments
10. Having students complete 5+ assignments

11. Building a problem set

12. Building an assistment

13. Viewing a report with data

14. Viewing their roster

5.3 Results

69 teachers created accounts over the experimental period, 35 which were in the control condition and 34 which were in the experimental condition. 4 teachers were removed from the control group and 4 from the test group because they did not use ASSISTments enough to reach either of the interfaces. Thus, the control condition had 31 teachers and the test condition had 30. Table 5.1 shows the raw task completion counts and rates for each of the conditions.

5.4 Analysis

Chi-squared tests using 2 x 2 contingency tables were conducted for each of the fourteen tasks to test if there was a significant difference in the completion rate for that task. A summary of these calculations can be seen in Table 5.2. Most of the tasks had p-values above 0.05, but the task of viewing a roster had a smaller p-value of 0.048. Teachers in the control condition (i.e. no embedded support) were almost twice as likely to view their roster page than those in the test condition with a p-value of 0.048.

Task	**Control Count**	**Control Rate**	**Test Count**	**Test Rate**
Created a class	30	97%	30	100%
Enrolled 1+ students	10	32%	7	23%
Enrolled 5+ students	8	26%	7	23%
Enrolled in own class	12	39%	12	40%
Assigned content	17	55%	14	47%
Assigned certified content	12	39%	10	33%
Completed 1+ problems	6	19%	7	23%
Completed 5+ problems	6	19%	7	23%
Completed 1+ assignments	5	16%	6	20%
Completed 5+ assignments	5	16%	4	13%
Built a problem set	10	32%	12	40%
Built an assistment	7	23%	4	13%
Viewed a report with data	9	29%	7	23%
Viewed a roster	24	77%	16	53%

Table 5.1: Task completion data for controlled experiment

Task	X^2	p
Created a class	0.984	0.321
Enrolled 1+ students	0.604	0.437
Enrolled 5+ students	0.050	0.823
Enrolled in own class	0.011	0.918
Assigned content	0.407	0.523
Assigned certified content	0.191	0.662
Completed 1+ problems	0.144	0.704
Completed 5+ problems	0.144	0.704
Completed 1+ assignments	0.155	0.694
Completed 5+ assignments	0.095	0.758
Built a problem set	0.396	0.529
Built an assistment	0.882	0.348
Viewed a report with data	0.256	0.613
Viewed a roster	3.918	0.048

Table 5.2: Chi-squared of task completion data

5.5 Discussion

This experiment has yielded several interesting results. Perhaps the biggest result is that there is no significant difference between the conditions on any substantial tasks (e.g. creating a class, enrolling students, assigning something, viewing a report with data). Although the intent of the embedded support was to help people get more things done, it seems that it had little effect in this regard. However, it could still be true that the embedded support has increased user satisfaction or a reduced error rate.

The support might have been ineffective for several reasons. Users might have felt the desire to rebel against the walkthrough. This occurred during the user tests. Users could have been following outside instructions (say from a partnering site). This seems unlikely due to the timespan and the difference in users' geographic locations. It could also be that the support just wasn't enough to help users.

Although there was little difference in most tasks, there was a significant difference in whether or not users viewed their roster page. Teachers who had the embedded support enabled did not view the roster page as frequently as those who did not have the support. The roster led to a lot of confusion throughout the user tests that were performed earlier. The users who got the embedded support got several pieces of information which might have influenced this: 1) a tooltip which briefly explained how the roster tab worked; 2) another tooltip which directed them to go and assign something; and 3) a splash screen that users saw after registering which told them that students need to enroll in a class (as opposed to teachers enrolling students in their classes via a form). The control condition saw none of these. It could be that more teachers understood the purpose of the roster page thanks to these tooltips, so they did not feel the need to view it. All of the teachers who participated in the user tests thought they had to enter their students' names in a form (which isn't the paradigm that ASSISTments uses).

Lastly, the data shows that a large portion of users from both groups are creating classes. This further supports the changes made from the first development iteration which were focussed on encouraging more users create a class. Recall that these changes consisted of simple changes like renaming links, reducing visual noise, and moving things on the page.

Chapter 6

Summary & Conclusions

6.1 Summary of Results

From our experimentation and analysis, we noticed several things. User testing was useful in the identification of usability issues. A few small changes based on modern interface design principles were able to significantly increase the number of users who were creating a class. Further user testing uncovered larger, more serious issues which led to the development of an embedded support system. This system did have some influence on how users navigated the site, but it did not have a significant influence on users' task completion rates.

6.2 Conclusions

First, while embedded support can influence how a user uses a system, we believe that trying to an embedded support system to help a user navigate a poorly designed user interface is fruitless. Users had no significant differences of task completion rates on important tasks when using an embedded support system.

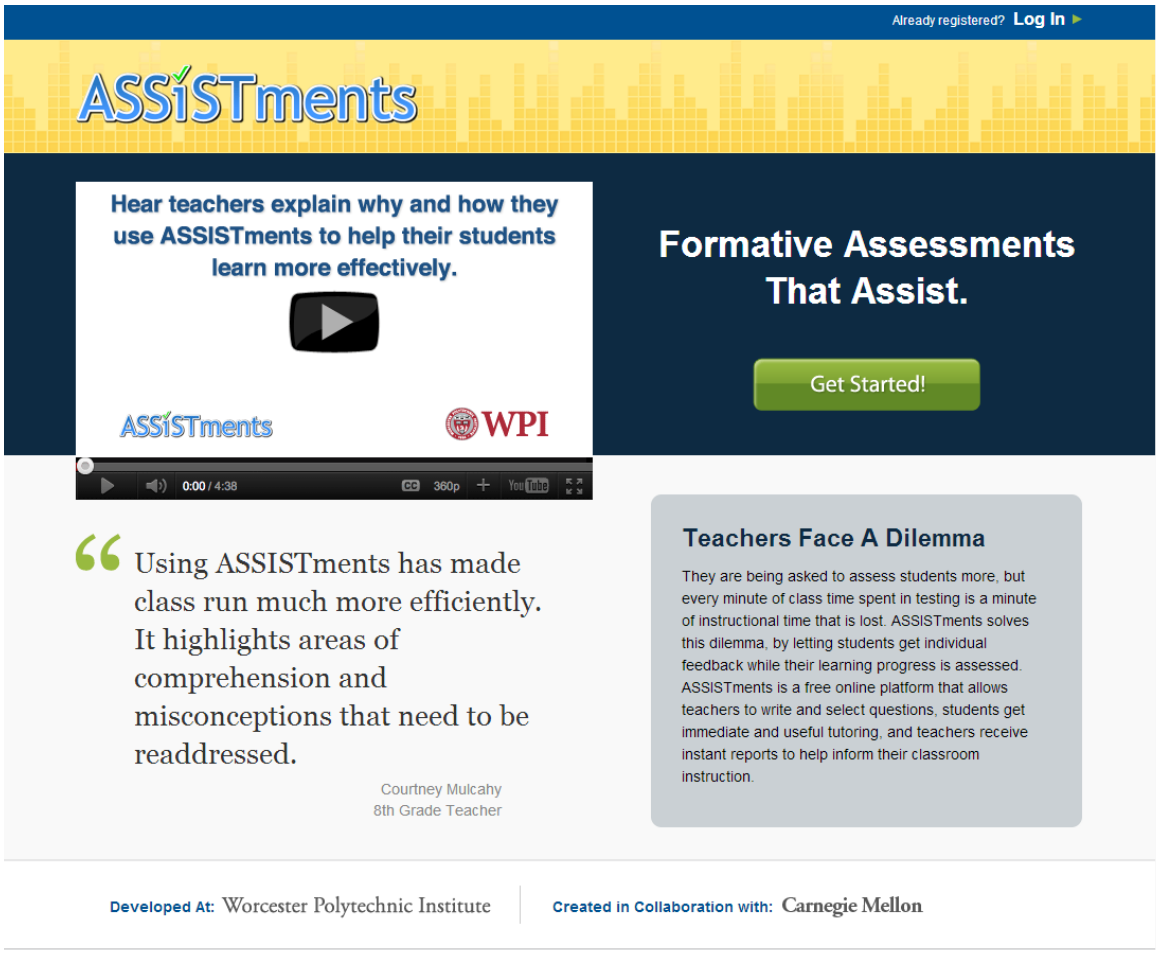
Additionally, we found that we were able to get big differences from small changes, such as rewording links and hiding or moving interface elements. These simple changes led to significant increases in the number of teachers who were creating a class.

Lastly, we support the ideas that usability testing and iterative development are crucial to effective interface design. Our user tests helped us identify issues and gave us a better understanding of our users. An iterative development cycle allowed us to get feedback quickly and tweak the interface. Finally, running a controlled experiment showed us that our preconceptions of how an interface will be used are not accurate. We believe that the embedded support system would at least show some improvement to task completion, and we were surprised when that was not the case.

Appendix A

Interface Screenshots

Below are some screenshots of the ASSISTments interface. The home page was used in every user test. The original interface design was used in the first user test. Each other user saw an iteration (e.g. the user in user test 2 saw iteration 1, etc). The controlled experiment compared iteration 1 (control condition with no embedded support) with iteration 4 (test condition with landing page and context-sensitive tooltips).



Already registered? Log In

ASSISTments

Hear teachers explain why and how they use ASSISTments to help their students learn more effectively.

Formative Assessments That Assist.

Get Started!

“Using ASSISTments has made class run much more efficiently. It highlights areas of comprehension and misconceptions that need to be readdressed.

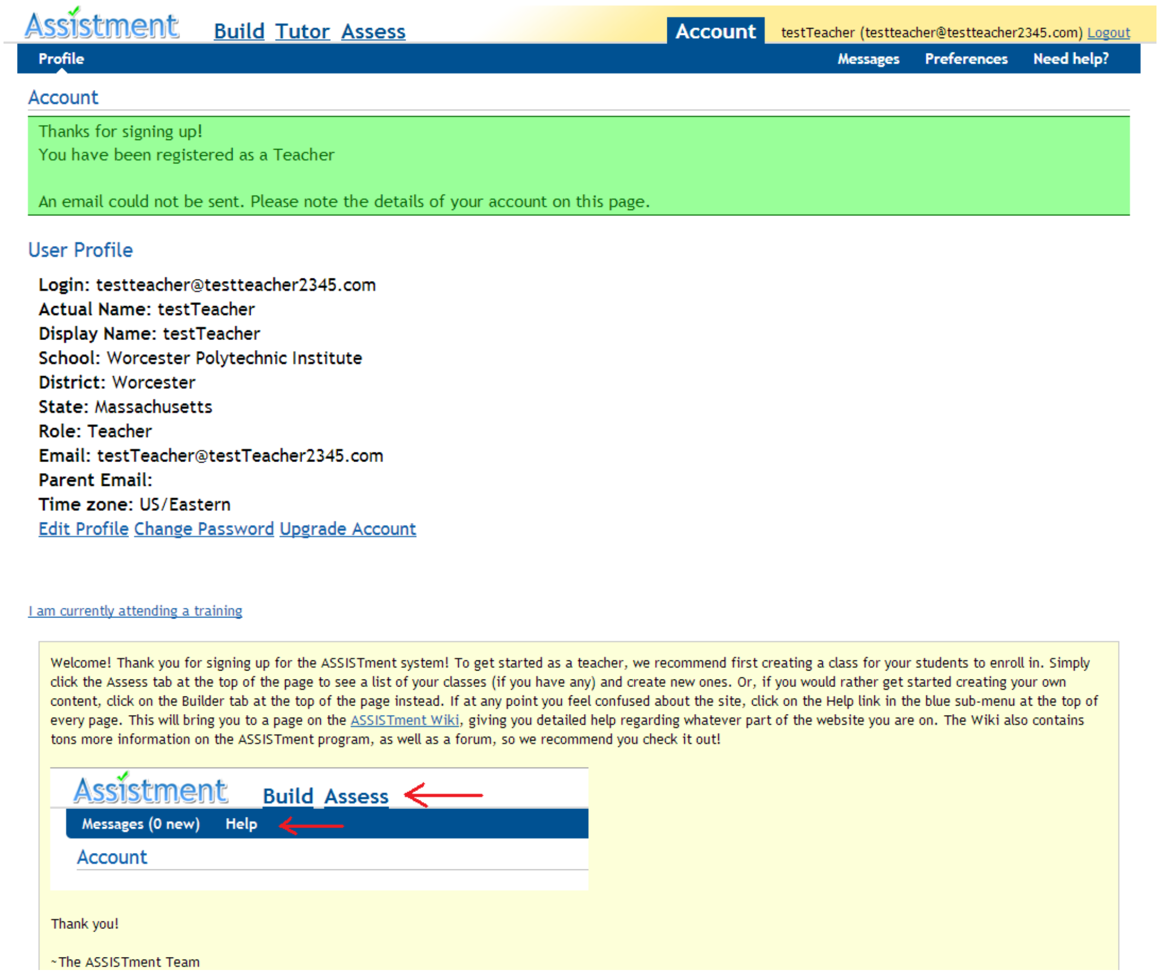
Courtney Mulcahy
8th Grade Teacher

Teachers Face A Dilemma

They are being asked to assess students more, but every minute of class time spent in testing is a minute of instructional time that is lost. ASSISTments solves this dilemma, by letting students get individual feedback while their learning progress is assessed. ASSISTments is a free online platform that allows teachers to write and select questions, students get immediate and useful tutoring, and teachers receive instant reports to help inform their classroom instruction.

Developed At: Worcester Polytechnic Institute | Created in Collaboration with: Carnegie Mellon

Figure A.1: ASSISTments home page


Assistment
Build Tutor Assess
Account
testTeacher (testteacher@testteacher2345.com) Logout
Profile
Messages Preferences Need help?

Account

Thanks for signing up!
You have been registered as a Teacher

An email could not be sent. Please note the details of your account on this page.

User Profile

Login: testteacher@testteacher2345.com
Actual Name: testTeacher
Display Name: testTeacher
School: Worcester Polytechnic Institute
District: Worcester
State: Massachusetts
Role: Teacher
Email: testTeacher@testTeacher2345.com
Parent Email:
Time zone: US/Eastern
Edit Profile Change Password Upgrade Account

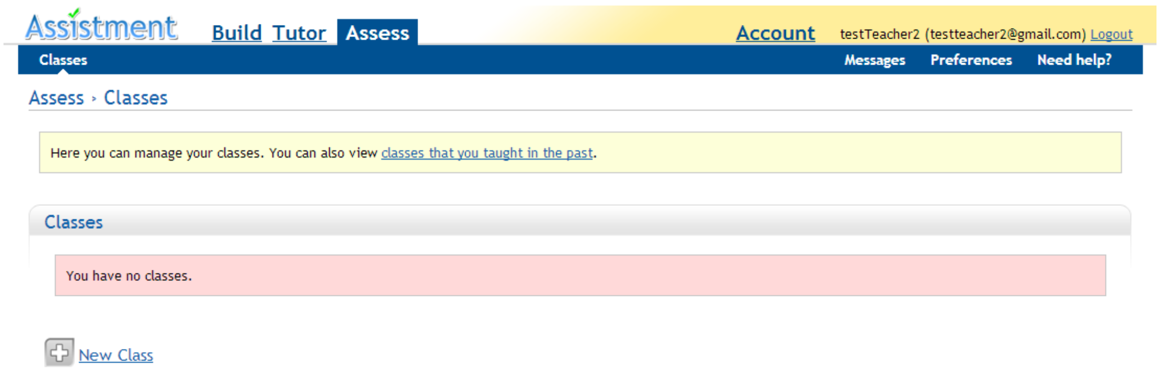
I am currently attending a training

Welcome! Thank you for signing up for the ASSISTment system! To get started as a teacher, we recommend first creating a class for your students to enroll in. Simply click the Assess tab at the top of the page to see a list of your classes (if you have any) and create new ones. Or, if you would rather get started creating your own content, click on the Builder tab at the top of the page instead. If at any point you feel confused about the site, click on the Help link in the blue sub-menu at the top of every page. This will bring you to a page on the ASSISTment Wiki, giving you detailed help regarding whatever part of the website you are on. The Wiki also contains tons more information on the ASSISTment program, as well as a forum, so we recommend you check it out!

Thank you!

~The ASSISTment Team

Figure A.2: (Original design) Account tab


Assistment
Build Tutor Assess
Account
testTeacher2 (testteacher2@gmail.com) Logout
Classes
Messages Preferences Need help?

Assess › Classes

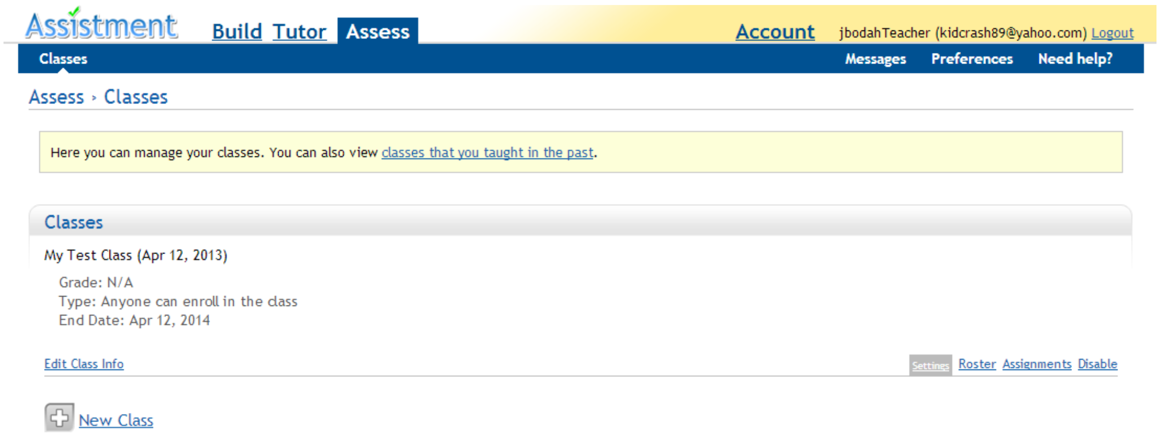
Here you can manage your classes. You can also view classes that you taught in the past.

Classes

You have no classes.

New Class

Figure A.3: (Original design) Assess tab



jbodahTeacher (kidcrash89@yahoo.com) Logout

Assess › Classes

Here you can manage your classes. You can also view classes that you taught in the past.

Classes

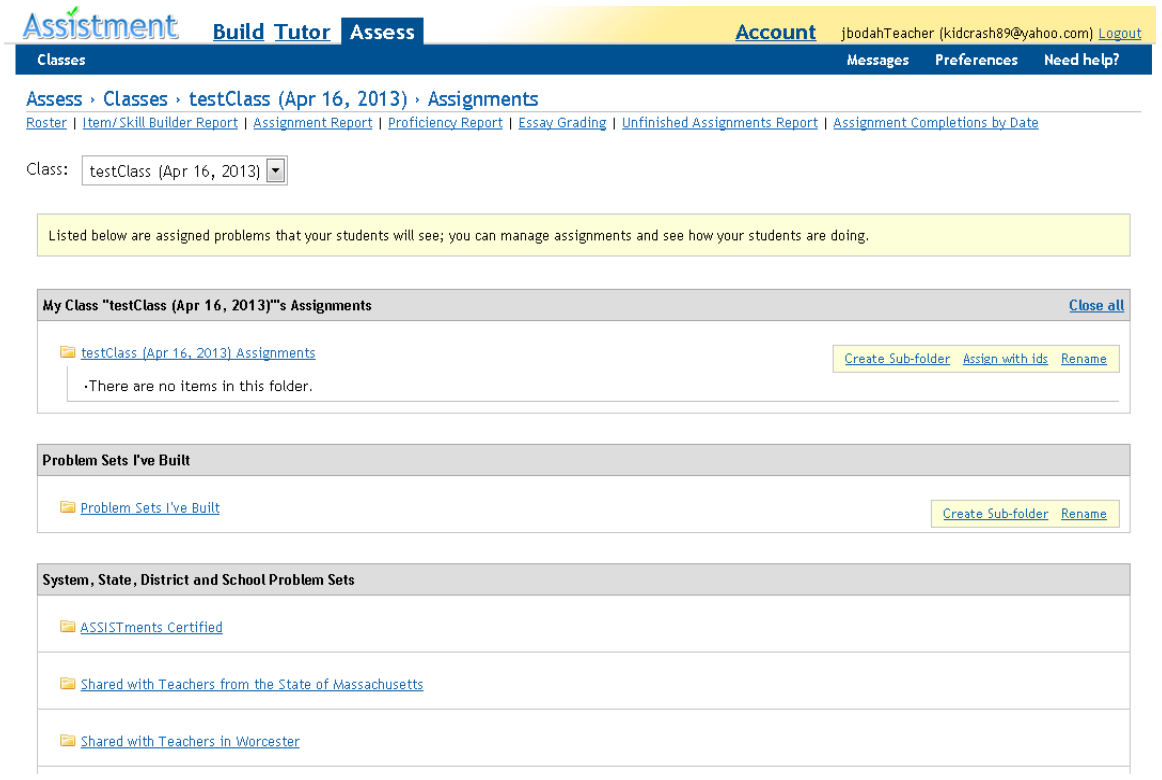
My Test Class (Apr 12, 2013)

Grade: N/A
Type: Anyone can enroll in the class
End Date: Apr 12, 2014

Edit Class Info | Settings Roster Assignments Disable

New Class

Figure A.4: (Original design) Assess tab with one class



jbodahTeacher (kidcrash89@yahoo.com) Logout

Assess › Classes › testClass (Apr 16, 2013) › Assignments

Roster | Item/Skill Builder Report | Assignment Report | Proficiency Report | Essay Grading | Unfinished Assignments Report | Assignment Completions by Date

Class: testClass (Apr 16, 2013)

Listed below are assigned problems that your students will see; you can manage assignments and see how your students are doing.

My Class "testClass (Apr 16, 2013)"'s Assignments — Close all

testClass (Apr 16, 2013) Assignments — Create Sub-folder Assign with ids Rename

·There are no items in this folder.

Problem Sets I've Built

Problem Sets I've Built — Create Sub-folder Rename

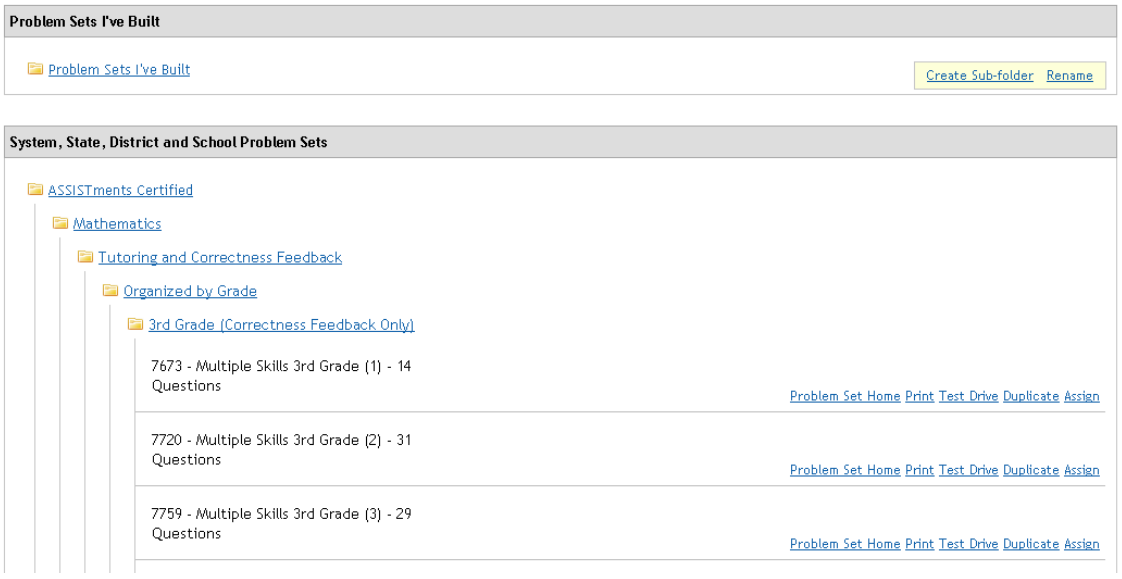
System, State, District and School Problem Sets

ASSISTments Certified

Shared with Teachers from the State of Massachusetts

Shared with Teachers in Worcester

Figure A.5: (Original design) Class' assignments page


Problem Sets I've Built

Problem Sets I've Built — Create Sub-folder Rename

System, State, District and School Problem Sets

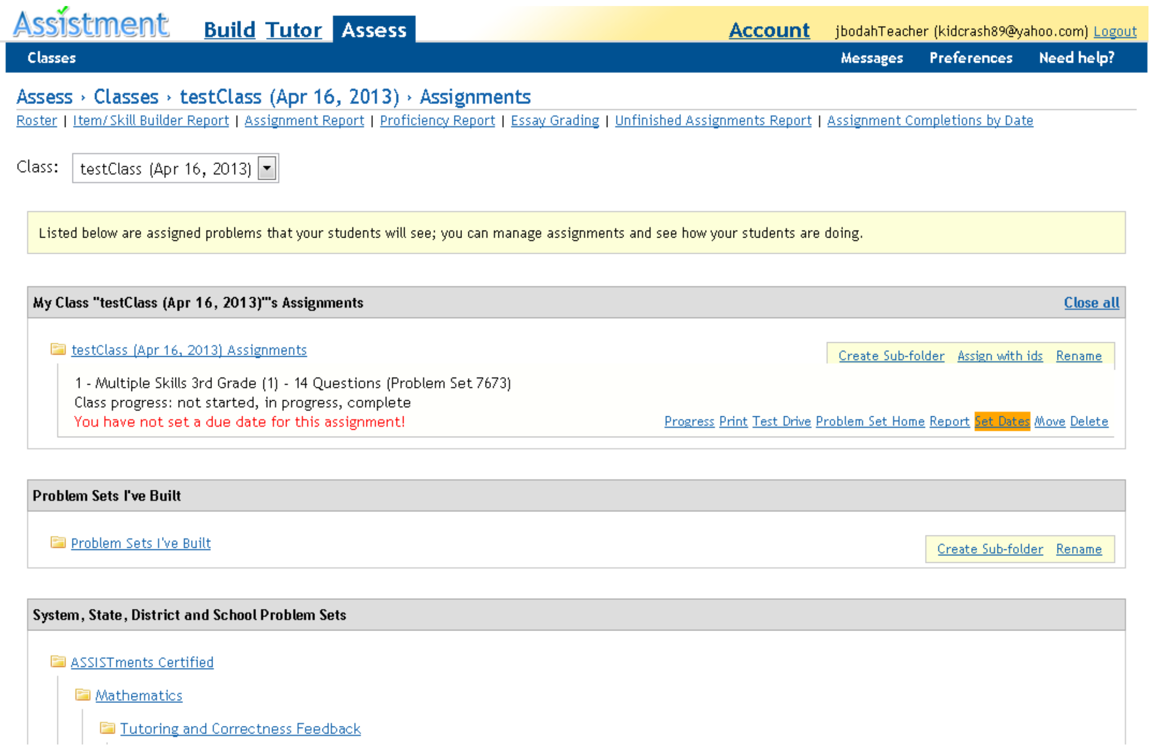
ASSISTments Certified
Mathematics
Tutoring and Correctness Feedback
Organized by Grade
3rd Grade (Correctness Feedback Only)

7673 - Multiple Skills 3rd Grade (1) - 14 Questions — Problem Set Home Print Test Drive Duplicate Assign

7720 - Multiple Skills 3rd Grade (2) - 31 Questions — Problem Set Home Print Test Drive Duplicate Assign

7759 - Multiple Skills 3rd Grade (3) - 29 Questions — Problem Set Home Print Test Drive Duplicate Assign

Figure A.6: (Original design) Class' assignments page, browsing content


Assistment Build Tutor Assess — Account — jbodahTeacher (kidcrash89@yahoo.com) Logout

Classes — Messages Preferences Need help?

Assess › Classes › testClass (Apr 16, 2013) › Assignments

Roster | Item/Skill Builder Report | Assignment Report | Proficiency Report | Essay Grading | Unfinished Assignments Report | Assignment Completions by Date

Class: testClass (Apr 16, 2013)

Listed below are assigned problems that your students will see; you can manage assignments and see how your students are doing.

My Class "testClass (Apr 16, 2013)"'s Assignments — Close all

testClass (Apr 16, 2013) Assignments — Create Sub-folder Assign with ids Rename

1 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set 7673)
Class progress: not started, in progress, complete
You have not set a due date for this assignment! — Progress Print Test Drive Problem Set Home Report Set Dates Move Delete

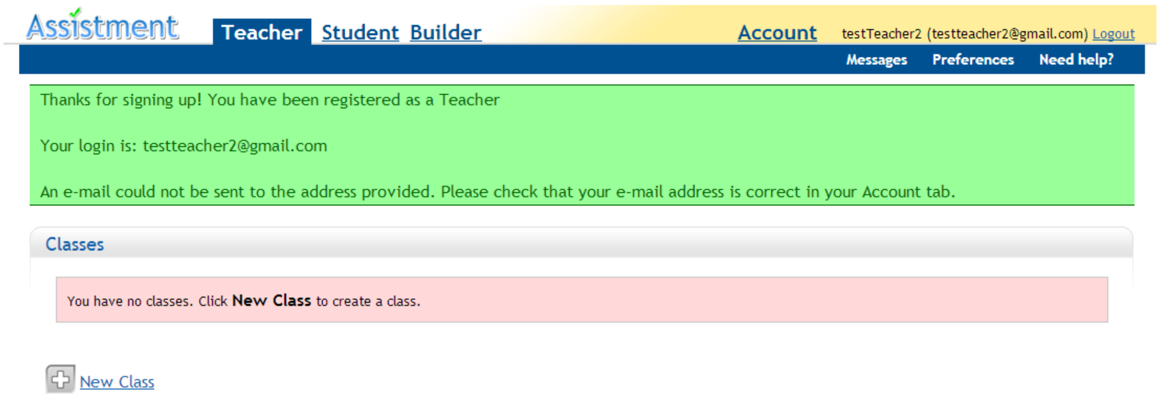
Problem Sets I've Built

Problem Sets I've Built — Create Sub-folder Rename

System, State, District and School Problem Sets

ASSISTments Certified
Mathematics
Tutoring and Correctness Feedback

Figure A.7: (Original design) Class' assignments page, after assigning



Thanks for signing up! You have been registered as a Teacher

Your login is: testteacher2@gmail.com

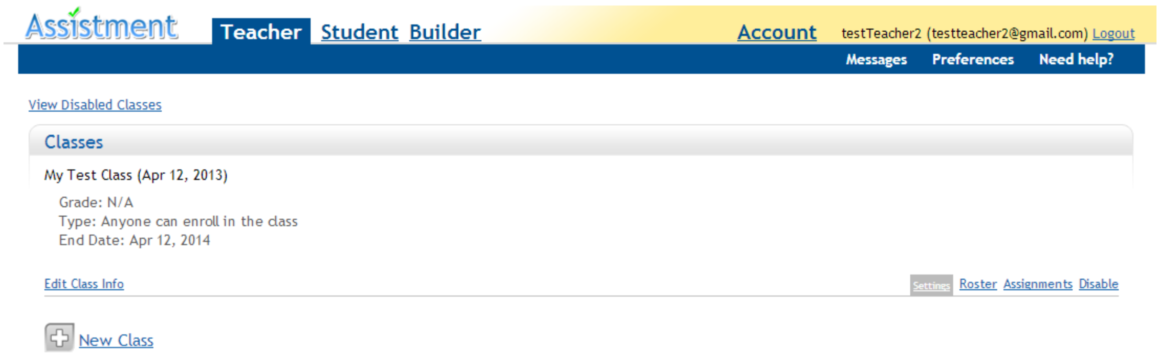
An e-mail could not be sent to the address provided. Please check that your e-mail address is correct in your Account tab.

Classes

You have no classes. Click **New Class** to create a class.

New Class

Figure A.8: (Iteration 1) Teacher tab

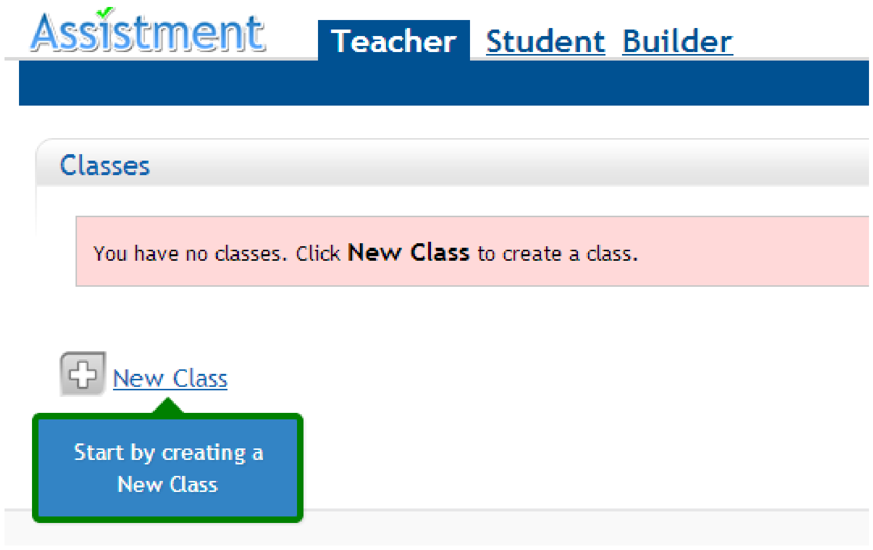


Classes

My Test Class (Apr 12, 2013)

Grade: N/A
Type: Anyone can enroll in the class
End Date: Apr 12, 2014

Figure A.9: (Iteration 1) Teacher tab with one class



You have no classes. Click **New Class** to create a class.

Figure A.10: (Iteration 2) Create a class tooltip

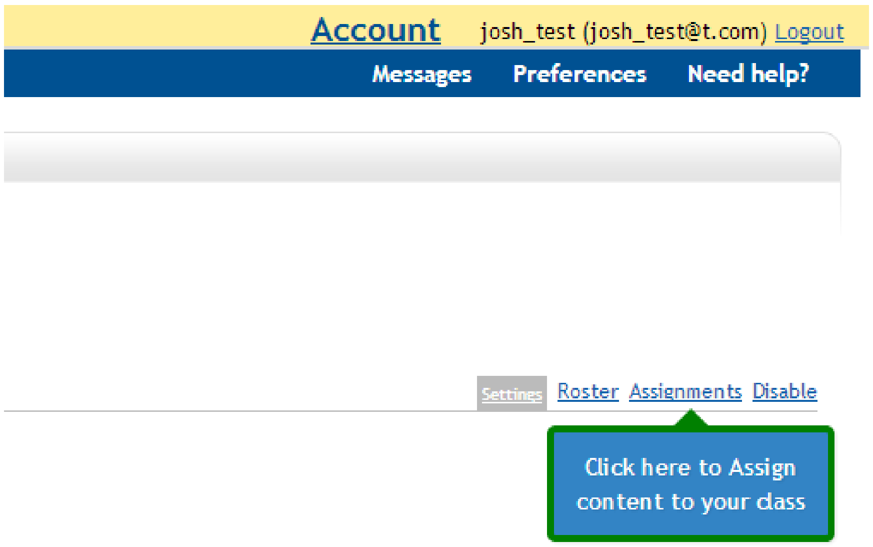
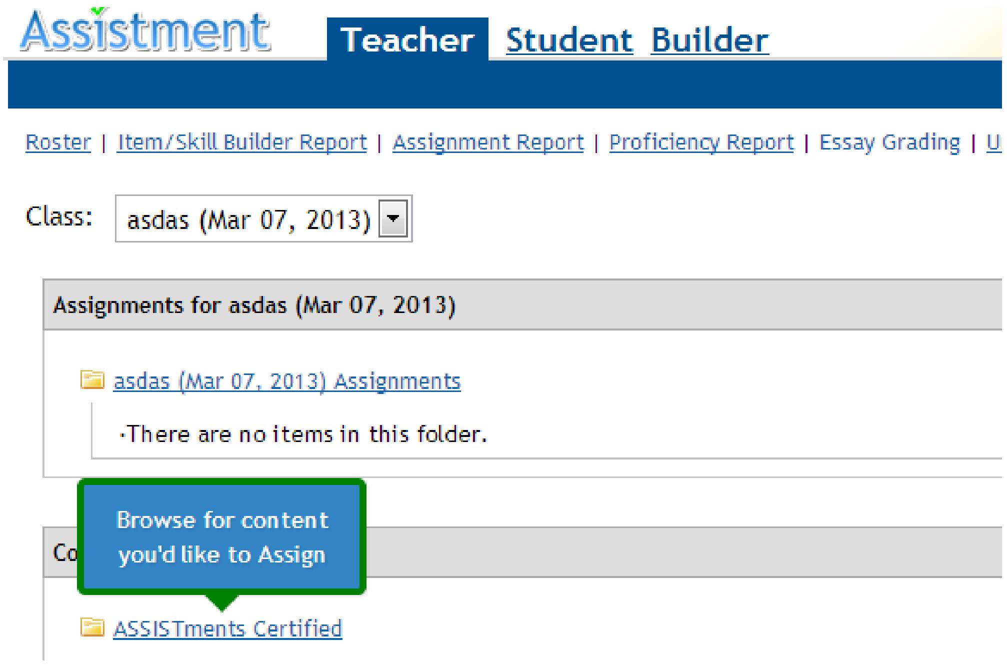

Account
josh_test (josh_test@t.com) Logout
Messages
Preferences
Need help?
Settings
Roster
Assignments
Disable
Click here to Assign content to your class

Figure A.11: (Iteration 2) Navigate to assignments tooltip

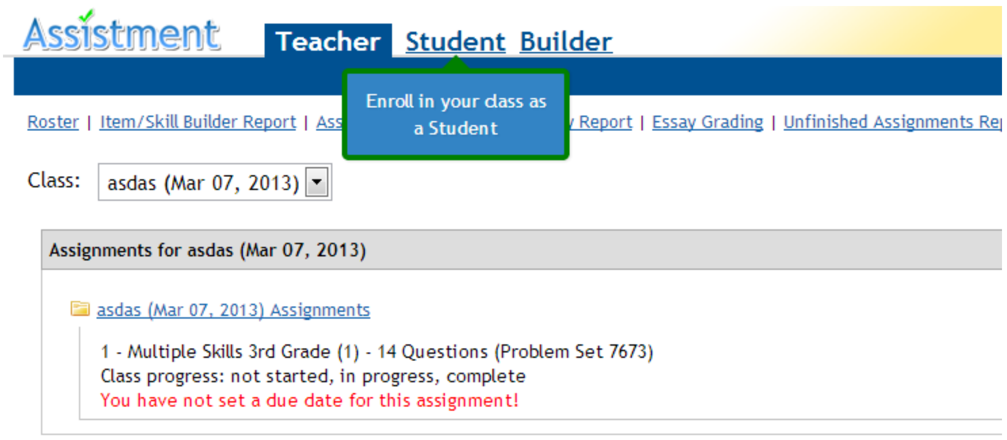

Assistment
Teacher
Student
Builder
Roster | Item/Skill Builder Report | Assignment Report | Proficiency Report | Essay Grading | U
Class:
asdas (Mar 07, 2013)
Browse for content you'd like to Assign
Co
ASSISTments Certified

Assignments for asdas (Mar 07, 2013)

asdas (Mar 07, 2013) Assignments

·There are no items in this folder.

Figure A.12: (Iteration 2) Browse for content tooltip

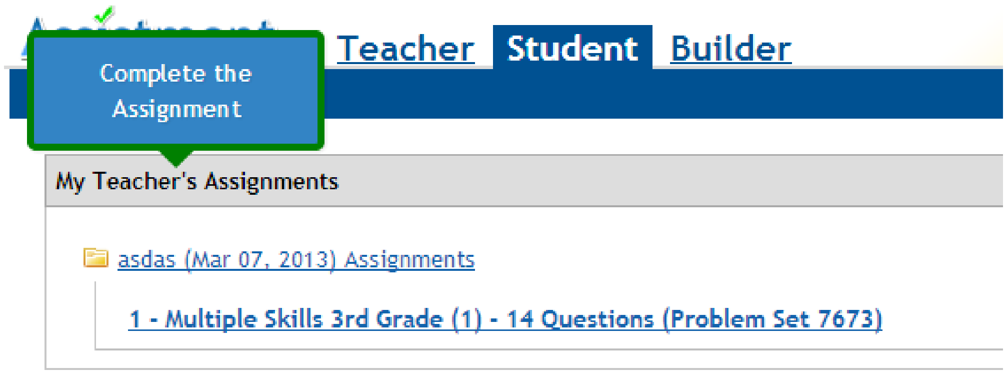

Assistment
Teacher
Student
Builder
Enroll in your class as a Student
Roster | Item/Skill Builder Report | Ass
Report | Essay Grading | Unfinished Assignments Re
Class:
asdas (Mar 07, 2013)

Assignments for asdas (Mar 07, 2013)

asdas (Mar 07, 2013) Assignments

1 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set 7673)
Class progress: not started, in progress, complete
You have not set a due date for this assignment!

Figure A.13: (Iteration 2) Navigate to Student tab tooltip

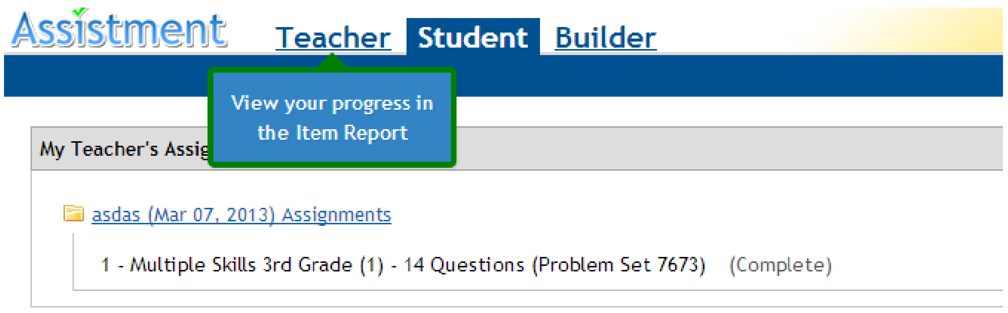


My Teacher's Assignments

asdas (Mar 07, 2013) Assignments

1 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set 7673)

Figure A.14: (Iteration 2) Complete the assignment tooltip



My Teacher's Assig

asdas (Mar 07, 2013) Assignments

1 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set 7673) (Complete)

Figure A.15: (Iteration 2) Navigate to Item Report tooltip

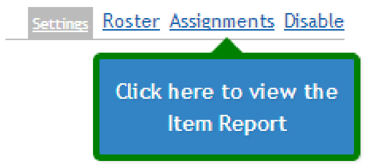
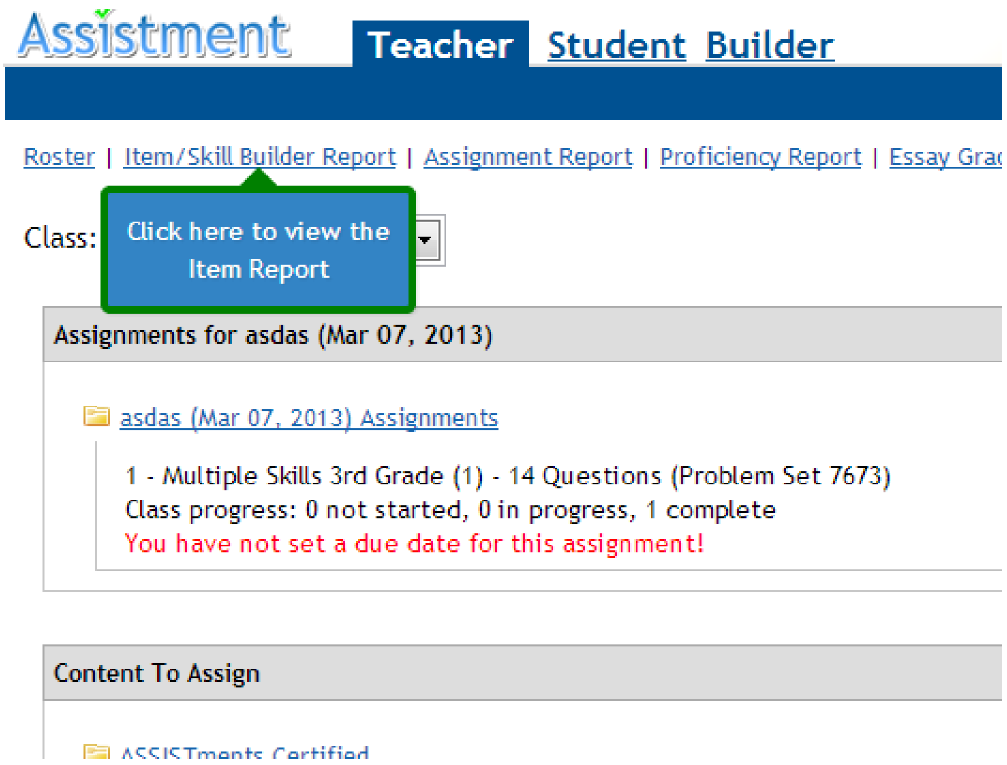


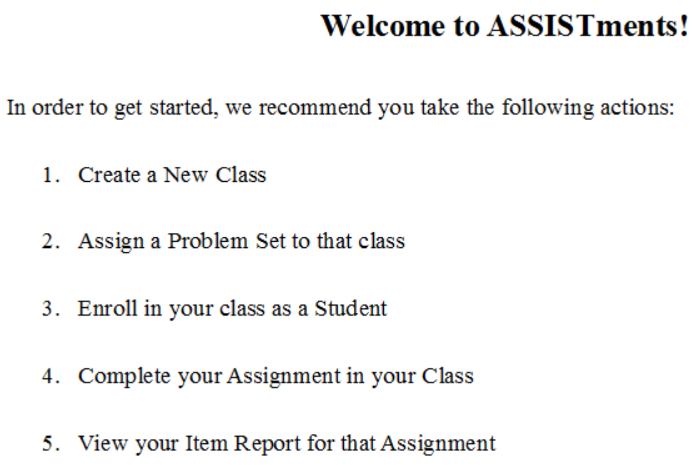
Figure A.16: (Iteration 2) Navigate to Item Report tooltip 2


Assistment
Teacher
Student
Builder
Roster | Item/Skill Builder Report | Assignment Report | Proficiency Report | Essay Grad
Class:
Click here to view the Item Report
Assignments for asdas (Mar 07, 2013)
asdas (Mar 07, 2013) Assignments

1 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set 7673)
Class progress: 0 not started, 0 in progress, 1 complete
You have not set a due date for this assignment!

Content To Assign
ASSISTments Certified

Figure A.17: (Iteration 2) View Item Report tooltip

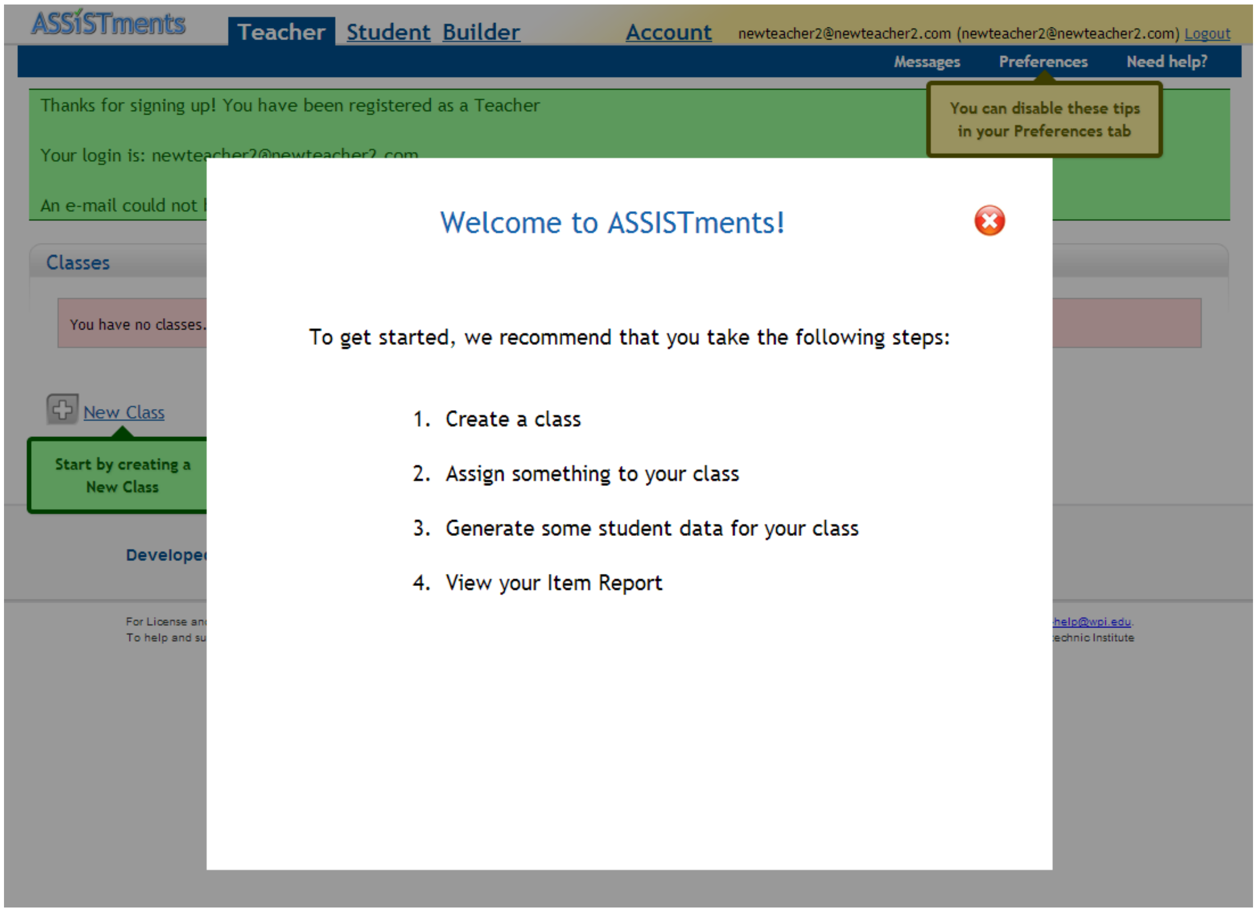


Welcome to ASSISTments!

In order to get started, we recommend you take the following actions:

1. Create a New Class
2. Assign a Problem Set to that class
3. Enroll in your class as a Student
4. Complete your Assignment in your Class
5. View your Item Report for that Assignment

Figure A.18: (Iteration 3) Paper prototype of landing splash


ASSISTments
Teacher
Student
Builder
Account
newteacher2@newteacher2.com (newteacher2@newteacher2.com) Logout
Messages
Preferences
Need help?
You can disable these tips in your Preferences tab
Classes
New Class
Start by creating a New Class
Develope

Thanks for signing up! You have been registered as a Teacher

Your login is: newteacher2@newteacher2.com

An e-mail could not

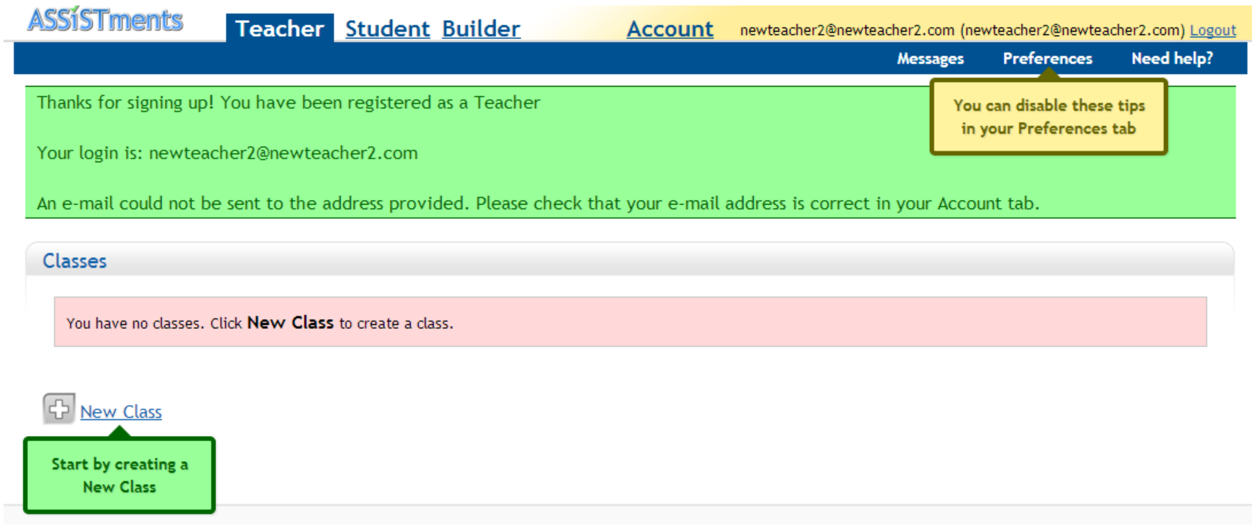
You have no classes.

Welcome to ASSISTments!

To get started, we recommend that you take the following steps:

1. Create a class
2. Assign something to your class
3. Generate some student data for your class
4. View your Item Report

Figure A.19: (Iteration 4) Landing splash



newteacher2@newteacher2.com (newteacher2@newteacher2.com)

Thanks for signing up! You have been registered as a Teacher

Your login is: newteacher2@newteacher2.com

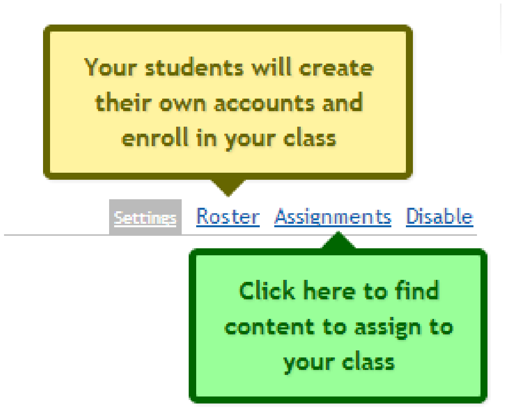
An e-mail could not be sent to the address provided. Please check that your e-mail address is correct in your Account tab.

You can disable these tips in your Preferences tab

You have no classes. Click **New Class** to create a class.

Start by creating a New Class

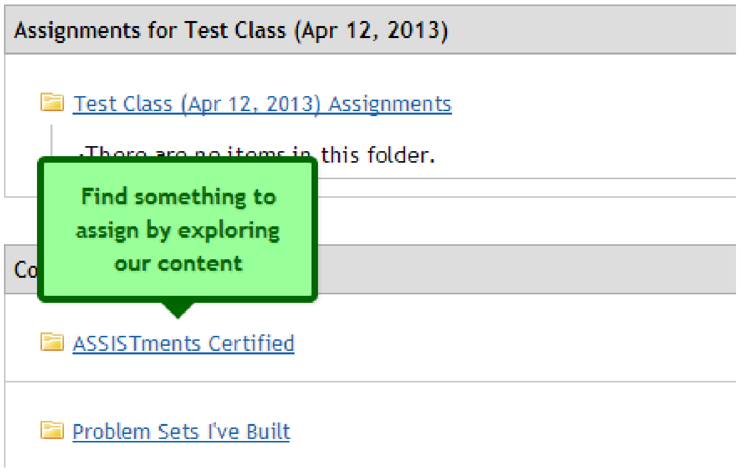
Figure A.20: (Iteration 4) Create a class and disable explanation tooltips



Your students will create their own accounts and enroll in your class

Click here to find content to assign to your class

Figure A.21: (Iteration 4) Navigate to assignments and roster explanation tooltips


Assignments for Test Class (Apr 12, 2013)

Test Class (Apr 12, 2013) Assignments

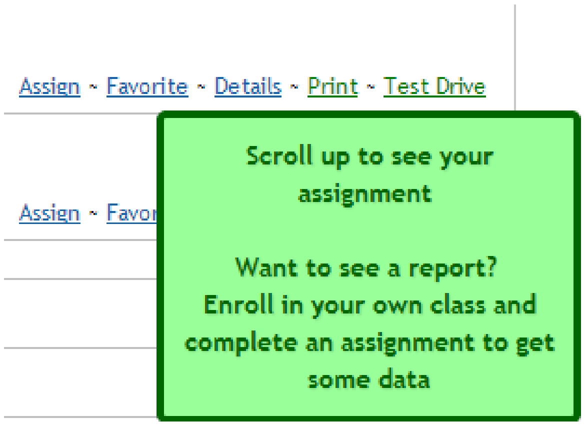
There are no items in this folder.

Find something to assign by exploring our content

ASSISTments Certified

Problem Sets I've Built

Figure A.22: (Iteration 4) Browse for content tooltip


Assign ~ Favorite ~ Details ~ Print ~ Test Drive

Assign ~ Favor

Scroll up to see your assignment

Want to see a report? Enroll in your own class and complete an assignment to get some data

Figure A.23: (Iteration 4) Scroll up to enroll tooltip

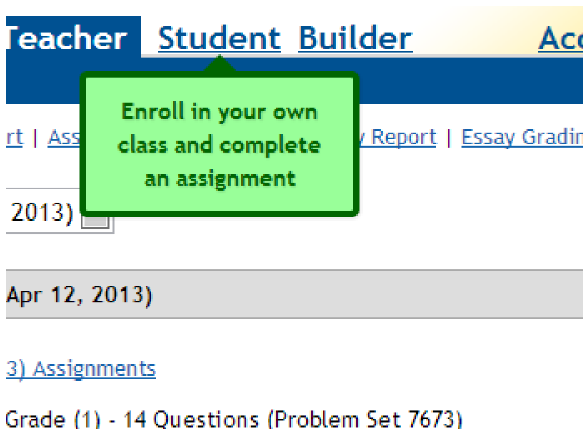

Teacher Student Builder Acc
Enroll in your own class and complete an assignment
rt | Ass
Report | Essay Gradir
2013)
Apr 12, 2013)
3) Assignments
Grade (1) - 14 Questions (Problem Set 7673)

Figure A.24: (Iteration 4) Navigate to Student tab tooltip

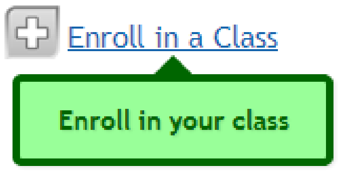


Figure A.25: (Iteration 4) Enroll in your class tooltip

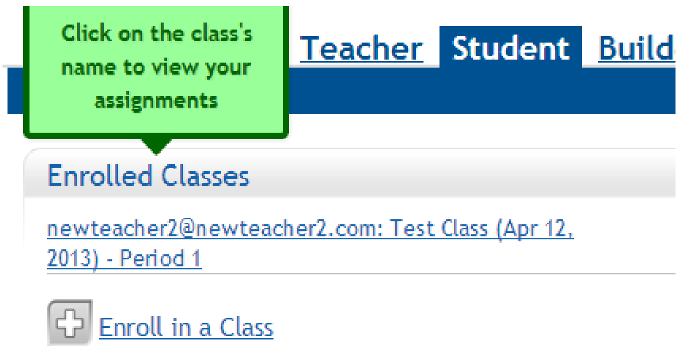

Click on the class's name to view your assignments
Teacher
Student
Build
Enrolled Classes
newteacher2@newteacher2.com: Test Class (Apr 12, 2013) - Period 1
Enroll in a Class

Figure A.26: (Iteration 4) Navigate to student assignments tooltip

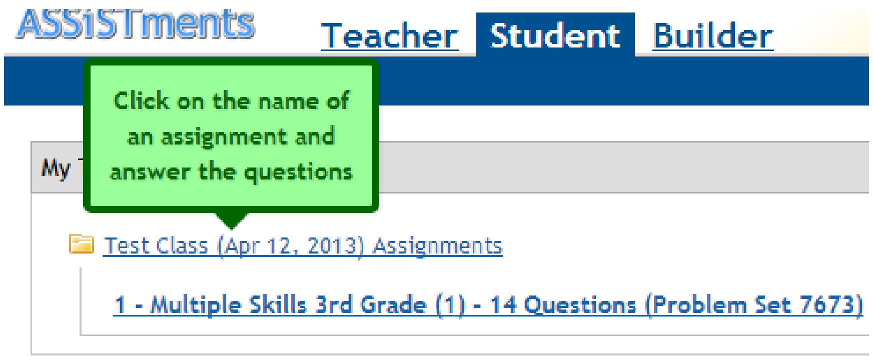
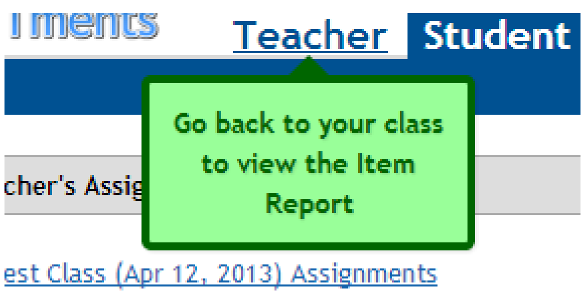
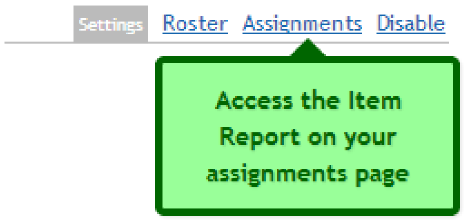

ASSISTments
Teacher
Student
Builder
Click on the name of an assignment and answer the questions
My
Test Class (Apr 12, 2013) Assignments
1 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set 7673)

Figure A.27: (Iteration 4) Complete the assignment tooltip



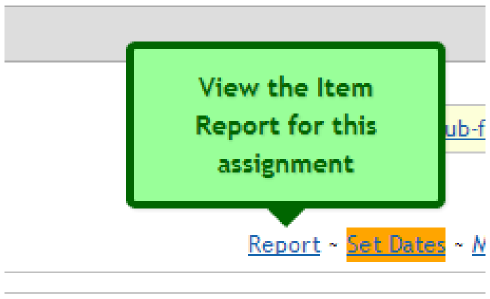
Go back to your class to view the Item Report

Figure A.28: (Iteration 4) Navigate to Item Report tooltip



Access the Item Report on your assignments page

Figure A.29: (Iteration 4) Navigate to Item Report tooltip 2



View the Item Report for this assignment

Figure A.30: (Iteration 4) View Item Report tooltip

Bibliography

[BOM$^+$12] Z. Broderick, C. O'Connor, C. Mulcahy, N. Heffernan, and C. Heffernan. Increasing parent engagement in student learning using an intelligent tutoring system. *Journal of Interactive Learning Research*, 22(4):523–550, 2012.

[Bur05] K. Burke. Online A/B testing. *Target Marketing*, 28(5):37–41, 2005.

[BWH$^+$08] R. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger. Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19(2):185–224, 2008.

[CFCM02] J. Cradler, M. Freeman, R. Cradler, and M. McNabb. Research implications for preparing teachers to user technology. *Learning & Leading with Technology*, 30(1):50–55, 2002.

[CFKL09] T. Crook, B. Frasca, R. Kohavi, and R. Longbotham. Seven pitfalls to avoid when running controlled experiments on the web. *KDD'09*, 2009.

[CR87] J. Carroll and M. Rosson. Paradox of the active user. In *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. MIT Press, Cambridge, MA, 1987.

[fS98] International Organization for Standardization. ISO/IEC 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) part 11: Guidance on usability. 1998.

[GBH10] Y. Gong, J. Beck, and N. Heffernan. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education*, 2010.

[GL85] J.D. Gould and C. Lewis. Designing for usability: Key principles and what designers think. *Communications of the ACM*, 28(3):300–311, 1985.

[Gra98] T. Grayling. Fear and loathing of the help menu: A usability test of online help. *Technical Communication*, Second Quarter:168–179, 1998.

[KCL09] R. Kohavi, T. Crook, and R. Longbotham. Online experimentation at microsoft. *Third workshop on Data Mining Case Studies and Practice Prize*, 2009.

[KDF$^+$12] R. Kohavi, A. Deng, B. Frasca, R. Longbotham, T. Walker, and Y. Xu. Trustworthy online controlled experiments: Five puzzling outcomes explained. *KDD'12*, 2012.

[KLSH09] R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne. Controlled experiments on the web: Survey and practical guide. *Data Mining and Knowloedge Discovery Journal*, 18(1):140–181, 2009.

[Kru06] S. Krug. *Don't make me think: A Common sense approach to web usability*. New Riders, Berkley, CA, 2nd edition, 2006.

[Kru10] S. Krug. *Rocket surgery made easy: The do-it-yourself guide to finding and fixing usability problems*. New Riders, Berkley, CA, 2010.

[Lew82] C. Lewis. *Using the "Thinking Aloud" method in cognitive interface design*. Technical report, IBM, 1982.

[Nie94] J. Nielsen. *Usability engineering*. San Diego: Academic Press, 1994.

[Nie01a] J. Nielsen. Error-message guidelines. http://www.useit.com/alertbox/20010624.html, 2001. [Online; accessed 2013-03-06].

[Nie01b] J. Nielsen. Horizontal attention leans left. http://www.nngroup.com/articles/horizontal-attention-leans-left/, 2001. [Online; accessed 2013-03-12].

[Nie05] J. Nielsen. Putting A/B testing in its place. http://www.useit.com/alertbox/20050815.html, 2005. [Online; accessed 2012-11-30].

[Nie07] J. Nielsen. Banner blindness: Old and new findings. http://www.nngroup.com/articles/banner-blindness-old-and-new-findings/, 2007. [Online; accessed 2013-03-12].

[Nie12] J. Nielsen. Usability 101: Introduction to usability. http://www.nngroup.com/articles/usability-101-introduction-to-usability/, 2012. [Online; accessed 2013-03-06].

[NM90] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. *CHI'90 Proceedings*, pages 249–256, 1990.

[PGBH12] Z. Pardos, S. Gowda, R. Baker, and N. Heffernan. The sum is greater than the parts: Ensembling models of student knowledge in educational software. *ACM's Knowledge Discovery and Datamining Explorations*, 13(2):37–44, 2012.

[PP01] A. Parush and D. Parush. Online help: Too much of a good thing? *Usability Interface*, 7(3), 2001.

[Ret91] M. Retting. Nobody reads documentation. *Communications of the ACM*, 34(7):19–24, 1991.

[RMN$^+$07] L. Razzaq, Feng M., Heffernan N., Koedinger K., Junker B., Nuzzo-Jones G., Macasek M., Rasmussen K., Turner T., and Walonoski J. A web-based authoring tool for intelligent tutors: Blending assessment and instructional assistance. *Intelligent Educational Machines: Studies in Computational Intelligence*, 44:23–49, 2007.

[Sir10] D. Siroker. How Obama raised $60 million by running a simple experiment. http://blog.optimizely.com/2010/11/29/how-obama-raised-60-million-by-running-a-simple-experiment, 2010. [Online; accessed 2012-11-31].

[SL04] D. Silvernail and D. Lane. *The impact of Maine's one-to-one laptop program on middle school teachers and students*. Technical report, Maine Education Policy Research Institute, 2004.

[Spo96] J. Spool. Tips and hints. *Eye for design*, 3(5):5, 1996.

[Spo97] J. Spool. Online help: Sometimes doesn't help. *Eye for design*, 4(3):9–10, 1997.