Digitool: Digital Logic Self - Study Toolkit

Traybourne North, Miguel Quinones, Spencer Clark, Langqing Dai

Office of Undergraduate Research and Creative Activity

College of Engineering, Informatics, and Applied Sciences, Northern Arizona University, Flagstaff, AZ

Client: Xi Zhou | Mentor: Fabio Santos

Abstract

Virtual learning has been integrated in almost every educational system throughout the nation. Because of this, creating a virtual learning environment for students who lack the ability to perform efficiently in the introductory electrical engineering course was essential. Understanding how a virtual learning environment works and functions was important when creating the components necessary for our virtual environment to function effectively. Many different design approaches could be used, but our team had to be aware of database communication, scalability, and portability issues that could occur when using a desired design approach. In summary, this poster introduces an alternative learning method to help increase student proficiency, while allowing students to learn in an interactive way. Once our product is finished, we will be able to gather results from our client, who had his students test our virtual learning environment. With this product, we plan to eliminate the amount of Teacher Assistants that get assigned to the course. By doing so, this will not only give Northern Arizona University extra funds to use elsewhere, but it will also reduce the workload given towards the professor that teaches the Introduction to Electrical Engineering course.

Motivation

The Intro to Digital Logic course here at NAU is a foundational course for many engineering students here, ranging from Electrical Engineering to Computer Science. As such, a very large number of students take the course each year, around 75 per semester. The course introduces complicated digital logic topics to the students that take it, such as translating, grouping, and creating equations for Karnaugh map truth tables, as well as performing numeric conversions from one base to another involving binary, octal, decimal, and hexadecimal. As such, students that have difficulty learning the topics of the course might want to hire a tutor to assist them, but this can cost a significant amount of money. Similarly, it costs money for professors to hire teaching assistants to help students. A potential solution to this problem for both sides could be for students to meet with professors when it is convenient for both sides, but this isn't always possible. This is why our client, Assistant Professor of Practice Xi Zhou, who teaches the Intro to Digital Logic course, tasked us with creating a solution to these problems.

The main problems our client needed us to solve were:

- Students might have difficulty learning the topics of the Intro to Digital Logic course
- Students only have a limited amount of time to work in the class
- Students need a free, straightforward application that would allow them to independently explore and work on the topics of the course on their own time

To recap, the main motivations of this project are:

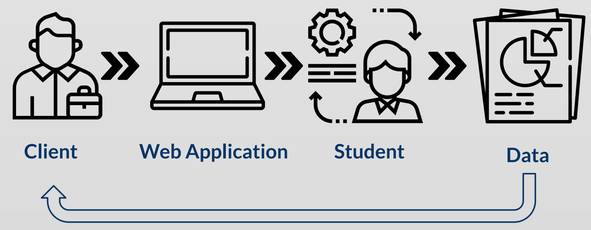
- Students might have difficulty learning the complex topics of the Intro to Digital Logic course, which our client teaches
- Students and professors have other obligations they need to meet throughout the day, meaning when there is no time to meet with each other students might hire tutors and professors might hire teaching assistants, costing both sides significant amounts of money
- A solution needed to be created that would address all the problems brought forth by our client in the form of a free piece of technology that students would be able to utilize

Solution Overview

Our solution is to design a web application that will provide students with exercises related to classroom content. Our goal is to improve students' understanding of various concepts discussed in the class such as Karnaugh maps and numeric conversions. To do this, we designed learning modules (with the option to select practice mode) in which questions will be provided to students based on the module they choose.

Another goal is to provide our client with statistical information to access how a student performs when it comes to each module. This part will be done by recording student answers into our database.

By implementing our proposed solutions, it can increase proficiency rates of each topic provided in this web application and it can reduce the number of TAs and SI leaders assigned to the course (EE110). This can result in NAU saving a lot of money.

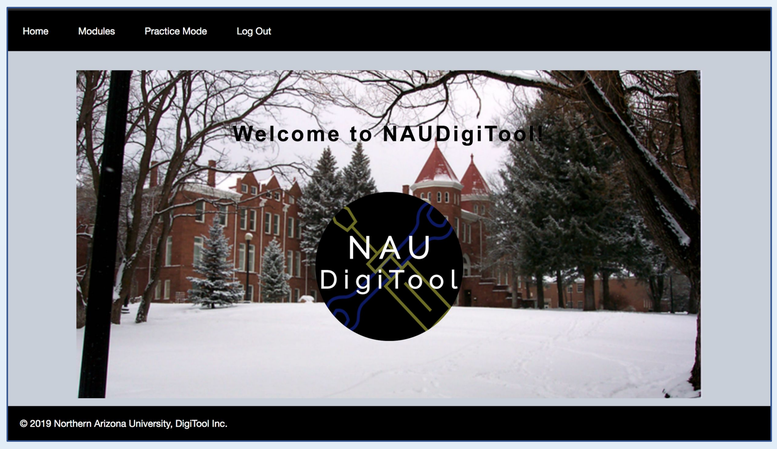


Key Features

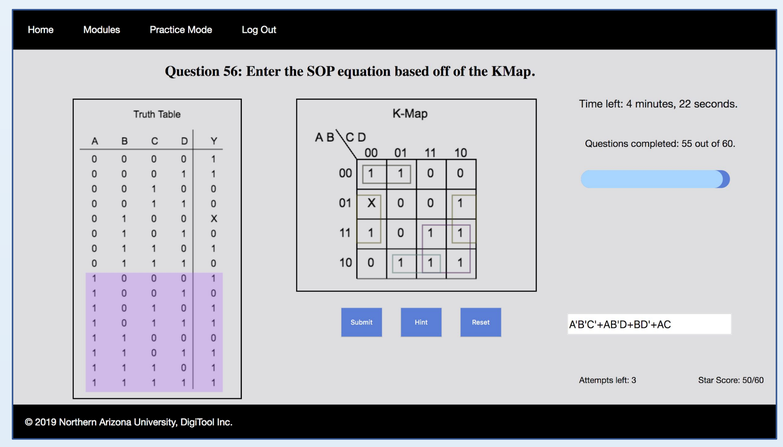
Our application will allow users to create an account where their problem completion is stored. The problem completion gives us two big features; saving where the user logged out within a module so they can resume where they left off, and statistics that can be given to an instructor to analyze and find problem areas in the course.

Most tools in the digital logic area are simple solvers that the user inputs the question and it spits out the answer. This application is not a solver, it provides the user a question in an interactive format and checks that their answer is correct.

The application does not use the database for storing questions, instead it is able to randomly generate them and compute the answer. This means our application can create any problem and the user will not run out of material to practice until they have exhausted all possible problems.



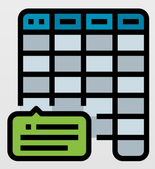
Web Application Homepage



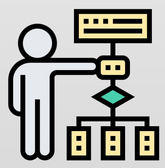
Example Problem

Challenges

Karnaugh maps (the main focus of our project) are a fairly complex logical problem and since we are randomly creating problems and not simply storing question answer pairs in a database, we have to generate the map and follow all of the logic necessary to compute the answer. In order to model them effectively we have to do a lot of heavy lifting behind the scenes to ensure that our application comes to the correct solution. If a student enters a correct answer, but our application did not generate the correct answer and it tells the student they are wrong, it completely destroys their ability to learn from it.



To fix this issue, we chose not to generate answers with an advanced algorithm, but instead to break each step into its own part that is easily comprehensible and testable. Not only does this make us confident our logic is correct, anybody new to our project or that has to maintain it can easily pick up where we left off.



Testing

In order to ensure that our web application meets the requirements and solves the problems given to us by our client, we performed a variety of tests on the key modules of our software, including unit tests, integration tests, and usability tests. We performed unit tests on our web application by using boundary values and equivalence partitions based on the questions asked of our application to test the correctness of our key modules. We ran integration tests to check that the data we get from user inputs within the application are properly sent to our database where it can be viewed by our client to see how well his students are doing with the course topics. Finally, we used usability tests to emulate the interactions a user would have with the front-end portion of our application to ensure that our software was easy to use at a first glance and that none of the front-end components were confusing for users.

Outcomes

Because of our web application, numerous outcomes were able to be achieved in the amount of time that we were given.

1. **Money saved by hiring less tutors**

2. **More practice given towards students**

3. **Access to student statistics**

Future Work

For future work, our web application could expand by implementing the ideas below.

1. **Adding additional module concepts**

2. **Creating a secondary application that can automatically analyze statistics from our database**