GradPlan

Create Your Program of Study with Ease

By JONATHAN ALMEIDA

Master's Project Report

Submitted to

Oregon State University

In partial fulfillment of
the requirements for the degree of
Master of Science in Computer Science

Presented September 3rd, 2019

COMMITTEE

Christopher Scaffidi
Mike Bailey
Justin Wolford

Contents

Abstract

New MS in CS students in the Electrical Engineering and Computer Science school at OSU are required to file their Program of Study by the end of their 2nd term. Many of them, especially international students, are in a totally new ecosystem, so they find it overwhelming to choose the right courses, and to understand the degree requirements. They face questions such as “I’m interested in Data Science, which courses should I be taking?”, “Would this course meet the graduation requirements set by school?”, “What are slash courses?”, “Which bucket does this course belong to?”, etc. Students must come up with a tentative coursework plan, also known as Program of Study. Typical situations students find themselves while filling out the Program of Study would be “How many credits was this class for again?”, or “Is this a slash course? Would it count towards graduation?”, etc. Further complicating the process, manually filling out the PDF template provided by the university is cumbersome.

The GradPlan web application addresses these issues by guiding the user through few steps and generating a PDF file which could be submitted as the Program of Study document required by the school. This document satisfies the course requirements set by the school for MS in CS students. The GradPlan system allows MS in CS students to create a satisfactory Program of Study document in a fast and efficient manner thus saving time. This promising result suggests that the GradPlan system could serve as a starting point for future enhancements for a fully automated Course Recommender system which could be used by all the graduate students in the School of EECS.

1. Introduction

One day as I was checking my emails, I saw an email titled “Your Program of Study is due!”. The email was from the Graduate Program Coordinator at the School of Electrical Engineering and Computer Science, and it said that I’m required to complete my Program of Study as soon as possible. I started to enquire about this notification with my friends and asked them if they had received such notification. And to my relief they did. There were many people in the same boat, and everyone was trying to wrap their heads around this.

I started working towards this by Google searching “program of study oregon state university”, and luckily the first link was the official Oregon State University Graduate School link, which had some useful information. It answered the questions “What is a Program of Study?” and “why is it needed?”. Somewhere on the page it was mentioned that MS students are required to file their Program of Study before they complete 18 credits are Oregon State University. I remembered that my advisor had mentioned this form and shown an example. It started to make sense. I and many other people had completed 18 credits, but we didn’t file the document, and hence we got this email.

Like many other people, I dislike paper work. As I opened the Program of Study PDF template, the only thoughts running through my head were “This is a tedious and a boring task. Let’s get done with this as quickly as possible!”. I asked my peers if they wanted to work on this task together, and soon we started working on it in one of the labs in Kelley Engineering center. Fortunately, my advisor and the Graduate Program Coordinator had provided some sample documents which served as a good starting point. However, since each of us had a different field of interest than the provided sample documents, we were struggling to verify if the courses we had in mind would meet the school's degree requirements, and hence we eventually found ourselves scrutinizing the EECS school 'Master of Science Degree Requirements' webpage and the OSU course catalog.

This was turning out to be a time-consuming process. For each course we were interested in, we searched for the course number and the number of credits in the course catalog, then checked the EECS webpage to see which bucket area the course belonged to, identify if the course was a slash course, then type it in the PDF template, and once we had done this for like 8 – 9 courses we had to verify if these courses met the required total credit count and did not exceed the maximum allowed slash course credits. This entire process took us a solid one hour.

Every graduate student in the Electrical Engineering and Computer Science school is required to submit the Program of Study document. And if this is the case, this should not be a tedious and a time-consuming task. The GradPlan system is aimed at solving this problem by allowing its users to create their Program of Study in a quick, easy, and efficient manner.

While there are several great coursework planning systems out there with many interesting features which help students to create and follow personalized roadmaps to degree completion, none of them help solve the problem mentioned in the above paragraphs. I will describe these systems in the related work section below, but before that, I will present the key claims about the GradPlan system in the next section.

2. Key Claims

The following key claims define why the GradPlan system is innovative. The primary objective of this document is to explain how the GradPlan system satisfies these key claims.

- **Key Claim #1:** *GradPlan generates a satisfactory MS CS Program of Study which meets the EECS school requirements*

  This claim is important because the objective of the GradPlan system is to allow MS in CS students to create their Program of Study that can be filed with the Electrical Engineering and Computer Science school. I am not claiming that all the students in the school would be able to create their Program of Study using GradPlan (particularly PhD students), but I do claim that most MS students in computer science at Oregon State will find it suitable.
- **Key Claim #2:** *GradPlan generates a Program of Study that is equally satisfactory as manually filling out the PDF template*

  This claim is important because building a system that creates a Program of Study which is not as good as manually created document would not be so useful. Also, the GradPlan system suggests course areas based on the student’s field of interest, and these suggestions must be as good as what the students could find for themselves on their own.
- **Key Claim #3:** *GradPlan saves time compared to manually filling out a PDF template*

  This is the most important claim since it solves the major problem with manually creating the Program of Study, which is that creating the Program of Study manually requires a lot of time.

I am aware of no other system at Oregon State University that satisfies all three of these claims. In fact, I have not yet come upon any system at another university that combines the key features of GradPlan: a course area suggestion agent, integrated checking for whether the plan meets school requirements, and automated generation of a PDF suitable for signatures and submission. I believe that all these claims are necessary to provide a way for MS students at our school to more easily and efficiently construct a Program of Study.

3. Related Work

This section presents various other systems aimed at solving a similar problem. These systems, while being innovative and useful to their specific problem scenarios, fall short of satisfying the key claims listed above.

1. **Manually filling out the PDF template**

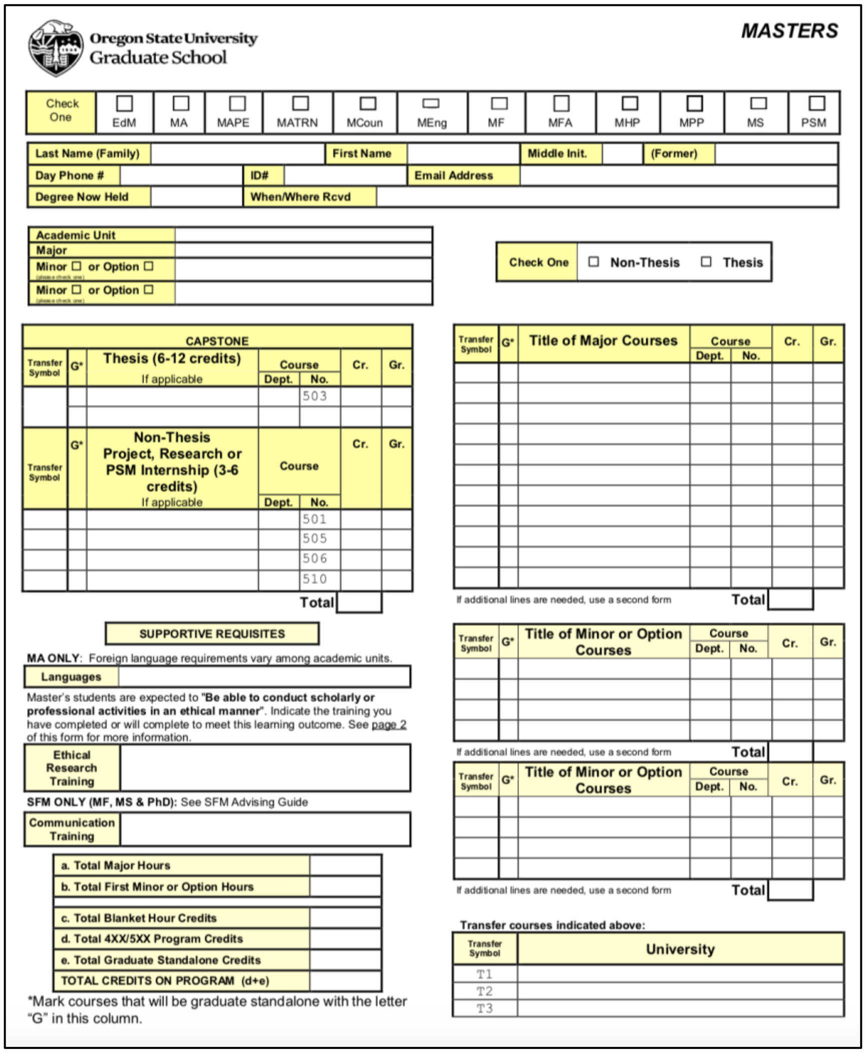
This is the primary GradPlan system competitor [OSU, 2019]. Manually filling out the Program of Study PDF template is the traditional method which MS students in the Electrical Engineering and Computer Science school have been using for years. The GradPlan system is aimed at improving this process.

The process involves downloading a PDF template from the graduate school website. Then the student can choose to either fill out the PDF electronically or print out the PDF and fill it out in ink. Figure 1 shows the Program of Study PDF template.

Manually filling out the pdf template has the following disadvantages:

- Students need to manually verify if the EECS school requirements are met, which is time consuming.
- Students must manually fetch the course details from the course catalog, which is also time consuming.
- Students are often unsure what courses are most relevant to their interest, and so they spend time hunting about for which courses they should be taking given their area of interest.

Although this method of creating the Program of Study has been successful, its inefficient and time consuming and, as mentioned in the introduction, can take up to an hour.



Oregon State University
Graduate School

MASTERS

Check One	□ EdM	□ MA	□ MAPE	□ MATRN	□ MCoun	□ MEng	□ MF	□ MFA	□ MHP	□ MPP	□ MS	□ PSM

Last Name (Family)		First Name		Middle Init.		(Former)	
Day Phone #		ID#		Email Address			
Degree Now Held		When/Where Rcvd					

Academic Unit	
Major	
Minor □ or Option □ (please check one)	
Minor □ or Option □ (please check one)	

Check One	□ Non-Thesis	□ Thesis

CAPSTONE						
Transfer Symbol	G*	Thesis (6-12 credits) If applicable	Course		Cr.	Gr.
			Dept.	No.		
				503		
Transfer Symbol	G*	Non-Thesis Project, Research or PSM Internship (3-6 credits) If applicable	Course		Cr.	Gr.
			Dept.	No.		
				501		
				505		
				506		
				510		
				Total		

Transfer Symbol	G*	Title of Major Courses	Course		Cr.	Gr.
			Dept.	No.		

If additional lines are needed, use a second form Total

SUPPORTIVE REQUISITES

MA ONLY: Foreign language requirements vary among academic units.

Languages	

Master's students are expected to **"Be able to conduct scholarly or professional activities in an ethical manner"**. Indicate the training you have completed or will complete to meet this learning outcome. See page 2 of this form for more information.

Ethical Research Training	

SFM ONLY (MF, MS & PhD): See SFM Advising Guide

Communication Training	

Transfer Symbol	G*	Title of Minor or Option Courses	Course		Cr.	Gr.
			Dept.	No.		

If additional lines are needed, use a second form Total

Transfer Symbol	G*	Title of Minor or Option Courses	Course		Cr.	Gr.
			Dept.	No.		

If additional lines are needed, use a second form Total

a. Total Major Hours	
b. Total First Minor or Option Hours	
c. Total Blanket Hour Credits	
d. Total 4XX/5XX Program Credits	
e. Total Graduate Standalone Credits	
TOTAL CREDITS ON PROGRAM (d+e)	

Transfer courses indicated above:

Transfer Symbol	University
T1	
T2	
T3	

*Mark courses that will be graduate standalone with the letter "G" in this column.

Figure 1: Current Program of Study PDF template [OSU, 2019]

Some students become so frustrated and confused by the process that they need to meet with their advisors to get advice about how which courses to take and how to complete the form. This adds still more time to the process because the student needs to spend time meeting with the advisor. The student also usually must wait a day or two until the advisor can find time to meet. And finally, even after getting advice, the student yet must return to the PDF template and fill it out, which still takes more time.

Throughout this process, even with an advisor's help, there is a risk that the student will make mistakes. In most cases, these mistakes won't be identified until when the student prints the form and brings it around to the advisor and committee for signatures. A simple arithmetic error of incorrectly adding up the credit hours can result in having to fix the form (which, for a hand-written form essentially means filling out a new form from scratch)—and then make another round of visiting the professors for signatures.

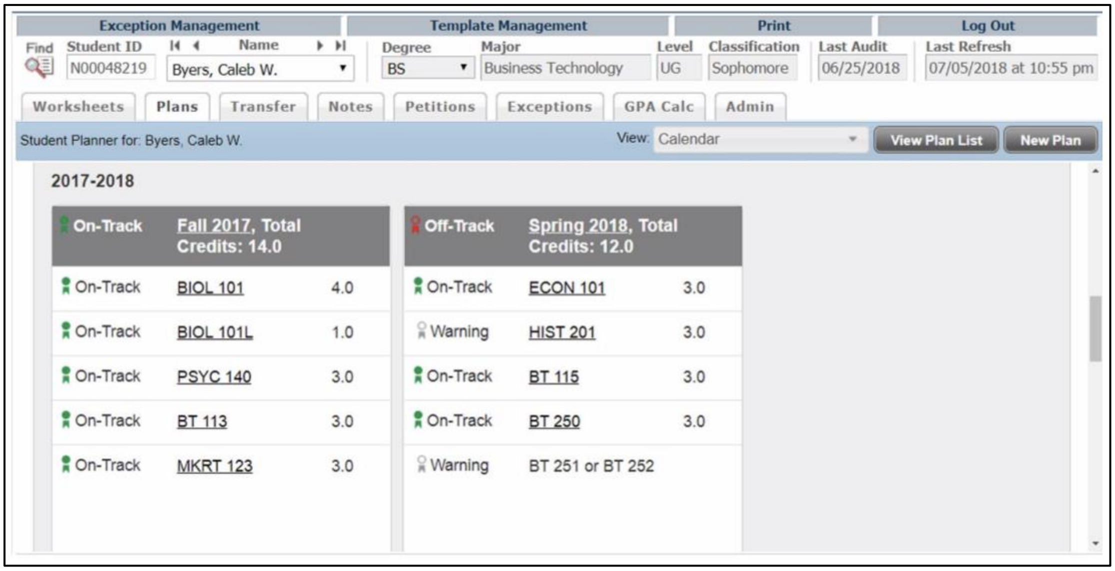
In summary, the current process is exceptionally tedious, error-prone and time-consuming.

Comparison with Automated Coursework Planner Key Claims:

- **KC #1 (Verification of EECS course requirements?)** No. The manually created Program of Study document must be verified manually to check if the requirements set by the Electrical Engineering and Computer Science school are met. This is a cumbersome process which involves checking the bucket requirements, credits requirements, and the slash course requirements.
- **KC #2 (Creates satisfactory Program of Study?)** Yes. This method eventually creates a satisfactory Program of Study document.
- **KC #3 (Efficient?)** No. Manually creating the Program of Study is efficient because it takes time to verify the EECS school requirements and selecting courses.

2. Ellucian Degree Works

Ellucian Degree Works is one of the best competitors I have seen (Figure 2) [Ellucian, 2019]. The key selling point of this system is that it aligns students, advisors, and institutions to a common goal – on-time graduation. Oregon State University allows students to use this system for degree planning.



Student ID	Name	Degree	Major	Level	Classification	Last Audit	Last Refresh
N00048219	Byers, Caleb W.	BS	Business Technology	UG	Sophomore	06/25/2018	07/05/2018 at 10:55 pm

Student Planner for: Byers, Caleb W.

View: Calendar

2017-2018

On-Track	Fall 2017, Total Credits: 14.0	
On-Track	BIOL 101	4.0
On-Track	BIOL 101L	1.0
On-Track	PSYC 140	3.0
On-Track	BT 113	3.0
On-Track	MKRT 123	3.0

Off-Track	Spring 2018, Total Credits: 12.0	
On-Track	ECON 101	3.0
Warning	HIST 201	3.0
On-Track	BT 115	3.0
On-Track	BT 250	3.0
Warning	BT 251 or BT 252	

Figure 2: Ellucian Degree Works Student Planning feature [Ellucian, 2019]

Comparison with Automated Coursework Planner Key Claims:

- **KC #1 (Verification of EECS degree requirements?)** No. The Electrical Engineering and Computer Science school at OSU has its own degree requirements. Although Ellucian allows students to create academic plans, it doesn't include verification of EECS degree requirements.
- **KC #2 (Creates satisfactory Program of Study?)** Perhaps. This system allows the student to create academic plans which achieve some of the goals of the creating a

Program of Study, which students might consider satisfactory—unless these invalid plans are subsequently rejected as an official Program of Study in the EECS school.

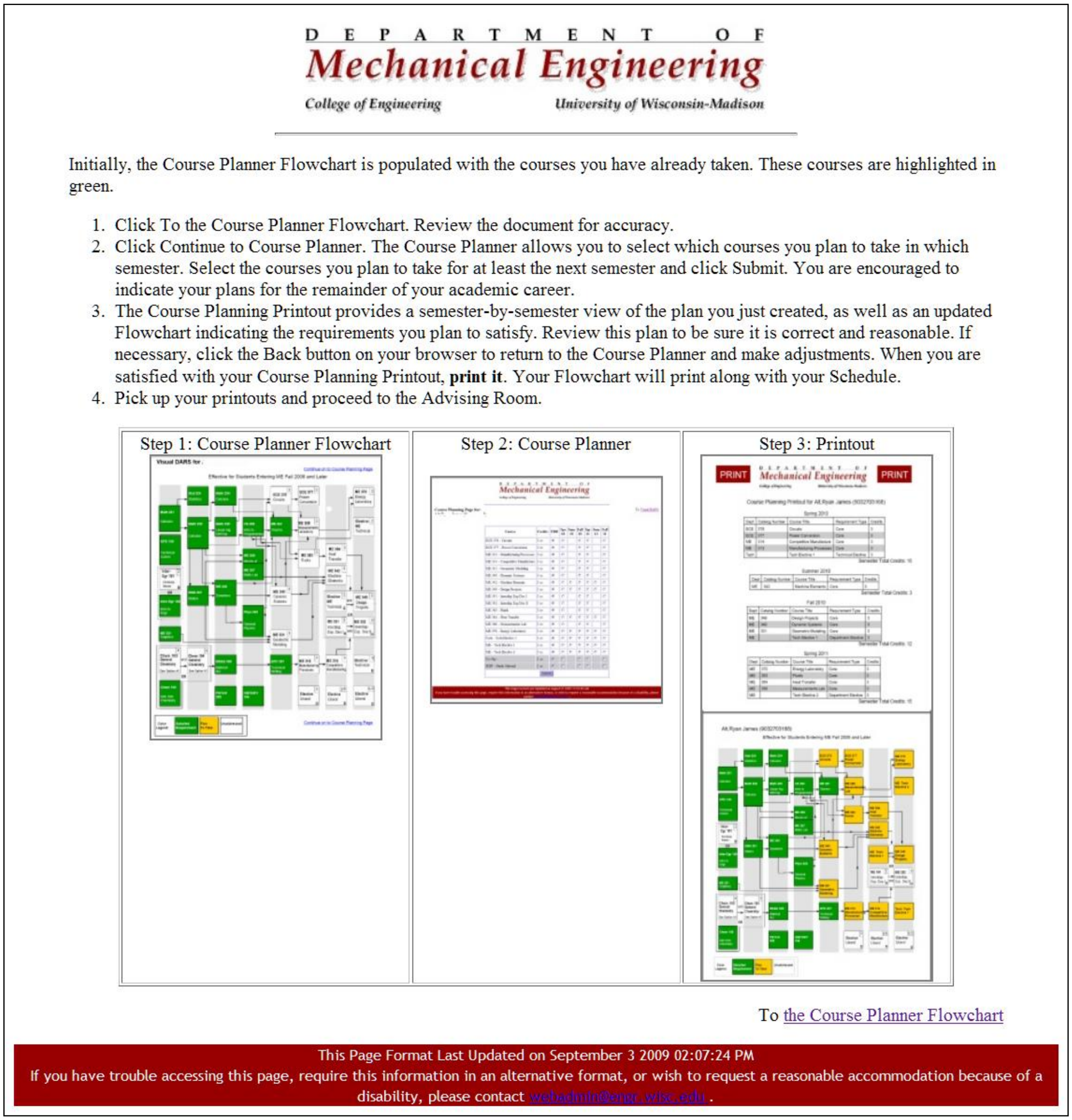
- **KC #3 (Efficient?)** No. The process of creating plans is not efficient. Students still must hunt back and forth among many screens and subsystems using the Ellucian coursework planner. In addition, as noted above, the student still may end up wasting time correcting a plan that does not meet the school's requirements.

3. Course Planner System

The Course Planner System was originally developed for the School of Mechanical Engineering at University of Wisconsin-Madison to provide undergraduate students and their advisors with an easy-to-use, visual tool for planning the students' academic career, and to assist in projecting the need for additional sections of popular courses (Figure 3) [WISC, 2019].

Like the Ellucian Degree Works system, the Course Planner System allows students to indicate the semesters in which they intend to take their courses. Since the students have already decided the courses they are going to take each semester, they have a clear plan for on time graduation.

However, Course Planner System unlike the Ellucian Degree Works system also allows students to see the prerequisites for courses they are interested in. All of the courses and their prerequisites are clearly indicated by a flowchart, and the courses which a student has already taken appear in a different color.



DEPARTMENT OF
Mechanical Engineering
College of Engineering University of Wisconsin-Madison

Initially, the Course Planner Flowchart is populated with the courses you have already taken. These courses are highlighted in green.

1. Click To the Course Planner Flowchart. Review the document for accuracy.
2. Click Continue to Course Planner. The Course Planner allows you to select which courses you plan to take in which semester. Select the courses you plan to take for at least the next semester and click Submit. You are encouraged to indicate your plans for the remainder of your academic career.
3. The Course Planning Printout provides a semester-by-semester view of the plan you just created, as well as an updated Flowchart indicating the requirements you plan to satisfy. Review this plan to be sure it is correct and reasonable. If necessary, click the Back button on your browser to return to the Course Planner and make adjustments. When you are satisfied with your Course Planning Printout, **print it**. Your Flowchart will print along with your Schedule.
4. Pick up your printouts and proceed to the Advising Room.

To the Course Planner Flowchart

This Page Format Last Updated on September 3 2009 02:07:24 PM
If you have trouble accessing this page, require this information in an alternative format, or wish to request a reasonable accommodation because of a disability, please contact webadmin@engr.wisc.edu .

Figure 3: Overview of the Course Planner System [WISC, 2019]

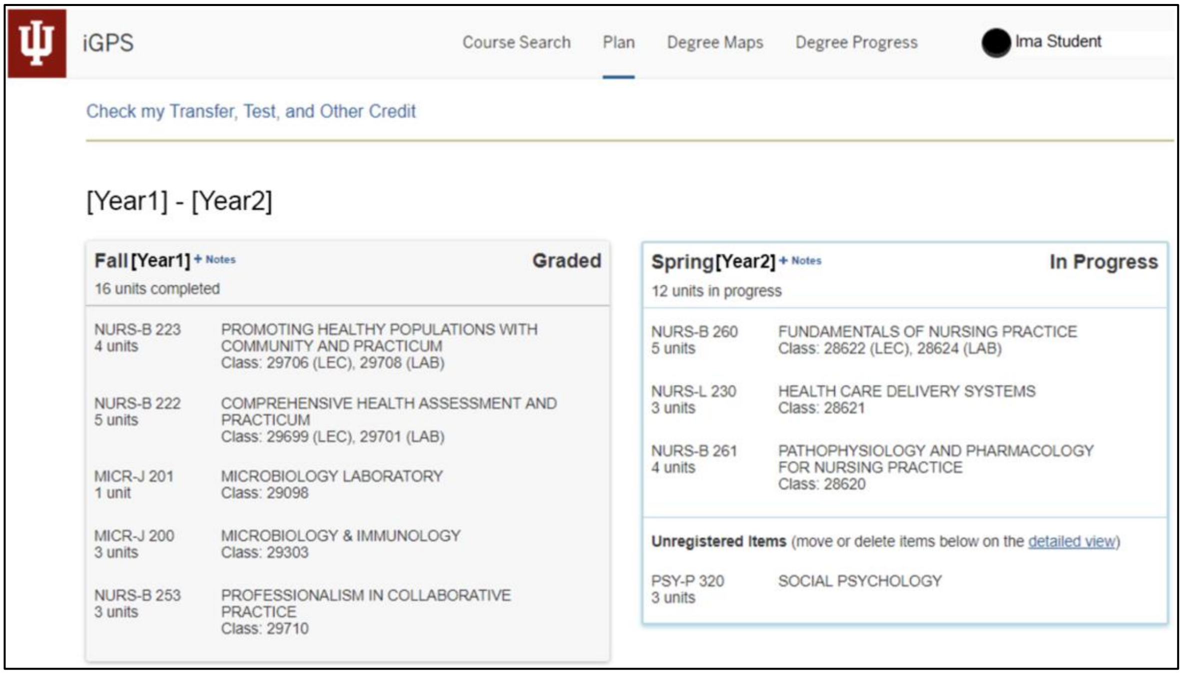
Comparison with Automated Coursework Planner Key Claims:

- **KC #1 (Verification of EECS degree requirements?)** No. The Electrical Engineering and Computer Science school at OSU has its own degree requirements and maybe after several design changes this system can be adapted to verify these requirements, but it would be much easier to create a new system suited for EECS requirement verification.

- **KC #2 (Creates satisfactory Program of Study?)** Yes. This system gives a visual representation of the courses taken and courses yet to be taken, and also indicates the course prerequisites. This system does exceedingly well here. It does achieve the purpose of a Program of Study document but doesn’t actually create one as per the one used in the Electrical Engineering and Computer Science school.
- **KC #3 (Efficient?)** No. This system allows the student to decide the semesters in which the student is interested in taking the course. This allows for creation of a detailed plan but unfortunately it also increases the time taken to create the plan thus making the entire process not so efficient.

4. Interactive Graduation Planning System (iGPS)

Indiana University's Interactive Graduation Planning System (iGPS) is a suite of online tools designed to help students create and follow personalized roadmaps toward degree completion (Figure 4) [Indiana, 2019]. With iGPS, students (and their advisors) can access comprehensive information about degree requirements and upcoming course offerings, and then use that information to devise personalized programs of study that systematically meet those requirements and lead to degree completion in eight semesters.



[Year1] - [Year2]

Fall [Year1] + Notes — **Graded**

16 units completed

NURS-B 223 4 units	PROMOTING HEALTHY POPULATIONS WITH COMMUNITY AND PRACTICUM Class: 29706 (LEC), 29708 (LAB)
NURS-B 222 5 units	COMPREHENSIVE HEALTH ASSESSMENT AND PRACTICUM Class: 29699 (LEC), 29701 (LAB)
MICR-J 201 1 unit	MICROBIOLOGY LABORATORY Class: 29098
MICR-J 200 3 units	MICROBIOLOGY & IMMUNOLOGY Class: 29303
NURS-B 253 3 units	PROFESSIONALISM IN COLLABORATIVE PRACTICE Class: 29710

Spring[Year2] + Notes — **In Progress**

12 units in progress

NURS-B 260 5 units	FUNDAMENTALS OF NURSING PRACTICE Class: 28622 (LEC), 28624 (LAB)
NURS-L 230 3 units	HEALTH CARE DELIVERY SYSTEMS Class: 28621
NURS-B 261 4 units	PATHOPHYSIOLOGY AND PHARMACOLOGY FOR NURSING PRACTICE Class: 28620

Unregistered Items (move or delete items below on the detailed view)

PSY-P 320 3 units	SOCIAL PSYCHOLOGY

Figure 4: Course plan generated by iGPS [Indiana, 2019]

Comparison with Automated Coursework Planner Key Claims:

- **KC #1 (Verification of EECS degree requirements?)** Analogous. This system provides guidance about whether students' plans will achieve their department's requirements. The system is specific to Indiana University, which is fine, but we do not yet have analogous functionality in the School of EECS at Oregon State.
- **KC #2 (Creates satisfactory Program of Study?)** Analogous. This system obviates the need for a printed Program of Study, as it effectively serves as an electronic Program of Study at Indiana University. Again, the system is specific to Indiana University, which is fine, but we do not yet have analogous functionality in the School of EECS at Oregon State.
- **KC #3 (Efficient?)** No. The system has no support for suggesting courses within the major based on personal interests, thus requiring the student to shuttle around among other

sources to make decisions. It also requires students to plan out each semester one-by-one, rather than planning out a Program of Study (i.e., just a list of courses).

The general trend in the competition:

Similar to the systems mentioned above, almost all the universities use some kind of degree planning system like the Smart Planner [Burgundy, 2019], and Course Schedule Planner [Rutgers, 2019]. These systems allow students to create a plan which gives them a sense of direction while pursuing their degree. These systems do a great job at allowing students and their advisors to see their progress. However, these systems do not fulfill the key claims defined in Section 2 of this document. The next section will explain how, in contrast, GradPlan provides support for selecting courses, verifying that the plan satisfies requirements, and generating a final document for gathering signatures.

4. GradPlan Prototype Description

The GradPlan prototype is designed to enable MS in CS students in the Electrical Engineering and Computer Science school to create their Program of Study quickly and without any hassles. In this section, I will describe the system components, and their functioning in detail.

This prototype is a web application built using the client-server system architecture such that it can work on any web browser. This choice of technology ensures that a simple laptop with a web browser would suffice for using this system, and no specialized software and hardware is required.

Figure 5 gives an overview of the system. The system can be broken down into three major components: user interface, the server-side, and the database.

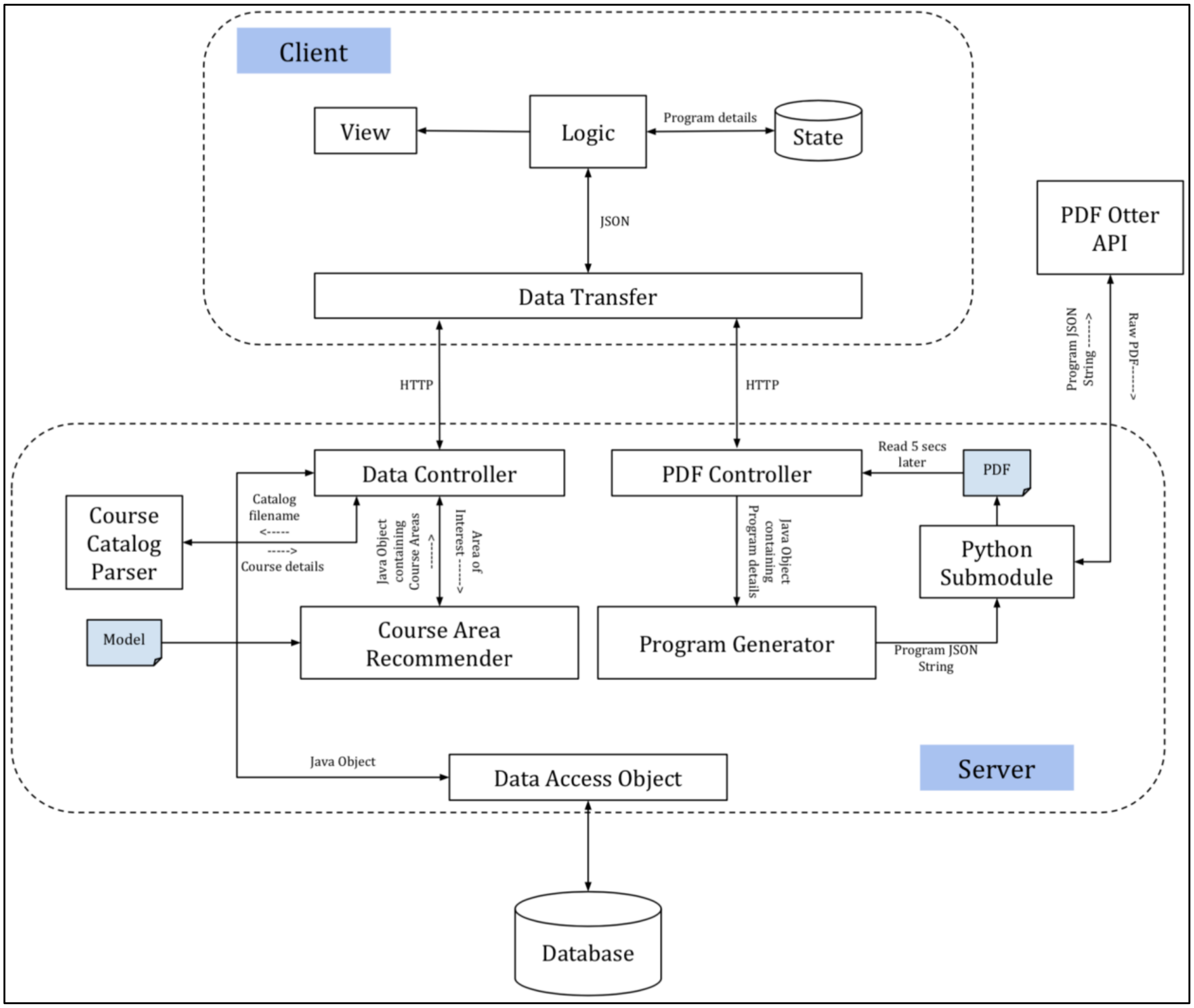
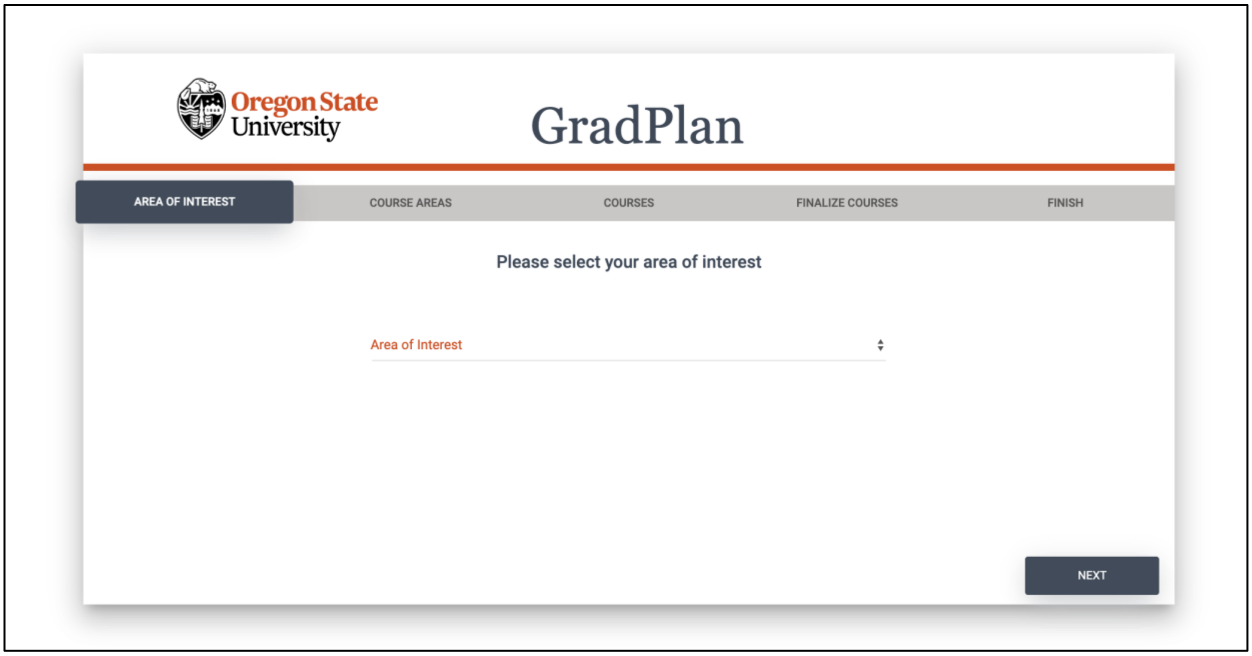


Figure 5: This is the system architecture diagram showing various components of the GradPlan system

1. User Interface

The user interface was designed to facilitate decision-making by the user, to divide the big task of creating a Program of Study into smaller steps, and to ensure that these small steps are completed in a specific sequence, resulting in a valid Program of Study. Having a wizard interface seemed like a good choice here. Figures 6 through 11 illustrate the various stages of this wizard user interface.



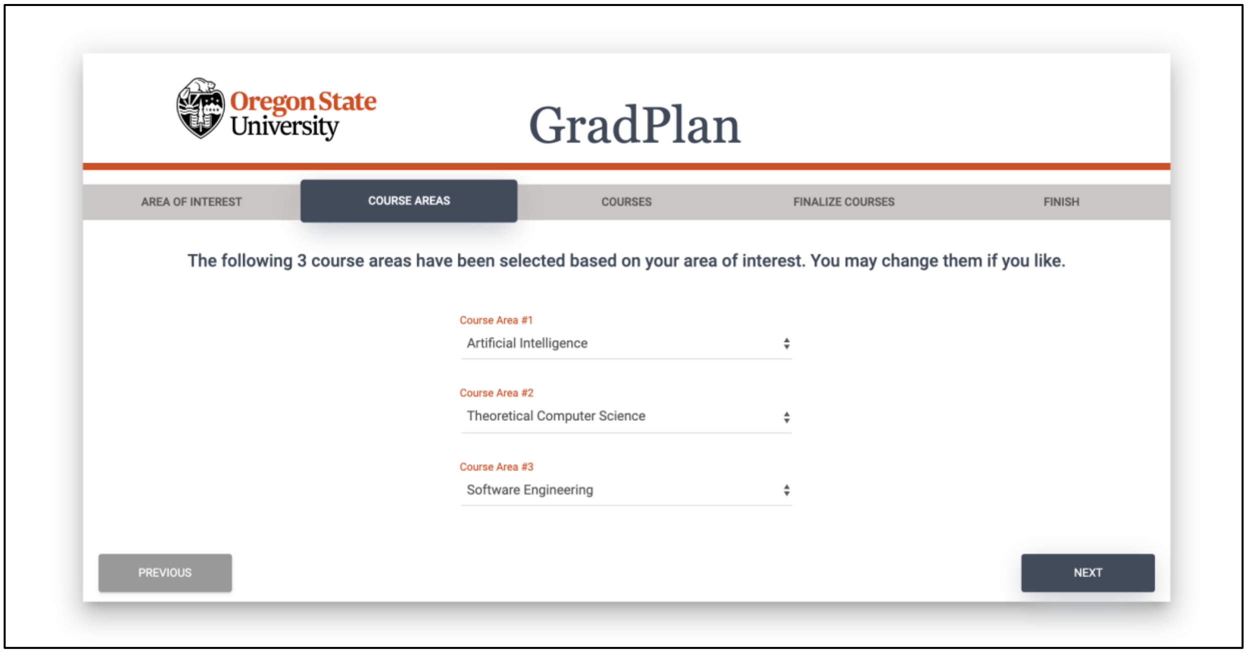
Please select your area of interest

Figure 6: Step 1 - Specify Area of Interest

The following choices for Area of Interest are available:

- Data Science
- Artificial Intelligence
- Cybersecurity
- Networking and Computer Systems
- Software Engineering
- Human Computer Interaction
- Programming Languages
- Theoretical Computer Science
- Computer Graphics and Visualization

These are the typical areas in computer science a MS in CS student in the school of EECS might be interested in.



The following 3 course areas have been selected based on your area of interest. You may change them if you like.

Course Area #1: Artificial Intelligence

Course Area #2: Theoretical Computer Science

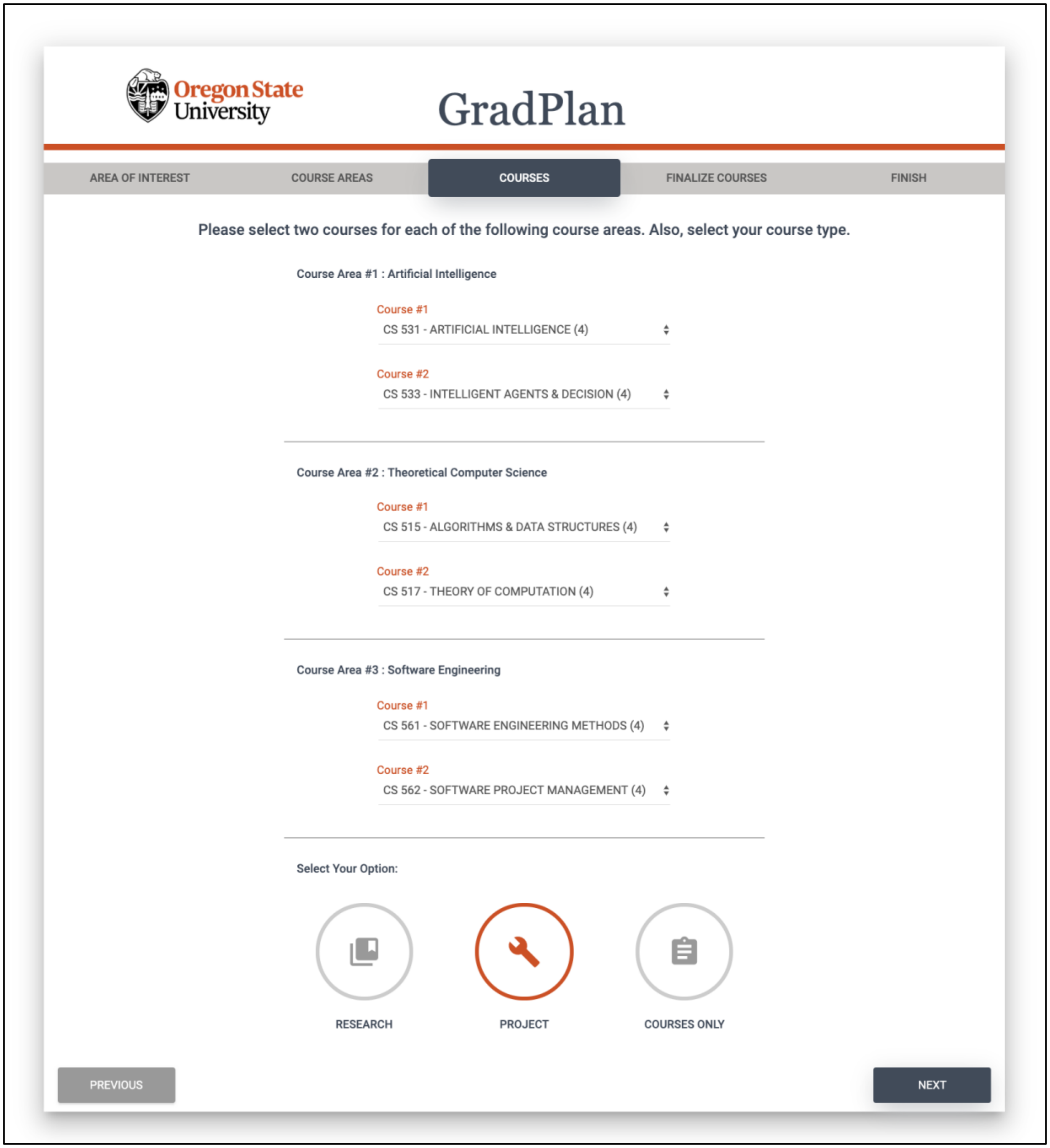
Course Area #3: Software Engineering

Figure 7: Step 2 - System suggests three course areas based on the student's interest. Student reviews/edits these course areas.

The following choices for course areas are available:

- Theoretical Computer Science
- Artificial Intelligence
- Computer Systems
- Programming Languages
- Software Engineering
- Human Computer Interaction
- Computer Vision and Graphics
- Statistics
- Business
- Robotics

The school of EECS offers courses in these course areas [EECS, 2019]. Students may also choose course areas not offered by the school of EECS with their advisor's permission.



GradPlan

Please select two courses for each of the following course areas. Also, select your course type.

Course Area #1 : Artificial Intelligence

Course #1: CS 531 - ARTIFICIAL INTELLIGENCE (4)

Course #2: CS 533 - INTELLIGENT AGENTS & DECISION (4)

Course Area #2 : Theoretical Computer Science

Course #1: CS 515 - ALGORITHMS & DATA STRUCTURES (4)

Course #2: CS 517 - THEORY OF COMPUTATION (4)

Course Area #3 : Software Engineering

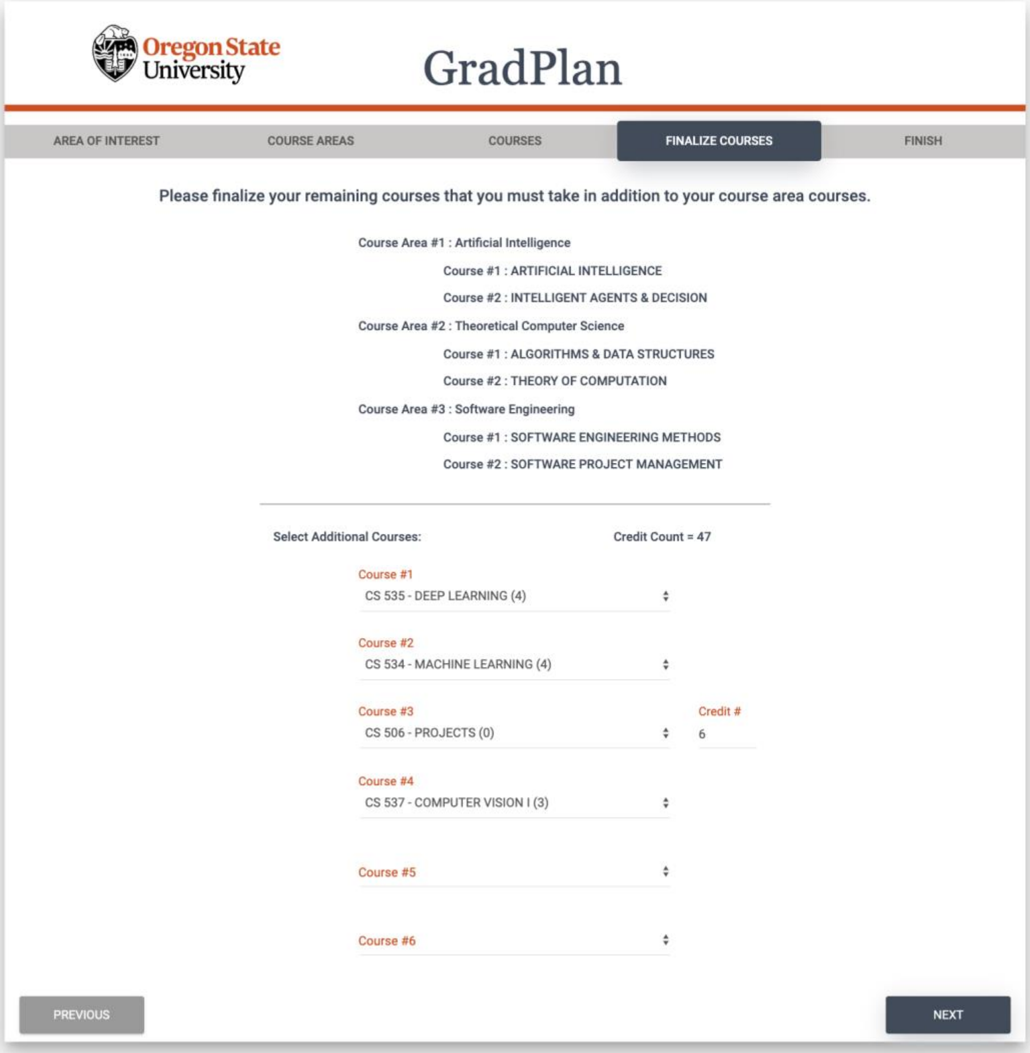
Course #1: CS 561 - SOFTWARE ENGINEERING METHODS (4)

Course #2: CS 562 - SOFTWARE PROJECT MANAGEMENT (4)

Select Your Option: RESEARCH [✓ PROJECT] COURSES ONLY

Figure 8: Step 3 - Select courses for each course area

The school of Electrical Engineering and Computer Science has a list of courses under each course area, out of which any two can be selected to meet a course area requirement [EECS, 2019]. Also, students can choose if their MS in CS degree involves a thesis or a project by selecting Research or the Project icon. Courses Only option allows students to select a Meng degree, which is a course work only program.



Please finalize your remaining courses that you must take in addition to your course area courses.

Course Area #1 : Artificial Intelligence

Course #1 : ARTIFICIAL INTELLIGENCE

Course #2 : INTELLIGENT AGENTS & DECISION

Course Area #2 : Theoretical Computer Science

Course #1 : ALGORITHMS & DATA STRUCTURES

Course #2 : THEORY OF COMPUTATION

Course Area #3 : Software Engineering

Course #1 : SOFTWARE ENGINEERING METHODS

Course #2 : SOFTWARE PROJECT MANAGEMENT

Select Additional Courses: Credit Count = 47

Course #1
CS 535 - DEEP LEARNING (4)

Course #2
CS 534 - MACHINE LEARNING (4)

Course #3
CS 506 - PROJECTS (0)

Credit #
6

Course #4
CS 537 - COMPUTER VISION I (3)

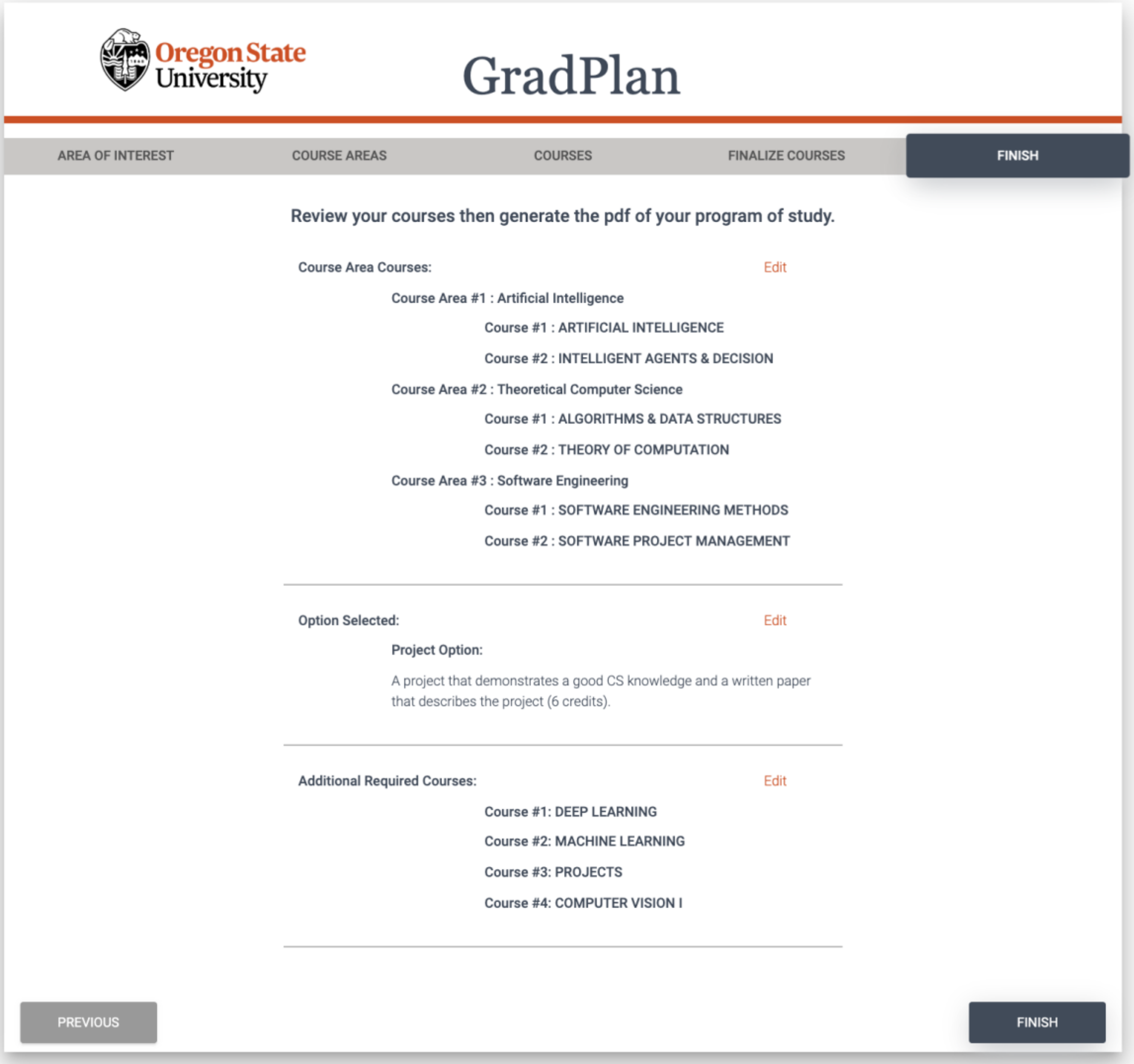
Course #5

Course #6

Figure 9: Step 4 - Select additional courses

After Step 3, students have selected 6 courses and indicated their thesis or non-thesis choice. That means they have 24 credits from the courses they have selected, plus 9 or 6 credits depending on their thesis or non-thesis choice. However, the school of Electrical Engineering and Computer

Science requires students to have at least 45 credits in total, which means there's lot of choice in the remaining credits. The additional courses section allows students to select their remaining credits.



Review your courses then generate the pdf of your program of study.

Course Area Courses: Edit

Course Area #1 : Artificial Intelligence

Course #1 : ARTIFICIAL INTELLIGENCE

Course #2 : INTELLIGENT AGENTS & DECISION

Course Area #2 : Theoretical Computer Science

Course #1 : ALGORITHMS & DATA STRUCTURES

Course #2 : THEORY OF COMPUTATION

Course Area #3 : Software Engineering

Course #1 : SOFTWARE ENGINEERING METHODS

Course #2 : SOFTWARE PROJECT MANAGEMENT

Option Selected: Edit

Project Option:

A project that demonstrates a good CS knowledge and a written paper that describes the project (6 credits).

Additional Required Courses: Edit

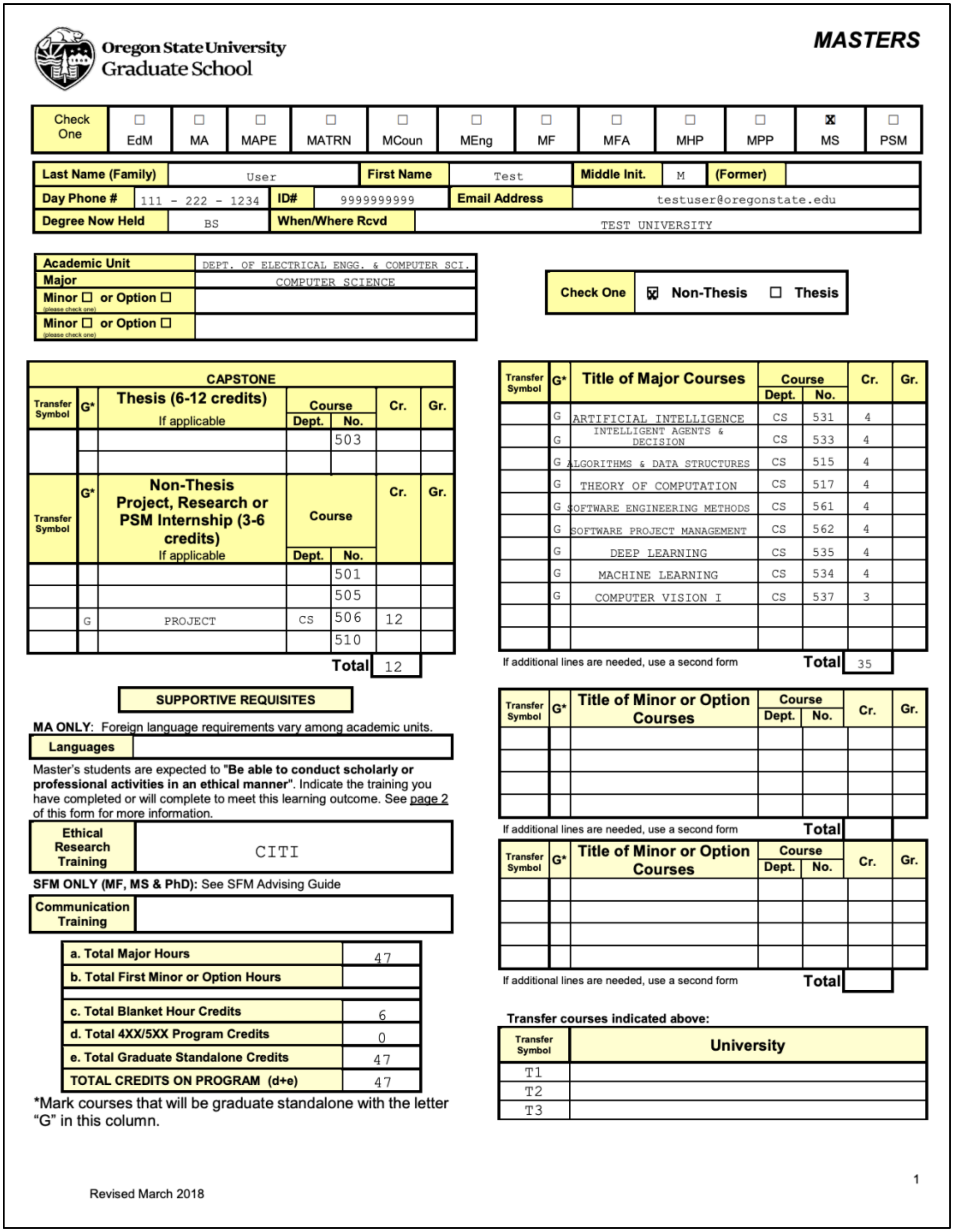
Course #1: DEEP LEARNING

Course #2: MACHINE LEARNING

Course #3: PROJECTS

Course #4: COMPUTER VISION I

Figure 10: Step 5 - Review course selection & Finish



Oregon State University
Graduate School

MASTERS

Check One	□ EdM	□ MA	□ MAPE	□ MATRN	□ MCoun	□ MEng	□ MF	□ MFA	□ MHP	□ MPP	☒ MS	□ PSM

Last Name (Family)	User	First Name	Test	Middle Init.	M	(Former)	
Day Phone #	111 - 222 - 1234	ID#	9999999999	Email Address	testuser@oregonstate.edu		
Degree Now Held	BS	When/Where Rcvd	TEST UNIVERSITY				

Academic Unit	DEPT. OF ELECTRICAL ENGG. & COMPUTER SCI.
Major	COMPUTER SCIENCE
Minor □ or Option □ (please check one)	
Minor □ or Option □ (please check one)	

Check One ☒ Non-Thesis □ Thesis

CAPSTONE

Transfer Symbol	G*	Thesis (6-12 credits) If applicable	Course Dept.	Course No.	Cr.	Gr.
				503		

Transfer Symbol	G*	Non-Thesis Project, Research or PSM Internship (3-6 credits) If applicable	Course Dept.	Course No.	Cr.	Gr.
				501		
				505		
	G	PROJECT	CS	506	12	
				510		

Total 12

Transfer Symbol	G*	Title of Major Courses	Course Dept.	Course No.	Cr.	Gr.
	G	ARTIFICIAL INTELLIGENCE	CS	531	4	
	G	INTELLIGENT AGENTS & DECISION	CS	533	4	
	G	ALGORITHMS & DATA STRUCTURES	CS	515	4	
	G	THEORY OF COMPUTATION	CS	517	4	
	G	SOFTWARE ENGINEERING METHODS	CS	561	4	
	G	SOFTWARE PROJECT MANAGEMENT	CS	562	4	
	G	DEEP LEARNING	CS	535	4	
	G	MACHINE LEARNING	CS	534	4	
	G	COMPUTER VISION I	CS	537	3	

If additional lines are needed, use a second form Total 35

SUPPORTIVE REQUISITES

MA ONLY: Foreign language requirements vary among academic units.

Languages	

Master's students are expected to **"Be able to conduct scholarly or professional activities in an ethical manner"**. Indicate the training you have completed or will complete to meet this learning outcome. See page 2 of this form for more information.

Ethical Research Training	CITI

SFM ONLY (MF, MS & PhD): See SFM Advising Guide

Communication Training	

a. Total Major Hours	47
b. Total First Minor or Option Hours	
c. Total Blanket Hour Credits	6
d. Total 4XX/5XX Program Credits	0
e. Total Graduate Standalone Credits	47
TOTAL CREDITS ON PROGRAM (d+e)	47

*Mark courses that will be graduate standalone with the letter "G" in this column.

Transfer Symbol	G*	Title of Minor or Option Courses	Course Dept.	Course No.	Cr.	Gr.

If additional lines are needed, use a second form Total

Transfer Symbol	G*	Title of Minor or Option Courses	Course Dept.	Course No.	Cr.	Gr.

If additional lines are needed, use a second form Total

Transfer courses indicated above:

Transfer Symbol	University
T1	
T2	
T3	

Revised March 2018

1

Figure 11: Final Output - Program of Study PDF

The implementation of the user interface can be divided into following components:

a. **View**

   The wizard interface implementation was based off of Creative Tim's material bootstrap wizard. The view component of the user interface deals with the look and feel of the system.

b. **State**

   This is the JavaScript object which stores the various details pertaining to the Program of Study while it's being created. Any selection or changes in course areas or courses, results in updates to this object. This simplifies checking whether or not the Program of Study meets the EECS requirements, and also ensures consistency as the user navigates back and forth in the wizard interface.

c. **Data Transfer**

   This module contains the synchronous and asynchronous implementations of AJAX calls, which send and receive JSON data from the Controllers on the server.

2. Server-Side

The server-side components run on the application server. The GradPlan system is designed to work with Apache Tomcat version 9.0. The system can be configured and deployed on other application servers too; however, I'm using Tomcat because it is incredibly lightweight and provides quick load and redeploy times to support rapid iteration during development.

The following sub-sections describe the functioning of the various server-side components.

a. Controllers

The two controllers namely the Data Controller and the PDF Controller are responsible for handing the HTTP requests from the User Interface.

The Data Controller receives HTTP Post requests:

1. for getting the course areas relevant to the user's area of interest
2. to fetch the courses for the selected course areas
3. to fetch all the courses in the database

It also receives an HTTP Get request to parse the course catalog which is stored in the JSON format on the file system.

The PDF Controller receives the Post request containing the validated course selection from the User Interface when the user clicks finish. It then calls other server-side components mentioned below to create the final PDF document and then pushes the PDF to the User Interface.

b. Course Catalog Parser

The Course Catalog Parser provides a way to automatically initialize the GradPlan database with course details from the Oregon State University course catalog. It receives a JSON string containing course details and the school name. It parses this string and maps various courses into their course areas as mentioned on the Electrical Engineering and Computer Science school webpage. It also fetches additional details such as number of credits and course graduate standing (which indicates slash courses) from the Oregon State

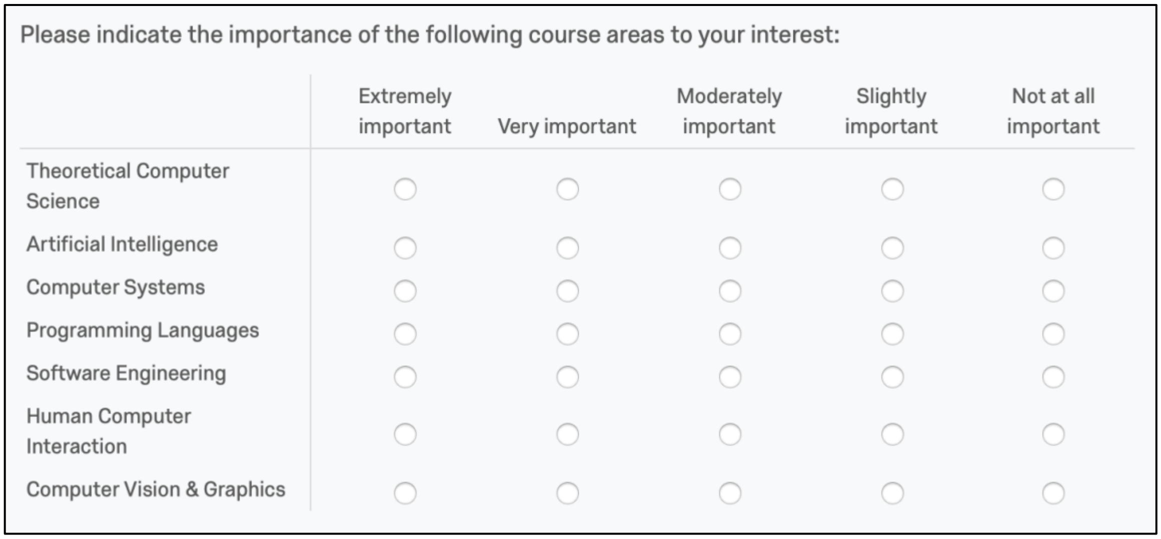
University course catalog. Internally within GradPlan, it finally returns a Java object containing the course details which is then put into the database.

c. Course Area Recommender

The Course Area Recommender is responsible for suggesting course areas based on the user's selected area of interest. It uses a simple model created based on the results from a survey that I conducted in Winter 2019 to gather inputs from faculty and graduate students in the School of EECS. The details pertaining to this survey and the model generation are mentioned in the next paragraph. This model spits out the three course areas in the order of importance for a given area of interest. The count is set to three because students need to select three course areas and select two courses from each area [EECS, 2019].

The survey mentioned above asked the graduate students and faculty to select their area of interest, and then indicate the importance of each of the course areas on a scale of 1 to 5 with 1 being Not at all important and 5 being extremely important, as shown in Figure 12. This data was then grouped on the basis of fields of interests, and the points indicating importance for each course area were averaged across survey respondents. The three course areas with maximum points are considered relevant for the mentioned field of interest.

In short, the course area recommender takes the user's selected interest and, in return, recommends the three course areas that were considered most important among students and faculty who completed the survey.



Please indicate the importance of the following course areas to your interest:

	Extremely important	Very important	Moderately important	Slightly important	Not at all important
Theoretical Computer Science	○	○	○	○	○
Artificial Intelligence	○	○	○	○	○
Computer Systems	○	○	○	○	○
Programming Languages	○	○	○	○	○
Software Engineering	○	○	○	○	○
Human Computer Interaction	○	○	○	○	○
Computer Vision & Graphics	○	○	○	○	○

Figure 12: Survey question asking students and faculty to rate the importance of each course area

d. Program of Study Validation

In order to generate a valid Program of Study, the GradPlan system has some rules built into it. These rules ensure that the Program of Study generated by the GradPlan system meets the degree requirements set by the school of Electrical Engineering and Computer Science.

The validation rules are implemented at every step mentioned above. The user is unable to proceed to the next step if the rules for the current step are not met. For example, if the user selects the same course area or course twice at a given step, an alert is shown when the user clicks next. The validation rules can be categorized as follows:

1. *<u>Two courses from three course areas requirement</u>*

   At step 3 wherein the user selects the course area courses, only two selections can be made for each course area. Also, the courses listed in the drop down are valid courses for that course area. These dropdowns cannot be left blank. This ensures that the user selects two valid courses for each course area before proceeding to the next step.

2. *Thesis or Project choice*

   The user cannot proceed to the next step until this choice is made. Selection of thesis causes the current credit count to increase by 9, and selection of project increases the current credit count by 6. This selection causes the appropriate course (CS 503 or CS 506) to be listed on the Program of Study.

3. *Blanket credits limitation*

   In addition to the thesis or project credits, students can select other CS 50X credits also called as blanket credits. However, maximum of six such blanket credits can be added on the Program of Study. In the finalize courses step, if more than six blanket credits are found, the user gets an alert when next is clicked.

4. *Slash courses limitation*

   There are some graduate courses (5XX) which are also upper level undergrad courses (4XX). Such courses are called slash courses. The number of slash courses cannot be more that 50% of the total credit count. If the student selects more than 50% slash course credits, an alert is shown, and the student cannot proceed to the next step.

5. *Total credit count requirement*

   At least 45 graduate credits are required on the Program of Study. The GradPlan systems sums up the credits for each course selected (only graduate courses can be selected using GradPlan) and verifies if this count is satisfied.

e. Program Generator and the Python Submodule

The Program Generator is responsible for constructing the JSON string summarizing the student's choices, which will be subsequently passed, below, to a 3rd-party library for use in generating a PDF. This object contains the program details such as courses selected, credit count, etc. These details have already been validated to meet the Electrical Engineering and Computer Science school course requirements, as explained above. The JSON string contains the actual text required to fill out the placeholders on the Program of Study PDF template.

The Program Generator launches a Python script tasked with communicating with PDF Otter API and sends it the JSON string using Sockets. The Python script is used as an intermediary because it allows easy access to the PDF Otter API, which actually writes the PDF. The Python script calls the external API and receives a PDF document which is stored on the server temporarily, ready to be read and uploaded by the PDF Controller.

f. PDF Otter

The PDF Otter is an external API used to fill out PDF templates [Otter, 2019]. The PDF template used by the GradPlan system is visually identical to the Program of Study document provided by the Electrical Engineering and Computer Science school. The placeholders for text in the template for user input are identified by keys. This API accepts the JSON string created by the Program Generator and uses the text against the keys to populate the required fields and then returns a PDF document.

3. Database

The database contains the course catalog details parsed by the Catalog Parser. I am using MySQL database for this system. The database schema is simple, since it consists of a single table, which is enough to store the course details. User-specific data does not get stored.

Table Name: course_master

Columns Names: course_id, course_code, course_title, course_crn, credits, school,
course_area, term_offered, is_grad_course

Primary Key: course_id

5. Laboratory Study Evaluation

This section describes how I evaluated the GradPlan system by gathering and analyzing data from laboratory study participants. This study investigated how well the GradPlan system performed as compared to manually filling out the Program of Study PDF template.

Research Questions:

This study attempted to answer the following three questions:

1. Does GradPlan generate a satisfactory Program of Study?
2. Does GradPlan generate a Program of Study that is equally satisfactory as manually filling out a PDF template?
3. Does GradPlan save time as compared to manually filling out a PDF template?

Free-response questions were utilized to understand ways in which the existing procedure or GradPlan were unsatisfactory or needed improvement.

Participants:

I recruited 16 individuals from the MS in Computer Science program in the School of EECS. Only MS in CS students were recruited because the GradPlan system is built with only MS in Computer Science students in mind. All the students involved in this study were in their first year who were yet to create their Program of Study. This ensured that the students were creating their Program of Study for the first time. Including students who had already been through this process would have rendered the results invalid because their prior knowledge of the course selection would help them create the Program of Study in less time.

Procedure:

Once the individuals were selected, they were randomly put into two counter-balanced groups wherein each participant performed two tasks in a certain order:

Group A: This group created the Program of Study using the GradPlan system first, and then filled out the Program of Study PDF template

Group B: This group filled out the Program of Study PDF template first, and then created their Program of Study using the GradPlan system

The Program of Study document created as the first task was discarded, and the participants were asked to create the document again from scratch as the second task. Dividing the participants into these two groups ensures that none of the two systems had an unfair advantage. For example, allowing all the participants to manually fill out the PDF template first would maybe cause them to create the Program of Study faster using GradPlan system, and this would be an unfair advantage to the GradPlan system. Also, the same machine and the same web browser was used by all the participants in this study.

Prior to the tasks, all the participants were given a brief introduction about the process they were going to follow. The introduction included a small description of the tasks they were about to do, a quick glance at the browser tabs containing the EECS degree requirements webpage, and the course catalog which could be used to view course information while manually filling out the PDF template. They did not get to try out GradPlan during the introduction. Thus, if anything, this brief introduction biased the evaluation in favor of the Program of Study PDF template rather than in favor of GradPlan.

Data Acquisition:

The time taken in seconds by each participant to complete each task was recorded after each task ended. At the end of both the tasks, each participant was given a survey as shown in the Table 1.

Table 1: Evaluation questionnaire

In the following questions, please indicate the extent to which you agree or disagree with the statements about the GradPlan system.
GradPlan generates a satisfactory Program of Study • Strongly agree • Agree • Disagree • Strongly disagree
GradPlan generates a Program of Study that is equally satisfactory as manually filling out a PDF template • Strongly agree • Agree • Disagree • Strongly disagree
GradPlan saves time compared to manually filling out a PDF template • Strongly agree • Agree • Disagree • Strongly disagree
What do you like best about GradPlan?
What are your suggestions for improving GradPlan?

6. Results

The laboratory study participants evaluated GradPlan system against manually filling out the PDF Program of Study template. This section details the results and the analysis.

The average time required for manually filling out the PDF template was 581 seconds, and the time required for creating a Program of Study using the GradPlan system was 227 seconds on average. The one-tailed Wilcoxon Signed-Rank Test between these was significant at alpha = 0.05 (p-value: 0.001, Z-statistic: -3.06, N=16). This result indicates that the GradPlan system saves time compared to manually filling out the Program of Study.

All sixteen survey participants agreed or strongly agreed that GradPlan system generates a satisfactory Program of Study (Table 2, below). In addition, I asked the EECS Graduate Coordinator, who is well versed with the EECS school degree requirements, for his opinions on my prototype. He gave a few suggestions but noted that the overall system was satisfactory. The suggestions mentioned by him and the survey participants are given in the following paragraphs. The feedback from users and the Graduate Coordinator indicate that GradPlan generates a satisfactory Program of Study.

Fifteen of the sixteen survey participants agreed that GradPlan system generates a Program of Study that is equally satisfactory as manually filling out a Program of Study. One of the participants disagreed, due to some difficulties with selecting a variant of CS 519. The complication is that CS 519 seems like a single course, but it’s actually a group of special topic courses that have a high variability in their offering. Not all CS 519 courses can be counted towards graduation, and if this is the case then they cannot be a part of the Program of Study. The current system prototype does not distinguish among versions of CS 519 that can and cannot be counted toward graduation. The Graduate Coordinator (above) also noted that the system should provide more specific support when allowing the students to choose among different versions of CS 519. Aside from this one issue, the consensus of the users was that the GradPlan system generates a Program of Study that is equally satisfactory as manually filling out a Program of Study.

Table 2: Summary of evaluation questionnaire results

Question	Strongly Disagree	Disagree	Agree	Strongly Agree
GradPlan generates a satisfactory Program of Study	0	0	7	9
GradPlan generates a Program of Study that is equally satisfactory as manually filling out a PDF template	0	1	3	12
GradPlan saves time compared to manually filling out a PDF template	0	0	1	15

7. Conclusion

The GradPlan system enables MS in CS students to create a Program of Study document in half the time it takes to manually create one. In addition, users agreed that the GradPlan system allows them to create a valid Program of Study that is just as satisfactory as manually filling out the PDF template. It includes suggestions of course areas based on area of interest, it automatically checks that the student's choices satisfy program requirements, and it automatically generates a PDF suitable for gathering signatures.

Overall, this system makes a tedious, boring, and time-consuming task easy, quick and hassle free. This software could be incorporated into the OSU's course registration system and linked with the course catalog to streamline the course selection process. This would allow the students to easily register for the courses they have on their Program of Study.

The idea of suggesting course areas based on students' field of interest could be expanded further to have a full-fledged course recommender system that can specify the actual courses instead of just course areas, by taking into account not only the students' field of interest but also students' future goals such as working in the industry or academia, ratings for the previous offerings of the course at the university, course instructor ratings, etc. The survey-based approach for developing a recommendation model could also be used to expand the system so that it also supports the ECE MS program, as well as the PhD program; these programs have different requirements than the MS CS program, so additional validation rules also would need to be implemented.

In conclusion, the GradPlan system is an innovation that opens up new directions for addressing Oregon State University students' need for selecting the right courses. It performed admirably in terms of usability. Equipping it with new features and integrating it with other systems such as the university course registration system could lead to a new robust system for fully automating the coursework selection and degree planning process.

8. References

1. OSU (2019) *Program of Study | Graduate School | Oregon State University*, visited 5 August 2018, https://gradschool.oregonstate.edu/progress/program-study
2. Ellucian (2019) *Ellucian Degree Works*, visited 17 July 2019, https://www.ellucian.com/solutions/ellucian-degree-works
3. WISC (2019) *Course Planner System*, visited 20 July 2019, https://sd3.engr.wisc.edu/documentation/course_planner_system.docx
4. Indiana (2019) *About Interactive Graduation Planning Components*, visited 13 August 2019, https://kb.iu.edu/d/igps
5. Burgundy (2019) *Smart Planner – The Burgundy Group*, visited 10 August 2019, https://www.tbginc.com/our-products/smart-planner
6. Rutgers (2019) *Course Schedule Planner – Rutgers University*, visited 13 August 2019, https://sims.rutgers.edu/csp
7. EECS (2019) *CS Program | Electrical Engineering and Computer Science | Oregon State University*, visited 5 August 2018, https://eecs.oregonstate.edu/academics/graduate-programs/cs
8. Otter (2019) *API for Filling in PDFs – PDF Otter*, visited 5 August 2018, https://www.pdfotter.com

9. Acknowledgements

Calvin Hughes, Christopher Scaffidi, Claudia Hilderbrand, computer science graduate students and faculty in the school of EECS who took the anonymous survey, Lara Letaw, and the sixteen anonymous lab study participants.

Thank you for helping me through.