1-28-2015

Development of a Modern Computational Infrastructure Around University Curricula

Jarred Kozlick

Jarred M. Kozlick

Candidate

Electrical & Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Gregory L. Heileman, Chairperson

Dr. Chaouki T. Abdallah

Dr. Terry Babbitt

Dr. Christopher Lamb

Development of a Modern Computational Infrastructure Around University Curricula

by

Jarred Kozlick

B.S., University of New Mexico 2012

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2014

Dedication

This thesis is dedicated to my wife Meghan, who has been an endless source of love and support.

Acknowledgments

I would like to thank my committee members, Prof. Gregory L. Heileman, Prof. Chaouki T. Abdallah, Dr. Terry Babbit and to Dr. Christopher Lamb. I would also like to extend special gratitude to Prof. Heileman for his advice and advocacy as my advisor. Much gratitude is also extended to Ahmad Slim, Elias Lopez, Michael Hickman, Kevin Warne, Wisam Quais Al Doroubi and the rest of the "Wild Weasels" with whom I worked to build the first version of this application.

Development of a Modern Computational Infrastructure Around University Curricula

by

Jarred Kozlick

B.S., University of New Mexico 2012

M.S., Computer Engineering, University of New Mexico, 2014

Abstract

This thesis investigates solutions required to construct a modern computational infrastructure around the delivery of student degree plans as a web service. The initial system consisted of a data store organized around an SQL relational database. In this thesis, subsequent new architectures are explored that allow student and institutional analytics to be seamlessly integrated into the application. The foundation of these new architectures involve data representation and storage, and several NoSQL options are explored for performing this task. Implementing the data stores with these technologies allowed the data to be stored in its native structure and greatly facilitates extensibility of the data model and the capability for analytics. This thesis describes some of the benefits of using NoSQL data stores in this domain, including increased ease of analyzing and processing data.

Contents

List of Figures

List of Figures

Glossary

ACID (Atomicity, Consistency, Isolation, Durability) A set of properties that ensures that relational database transactions execute properly.

BASE (Basically Available, Soft state, Eventual consistency) Database transactional model favored by NoSQL databases that allows system reliability.

Binary JSON Binary-encoded serialization of JSON-like documents.

CAP Theorem Theorem in computer science stating that a system can not provide consistency, availability, and partition tolerance simultaneously.

Cypher Declarative language used to query Neo4j graph data stores.

D3.js (Data Driven Documents or just **D3)** JavaScript library that uses digital data to drive creation dynamic interactive web based data visualizations.

Entity Relationship Diagram (ERD) Data model used to describe the data domain of an application, usually lending itself to be implemented in a relational database.

Git Open-source distributed version control system.

Hypertext Transfer Protocol (HTTP) Application protocol for distributed systems.

JavaScript Object Notation (JSON) Lightweight data-exchange language designed to be human readable.

Model View Controller (MVC) Software architectural pattern widely chosen for implementing user interfaces.

MongoDB Document oriented-database that utilizes a dynamic schema and JSON-like storage model.

Neo4j Disk-based transactional persistence engine that stores data natively as graphs.

NoSQL Databases designed to meet requirements of web deployed applications; can be broadly described as non-relational, distributed, open-source and horizontally scalable.

Object Relational Mapper (ORM) Programing technique used to persist object-oriented code in a relational database.

Platform-as-a-Service (PaaS) Cloud computing platform that provides a solution stack to the developer, while alleviating the need to manage underlying hardware and software systems.

Relational Database Management System (RDBMS) Database management system that stores information in the form of related tables.

Ruby on Rails (Rails) Open source web applications framework written in Ruby.

Single Table Inheritance (STI) Used to represent an inheritance hierarchy using a single table in a database that has columns for all the fields of the inherited classes.

Structured Query Language (SQL) Declarative programming language used to manage information stored in a relational database system.

Unified Modeling Language (UML) Modeling language that provides a standardized way of visualizing the data domain of a system.

Web Applications Framework Software framework that aims to alleviate overhead associated with web development by providing libraries for common tasks.

YAML Human-readable data serialization format.

Chapter 1

Introduction

At the University of New Mexico, in order for a student to be awarded a specific degree, he or she must satisfactorily complete a given set of courses. This statement sounds on its face both obvious and unusual. However, up until recently, if a degree seeker wished to know what the given set of courses for a degree were, it may have taken a bit of work to track these requirements down. As it stood, there was no centralized location for a student to find information on what courses were needed to complete a degree, or to compare the requirements of different degrees. The responsibility of creating the plans for the degrees and making them available lied solely with personnel in the university's many departments and was supported by the university's advising office via a standard spreadsheet template that departments were asked to use.

The first steps to resolve this problem were taken in August of 2013 when team members at the Electrical & Computer Engineering Department's Informatics Laboratory created and deployed the first version of a website that allowed students access to all of the degrees and their associated plans available at the University of New Mexico. This website allows anyone to view specific degree plans by college or

alphabetically, or to explore degree plans by interest areas. An option is given for the user to compare degree plans offered by description, admission requirements, and career opportunities.

Providing this information in a centralized, concise manner provided value to those already attending the university as well as potential students. While this in itself is a respectable goal, it would be beneficial for the university to use this system as the foundation to build a modern computational framework. This new architecture will allow the university to perform student tracking, student analytics, and institutional analytics related to specific programs.

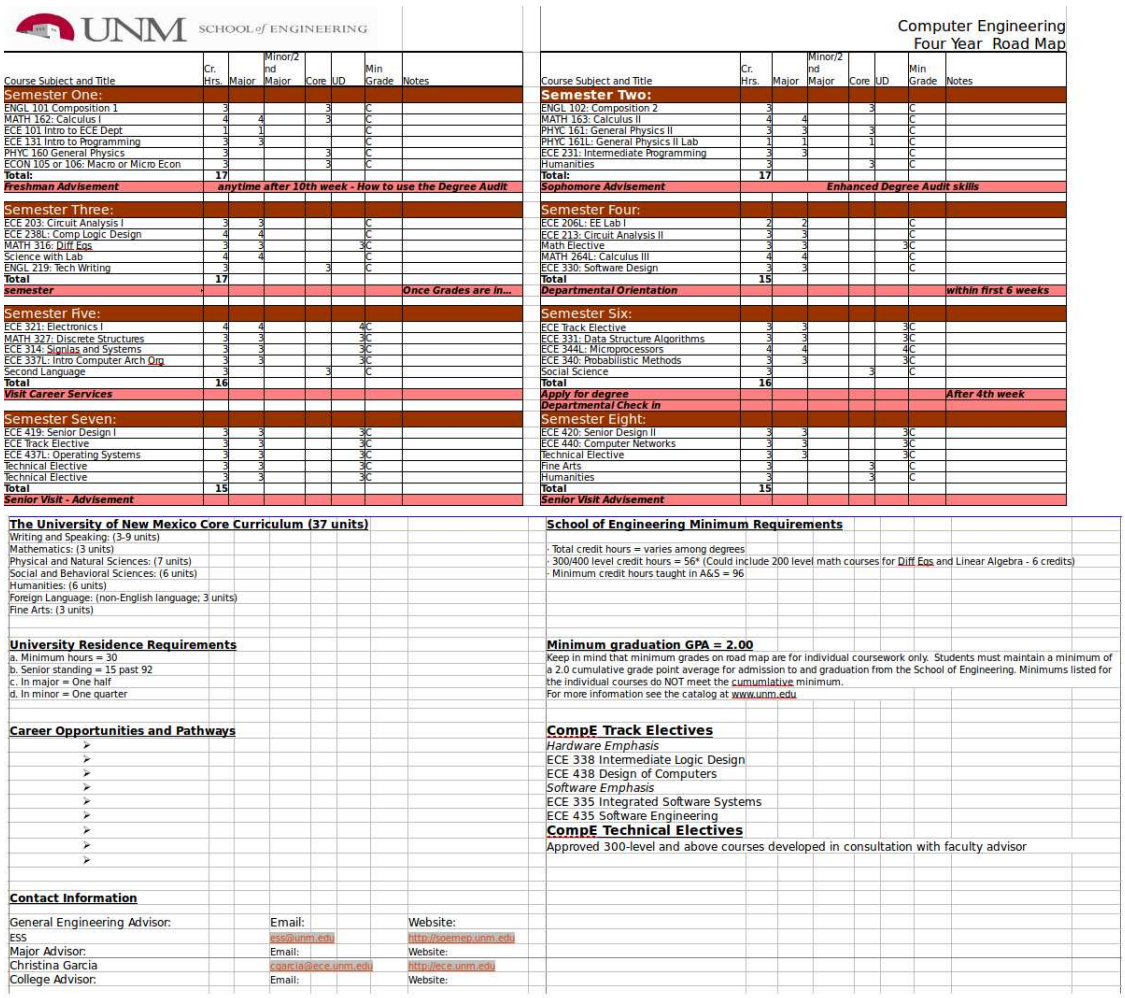
Chapter 2

High Level Design

2.1 Motivation

One of the main reasons that the degree plans application was implemented was a lack of a standardized system that was available to perform such a function. Before the current system was deployed, it was generally the responsibilities of the department advisors to create a degree plan for their students. Degree plans that were created were generally saved as spreadsheets. Shown in Figure 2.1 is a sample of a spreadsheet that was used for advisement of a student pursuing a Computer Engineering B.S. in the Electrical and Computer Engineering Department.

The spreadsheet is divided into two parts. The first part of the spreadsheet is essentially a chart that describes the degree plan term by term. In this first part, the courses that a student should take each term are listed. For each course, the number of credit hours is indicated. It is also indicated if the course counts towards the university's core requirements, upper division, or lower division requirements. The second part of the spreadsheet provides information such as university core requirements, residency requirements, and career opportunities.



UNM SCHOOL of ENGINEERING

Computer Engineering
Four Year Road Map

Course Subject and Title	Cr. Hrs.	Major	Minor/2nd Major	Core	UD	Min Grade	Notes
Semester One:							
ENGL 101 Composition 1	3			3		C	
MATH 162: Calculus I	4	4		3		C	
ECE 101 Intro to ECE Dept	1	1				C	
ECE 131 Intro to Programming	3	3				C	
PHYC 160 General Physics	3			3		C	
ECON 105 or 106: Macro or Micro Econ	3			3		C	
Total:	**17**						
Freshman Advisement	***anytime after 10th week - How to use the Degree Audit***						
Semester Three:							
ECE 203: Circuit Analysis I	3	3				C	
ECE 238L: Comp Logic Design	4	4				C	
MATH 316: Diff Eqs	3	3				3C	
Science with Lab	4	4				C	
ENGL 219: Tech Writing	3			3		C	
Total	**17**						
semester							***Once Grades are in...***
Semester Five:							
ECE 321: Electronics I	4	4				4C	
MATH 327: Discrete Structures	3	3				3C	
ECE 314: Signlas and Systems	3	3				3C	
ECE 337L: Intro Computer Arch Org	3	3				3C	
Second Language	3			3		C	
Total	**16**						
Visit Career Services							
Semester Seven:							
ECE 419: Senior Design I	3	3				3C	
ECE Track Elective	3	3				3C	
ECE 437L: Operating Systems	3	3				3C	
Technical Elective	3	3				3C	
Technical Elective	3	3				3C	
Total	**15**						
Senior Visit - Advisement							

Course Subject and Title	Cr. Hrs.	Major	Minor/2nd Major	Core	UD	Min Grade	Notes
Semester Two:							
ENGL 102: Composition 2	3			3		C	
MATH 163: Calculus II	4	4				C	
PHYC 161: General Physics II	3	3		3		C	
PHYC 161L: General Physics II Lab	1	1		1		C	
ECE 231: Intermediate Programming	3	3				C	
Humanities	3			3		C	
Total:	**17**						
Sophomore Advisement	***Enhanced Degree Audit skills***						
Semester Four:							
ECE 206L: EE Lab I	2	2				C	
ECE 213: Circuit Analysis II	3	3				C	
Math Elective	3	3				3C	
MATH 264L: Calculus III	4	4				C	
ECE 330: Software Design	3	3				C	
Total	**15**						
Departmental Orientation							***within first 6 weeks***
Semester Six:							
ECE Track Elective	3	3				3C	
ECE 331: Data Structure Algorithms	3	3				3C	
ECE 344L: Microprocessors	4	4				4C	
ECE 340: Probabilistic Methods	3	3				3C	
Social Science	3			3		C	
Total	**16**						
Apply for degree							***After 4th week***
Departmental Check in							
Semester Eight:							
ECE 420: Senior Design II	3	3				3C	
ECE 440: Computer Networks	3	3				3C	
Technical Elective	3	3				3C	
Fine Arts	3			3		C	
Humanities	3			3		C	
Total	**15**						
Senior Visit Advisement							

The University of New Mexico Core Curriculum (37 units)

Writing and Speaking: (3-9 units)
Mathematics: (3 units)
Physical and Natural Sciences: (7 units)
Social and Behavioral Sciences: (6 units)
Humanities: (6 units)
Foreign Language: (non-English language; 3 units)
Fine Arts: (3 units)

University Residence Requirements

a. Minimum hours = 30
b. Senior standing = 15 past 92
c. In major = One half
d. In minor = One quarter

Career Opportunities and Pathways

-
-
-
-
-
-
-
-
-

Contact Information

General Engineering Advisor:	Email:	Website:
ESS	ess@unm.edu	http://coemep.unm.edu
Major Advisor:	Email:	Website:
Christina Garcia	cgarcia@ece.unm.edu	http://ece.unm.edu
College Advisor:	Email:	Website:

School of Engineering Minimum Requirements

- Total credit hours = varies among degrees
- 300/400 level credit hours = 56* (Could include 200 level math courses for Diff Eqs and Linear Algebra - 6 credits)
- Minimum credit hours taught in A&S = 96

Minimum graduation GPA = 2.00

Keep in mind that minimum grades on road map are for individual coursework only. Students must maintain a minimum of a 2.0 cumulative grade point average for admission to and graduation from the School of Engineering. Minimums listed for the individual courses do NOT meet the cumumlative minimum.
For more infomation see the catalog at www.unm.edu

CompE Track Electives

Hardware Emphasis
ECE 338 Intermediate Logic Design
ECE 438 Design of Computers
Software Emphasis
ECE 335 Integrated Software Systems
ECE 435 Software Engineering

CompE Technical Electives

Approved 300-level and above courses developed in consultation with faculty advisor

Figure 2.1: Sample Degree Plan Spreadsheet

Although all of the degree plans are developed using a single template, there are still instances of data inconsistencies depending on how an individual within a department entered their data. The way that a course is represented could vary from plan to plan. The degree plan describing the Computer Engineering degree lists the number and title for each course. In contrast, the degree plan for Signed Language Interpreting lists only the number of the course.

There is also the issue of out of date information being entered into the degree plans. In the Computer Engineering plan, the student is instructed to take ECE

437L: Operating Systems in the seventh semester. However, this course no longer exists in the university course catalog, and instead is listed as ECE 437: Operating Systems; a slight difference, but one that could cause confusion. There are also instances when courses are represented in a degree plan with incomplete information. In the degree plan for Interdisciplinary University Studies, there is a course in the second semester listed as ENGL Composition 2. This incompleteness can once again be a source of confusion. With this course present in the first semester, it has the potential to bewilder first year students.

However, providing a central location for students to access degree information was not the only benefit. The Education Trust Higher Education Practice Guide describes five features common among high performing educational institutions [21]. One of the common areas is creating clear student pathways to success. Creating a centralized access point for degree plans offered at the university allows current and potential students to view what courses are necessary to take when.

The system as it currently exists and planned future implementations are designed in order to help satisfy many of the other features specified by Education Trust such as collecting data on student success and acting upon it, and not hesitating to "demand and require" of their student body. Every student at the university will be able to select one of the degree plans available within the system, or with the assistance of an advisor, make a custom degree plan. Advisors will then have a tool to ensure that a student stays on track in their given degree program and to advise their students accordingly.

2.2 Advantages

An online institution-wide degree plan application holds many advantages over the system of paper-based degree plans. These improvements benefit both the student

as well as the university.

Current Information. One of the areas where an online degree plan surpasses a paper plan is in the ability to always provide the user with the information that they need. As new requirements or courses are added, the degree plan that is available online is automatically updated with the latest information. This ensures that a potential student will always have access to the most current data available on a given degree plan within a department. Alternatively, if a student has been attending the university and has been admitted to a program against an older degree plan, that degree plan would still be readily available for that student to track progress.

Accessibility. With the application deployed "in the cloud," the degree plans can be accessed from any device that has a connection to the Internet. This allows potential students access to degree plans no matter their location. In addition to access, the degree plans can also be easily translated into different languages. This is a particularly useful feature for international students. Plug-ins are also widely available for web browsers such as Mozilla Firefox and Google Chrome, allowing the end user to select their language of choice.

Standardization. The web application allows the degree plans to be displayed to the student in a standardized manner. All of the data that comprises a specific degree plan is stored using a structured database, currently a relational database, but moving forward several NoSQL options are being considered. This essentially removes the way that a degree plan looks from the content that it contains. This, along with the wide accessibility of degree plans through the application, makes it easy to compare degree plans in a meaningful way, even if the degree plans exist in different departments. Performing these comparisons using paper-based degree plans would be significantly more difficult, particularly between departments.

Verification. Because all of the degree plans offered by the university are available in one central location, this makes it very straightforward for the degree plans to be reviewed for correctness, both automatically and manually. All degree plans at the University of New Mexico currently contain at least 120 credit hours, and no more than 18 credits can be taken in any semester without advisor approval. When a degree plan is constructed, it would be very easy for an automated system to perform these checks. An automated system that ensures that a degree plan satisfies the university core requirements, college or department core requirements, and any other requirements could also be constructed.

Analytics. One of the elements that has driven the development of this system is the ability to seamlessly integrate analytics. Introducing analytics into a cloud-deployed system is far easier than trying to produce the same results using a paper-based spreadsheet system. One basic piece of student-level analytics that is currently available in the deployed application is the ability to compare two degree plans, even across departments. Although it is feasible to perform this type of comparison using spreadsheets, it is far easier to accomplish a side-by-side comparison when both of the degree plans are hosted on the same site. Performing this comparison using spreadsheets would require a good deal of student time and "leg-work."

Using the deployed system, students will be able to compare how "on-track" they are within a given degree plan. While this would require integrating the degree plan system with existing student data, most likely data stored within the Student Data Mart maintained by the UNM Office of Institutional Analytics, this would be beneficial for both students and the university as a whole. If the number of students who are on track in a degree plan are known, the university would be able to more accurately make predictions on the number of courses that need to be offered on a semester-by-semester basis.

2.3 Curriculum Description

In order to gain a better understanding of the problem domain, the relationship of the contributing actors was analyzed. What at first seemed to be a straightforward problem domain quickly turned out to be more complex than anticipated. Many of the problems associated with issues such as representing prerequisites are present in the current implementation of the degree plan website, while many of these issues will be addressed in the subsequent versions.

2.3.1 Organizational Description

In order to create a degree plan, the manner in which the different entities at the University of New Mexico interact with one another needed to be analyzed. While the application being designed was focused mainly on degree plans, it was important to understand the other actors that exist in this space. In order to simplify this process, a Unified Modeling Language (UML) relationship diagram was created. The diagram can be viewed in Figure 2.2. This diagram is meant to provide a high level view as to how different entities relate to one another within the university as well as with other entities within the state of New Mexico.

This modeling is useful in capturing the important objects that exist within the domain of interest along with their relationships to one another. Such modeling is crucial in order to guarantee that the proper attributes are built into any system that is developed. It is also helpful in providing a framework for developers to better understand, build, and maintain the system.

The diagram in Figure 2.2 offers a generic high level view for any state institution of higher education in New Mexico. A generalized model of all of the institutions state wide was desired so that transfer agreements and articulations between state

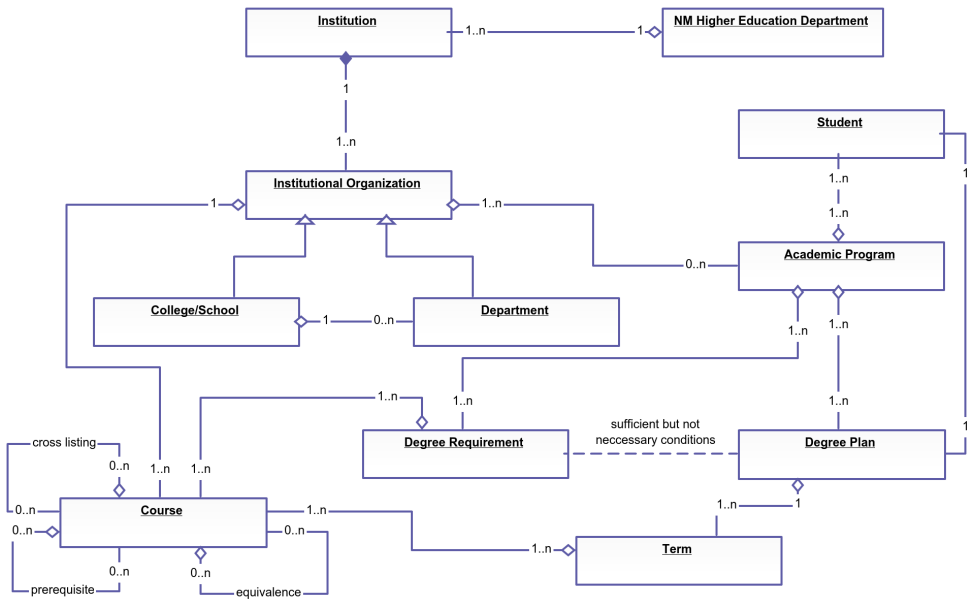


Figure 2.2: Description of Problem Domain

universities could be created. The model tries to be as descriptive as possible, while still retaining its generality. For example, the *organization* object within the model could stand for many different departmental divisions depending upon the university. The University of New Mexico organizes its divisions into schools or colleges, and then into departments. Central New Mexico Community College on the other hand, only uses schools as subdivisions of the institution. While this model may serve to describe any school in the state, any specific examples used in this thesis will generally involve the University of New Mexico.

What is apparent from the diagram is that all of the universities within the state are aggregated and overseen by the New Mexico Higher Education Department. This is very important in transferring coursework between universities, as the Higher Education Department defines the state general education core requirements and guarantees transfer of the courses between public institutions [5].

Outside of the State Higher Education Department is an *institution*, which is

composed of its lower level *institutional organizations*. An institutional organization can exist as any entity, such as a *college* or *department*. These objects are able to aggregate one another; this design was chosen in order to properly model the institutional hierarchy of a university. This way, organizations such as colleges or schools are able to accumulate and administer departments. Any organization can also have academic programs associated with it. This was an important decision because this allowed academic programs to be attached to any level of the institutional hierarchy.

A key object concerning the design of this system is the *academic program*. Within the UML diagram, an academic program class models any degree offered by a specific organization, be it a college, school, department, etc. An academic program is described fully by its set of degree requirements. A single *degree requirement*, which is described in more detail in the following section, is at its foundation a course or a collection of courses that a student must complete.

What a *degree plan* aims to do is to provide an ordered list of the courses a student must complete in order to be awarded a degree. An important constraint is that a degree plan must satisfy all of the degree requirements associated with a given academic program. Thus, if a student follows a given degree plan, they will satisfy the degree requirements and earn a degree. As shown in the diagram, a degree plan is split into terms. A *term* is the collection of all of the courses that a student must take, or at least possibly could take, within a given semester. Each academic program has one generalized degree plan, which students will be able to customize in order to create personalized degree plans.

Note in the UML diagram shown in Figure 2.2 the constraint that exists between the degree plan and the degree requirement. The courses outlined in the degree plan are sufficient to satisfy the degree requirements associated with the academic program that it is describing. However, there may exist other courses in the degree plan that are not enumerated degree requirements. This was important to build into

the system in order to build plans that contain prerequisites for required courses.

A good example of this situation exists in the Electrical & Computer Engineering Department. The lowest mathematics degree requirement necessary to obtain a bachelors degree in either electrical or computer engineering is Calculus I. However, it was discovered that a large percentage of students were entering the program at levels below this. In order to solve this problem, degree plans were created that included lower math courses to satisfy the prerequisites of Calculus I. While these lower courses were not necessary to be awarded an engineering degree, they were a necessary part of the degree plan for any student who did not meet the requirements for Calculus I.

A *student* also plays a key role in this system. A student can be anyone who is studying to receive a degree from a university. A student is able to belong to any number of academic programs. This is especially important at The University of New Mexico, where the majority of students are required to obtain a minor as well as a major. While the degree plan for a student should satisfy all requirements necessary to satisfy any major, minor or second major requirements, only one specific degree plan that allows the student to obtain his or her specific degree should be assigned to that student.

One of the more pivotal and complicated objects in this diagram is a *course*. Taking courses is the main way that a student interacts with a university, and courses are also what generally satisfy degree requirements. In Figure 2.2, it is shown that courses are also connected to one other in various ways. While a course can be offered by one department, it can have a *cross listing* in another department. For example, AFST 109, Introduction to Comparative Global and Ethnic Societies, is also offered as WMST 109, CCS 109, NATV 109, and SUST 109. This is an important relationship to capture; any cross-listed course can satisfy a requirement where any of the others are listed. Knowing which courses are cross-listed could also prevent a

student from taking two courses twice and accumulating excessive credits.

Another relationship that courses have among themselves is *equivalence*. This mainly occurs when one course is replaced by another equivalent course. One of the courses will generally cease to exist, but some students will still have the old course listed in their transcript. An example of this situation at the University of New Mexico occurred in the Fall 2014 semester. In this semester, the English department replaced ENGL 102 with ENGL 120. It is important that these two courses are recognized as being equivalent. If a student was working towards a degree consulted an earlier degree plan that listed ENGL 102, it is important that when the student takes ENGL 120, the requirement for ENGL 102 be recognized as being completed.

The last relationship, and probably the most common, that can occur between courses is the *prerequisite* relationship. This relationship defines which courses must be taken prior to enrolling is subsequent courses. Other variations of this relationship could be a *co-requisite*, a course that is required to be taken in the same semester as another course; or a *pre-or-co-requisite*, which can be taken in the same semester or at anytime before another course. These two relationships are left off of the UML diagram in Figure 2.2 for simplicity. It is important that these two relationships be represented so that when degree plans are created, the courses can be ordered properly among the semesters.

Not currently listed in this diagram is the notion of placement of a student. When a student is admitted into a university, the student will have doubtlessly taken some standardized admission test – be it the SAT, ACT, or some sort of institutional placement exam. These tests will allow the university to specify which courses a student will be able to start taking immediately regardless of prerequisites satisfied.

2.3.2 Requirements Description

Areas of the system that were considered to be more complex and requiring deeper analysis were the degree requirements and devising a way to represent how a student should satisfy them in order to be awarded a degree. One method to simplify this matter demands requirements be split into separate subtypes, and then aggregated to form a degree for a student. Figure 2.3 is a detailed UML diagram representing this system.

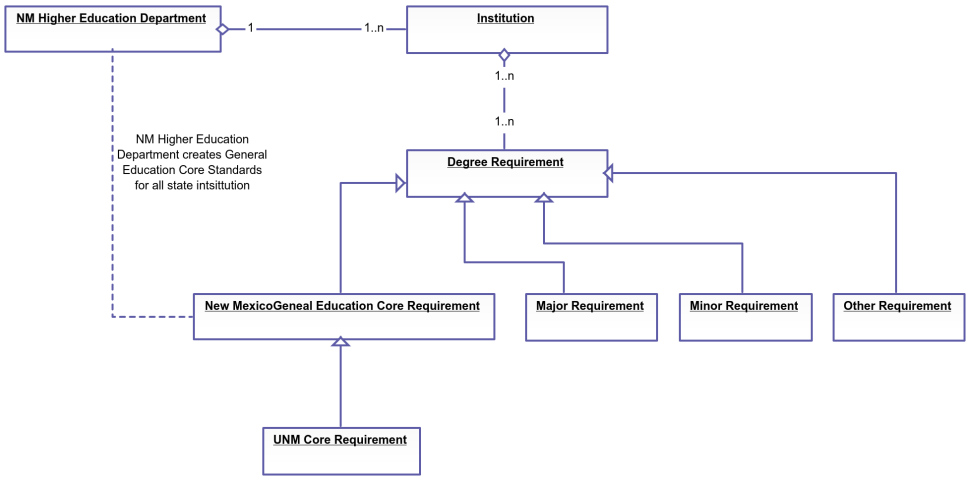


Figure 2.3: Requirements Breakdown

This diagram is basically a subset of the objects described in Figure 2.2. This diagram describes the different types of requirements that are available within the system as well as the way that the requirements are related between universities.

Requirements defined by a university within the state of New Mexico can be modeled as *major*, *minor*, *New Mexico general education core*, or simply *other*, and then aggregated accordingly. Predictably, a major requirement is one that fulfills courses for a student's declared major, while a minor requirement fulfills courses for a student's declared minor. The *other* requirement can be used to describe any number of departmental stipulations that are necessary to be awarded a degree. An

example of this situation would be the group requirements that are required by the Department of Arts and Sciences at the University of New Mexico.

One special set of requirements that needed to be satisfied were the New Mexico general education core requirements. These requirements, as shown in the diagram in Figure 2.3 are set by the state Higher Education Department. There are six areas that are required to be satisfied to be awarded a degree in a New Mexico public university, and these courses are guaranteed to transfer to any institution within the state. In addition, universities can themselves add core requirements specific to their own university. The University of New Mexico for example, adds a seventh core area, foreign language, that all of its students must complete in order to be awarded a degree.

It was necessary to split the requirements that are needed to satisfy a degree into separate requirements because of the flexibility needed to complete a degree plan in most departments. According to the UNM catalog, in Arts & Sciences alone there are 44 options from which to select a major, and 54 from which to select a minor. This implies that there are over 2000 ways in which a student could make a major/minor selection just within the Arts & Sciences Department.

Splitting the requirements into smaller sets allows requirements from different collections (e.g. major, minor, core) to be combined, and a resultant degree plan be generated for the student.

Chapter 3

Current Degree Plan Website

3.1 Web Applications Framework

The current degree plan website for the University of New Mexico can be found at `degrees.unm.edu`. The website was first deployed in July of 2013, and is currently the web site of record for degree plans at the University of New Mexico. Developed using a web applications framework, the site provides current and potential students as well as advisors and faculty one common repository for all of the degrees offered by the university

The website was built using a web applications framework that provided a set of core functionality such as an HTML templating system, data persistence model, and user session management [19]. The specific web applications framework that was used was Ruby on Rails (Rails), which is built on top of the Ruby programming language. Rails is an all inclusive framework designed to build database-backed web applications that conform to the model-view-controller (MVC) design pattern.

MVC splits the application into three separate layers, each with its own respon-

sibility. The *model* layer represents the application data model, and also connect to the back-end database. The *controller* layer receives incoming HTTP requests, and handles them accordingly. The *view* layer consists of Ruby-embedded HTML templates that provide the user an interface to the application [16]. Since the MVC framework was provided "out of the box" by Rails, this allowed developer time to be spent mainly on developing actual code for the application instead of devoting time solely to configuration. This allowed the development team to quickly develop and deploy the application.

Another component of the Rails framework that allows for agile development is ActiveRecord, the Object Relational Mapper (ORM) supplied by Rails. ActiveRecord maps database tables to classes, rows of database tables to objects of the associated class, and database columns to class attributes. A set of class-level methods are also supplied by the classes that correspond to table operations. ActiveRecord helps abate the amount of time a developer would have spent configuring the database, as well as dealing with the object-relational mismatch, leaving more time to develop business logic [15].

3.2 Deployment

Heroku was used for production-level deployment of the application. Heroku is a Platform as a Service (PaaS) cloud computing service managed by SalesForce that provides support for Rails. Heroku allowed the University of New Mexico degree plan application to be easily deployed and managed. Applications hosted on Heroku are deployed using Git, a distributed version control system. This was a good fit for the application since Git was already in use as the application source code management system. The application was also easily scaled up or down depending on the number of users of the site. Heroku provides enterprise-grade PostgreSQL-as-a-service

for deployed applications [8]. This influenced the choice of the relational database management system (RDBMS) for the application.

3.3 Relational Database Management System

3.4 Data Model

Although Rails does alleviate the complexity of configuring and interfacing with a database, some consideration still has to be focused on the overall database design. A useful tool to assist in this process is an Entity Relationship Diagram (ERD). Figure 3.1 shows the ERD for the degree plans application. The ERD is a very convenient way to visualize the structure of the database, and aids in its design and continual refinement.

This ERD serves a different purpose than that of the UML diagram. The UML diagram is used to model the domain of the problem, while the ERD is used to model the data from the perspective of linked tables of data. The ERD shows values that are available in each of the rows of the database. The primary keys are indicated by a yellow bullet, and are all integers. Each row of a table has a unique primary key. Foreign keys are indicated by a red bullet, which are used to track relationships among the database entries. All other attributes on the tables are either integers, represented by the field *INT*; strings, shown in the diagram as *varchar(255)*; *boolean*; or large *text* blocks.

For a simple one-to-many relationship, it acceptable for the aggregated object to simply contain the primary key of the other object that is aggregating it. For example, a college has many departments, while a department can only belong to one college. In order to model a many-to-many cardinality, join tables which contain

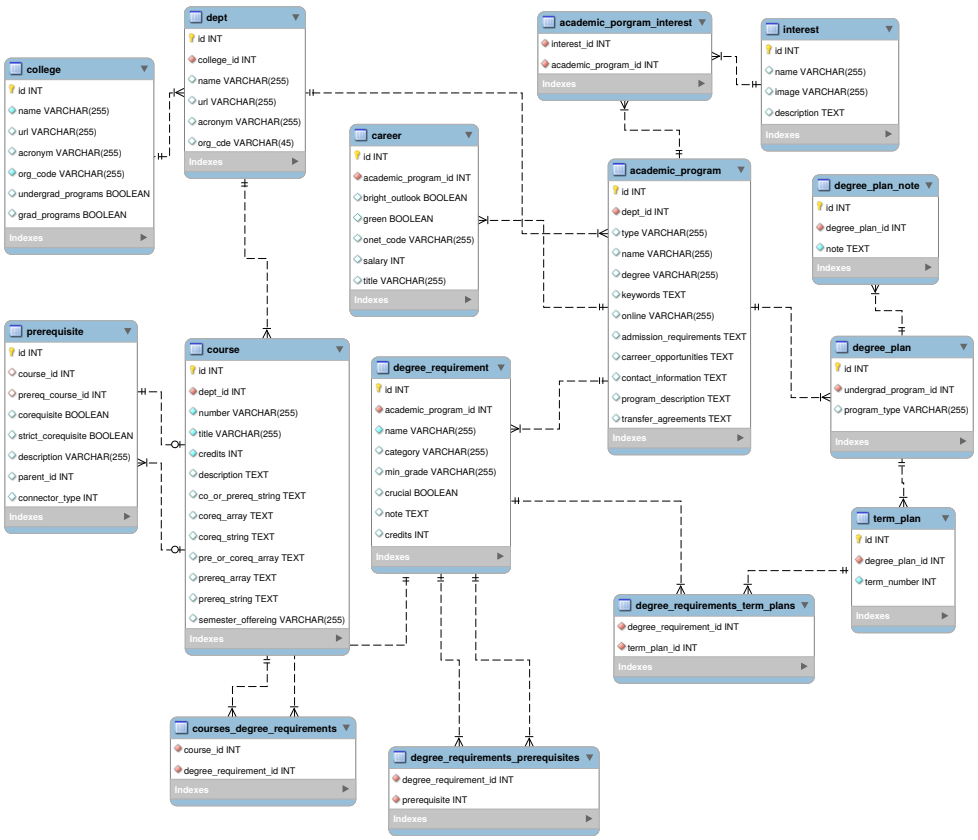


Figure 3.1: UNM Degree Plan Entity Relationship Diagram

pairs of primary keys of associated objects are required. Each type of object is then able to belong to more than one of the associated type of object and vice versa. For example, an interest area can cover many different academic programs, but an academic program in one interest are may also exist in several others.

The data model described by this ERD also employs a methodology known as Single Table Inheritance (STI) in order to differentiate amongst the different types of academic programs, e.g., undergraduate, graduate, and certificate [1]. The different academic programs are differentiated by a *type* column in the *academic_program* table that stores the name of the inheriting class. This technique was employed to capture the different actions associated with different types of academic programs. The primary difference between the academic programs is that only undergraduate

programs are able to have degree plans associated with them. While other academic programs such as graduate programs do have requirements, most are very flexible and are not easily coerced into a degree plan.

The database structure defined in this ERD is very similar to the object relations defined in the UML diagram in Figure 2.2 with some exceptions. The concept of a *student* is not yet present in this data model. This is due to the fact that this version of the application is meant to only to provide generalized degree plans for each department, not provide customizable degree plans at the level of a student. Also, entities for *interest* and *career* exist in the data model but not the UML diagram. An interest was added in order to allow application users to explore degree plans by areas of interest. A career entity was added in order to allow application users to view possible career options after being awarded a specific degree. The *degree plan note* was also added in order to allow a degree plan to be described with human readable text.

3.5 Application Interface

Significant effort went into the application interface that students or potential students use in order to access the degree plans website. An interface was designed to allow the user many different avenues in order to access degree plans. Considerations were made as to how different users would navigate the website, including students who were in search of a particular degree plan, those who wanted to search by a college, or those who wanted to search by an area of interest

The application landing page was designed to be as simple as possible, and was meant simply to direct users to the type of degree plan that they wanted, either undergraduate, graduate or professional, or associate and certificate programs. There is also an area where an individual can view the accreditation of the university. Figure

3.2 shows a screen capture of the landing page.

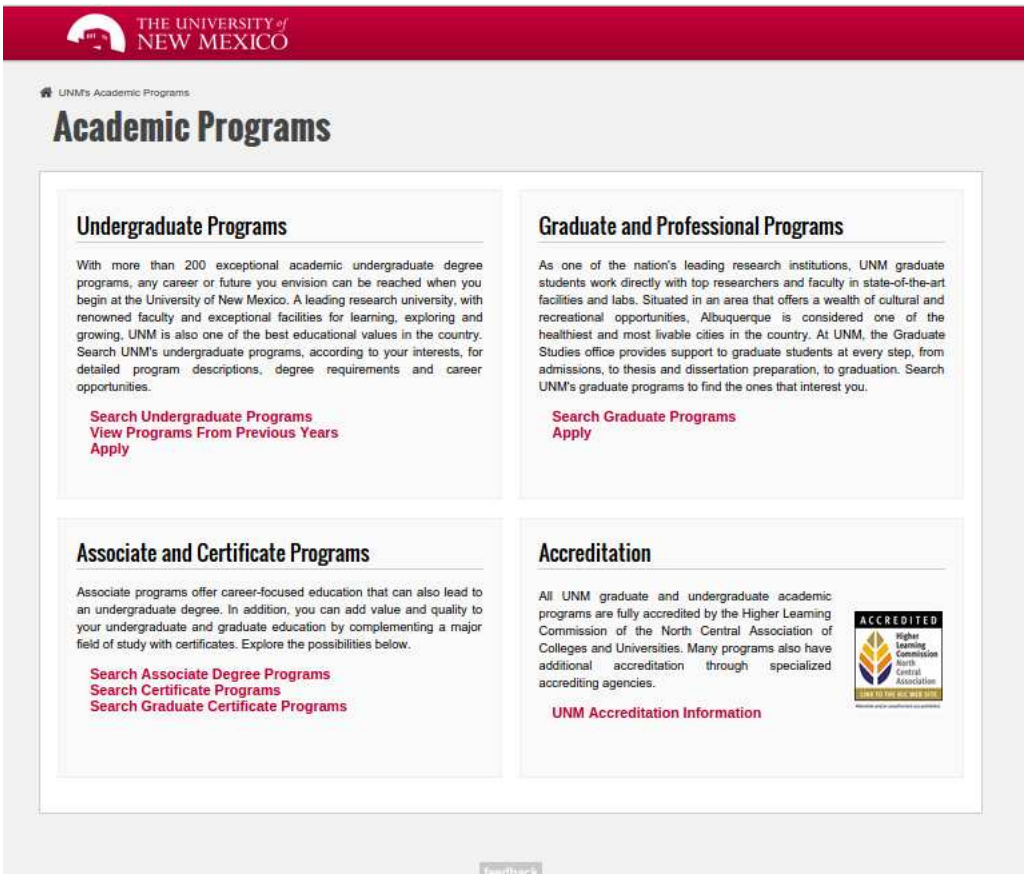


Figure 3.2: UNM Degree Plan Landing Page

Following any of the links labeled "Search", the user will be directed to an explore page, where the degree plans located on the website can be browsed in one of four ways: by college, interest, alphabetically, or keyword. No matter which link users follow, he or she will be presented with a simple list of the degree plans.

Once a degree plan is displayed, the user has the option of viewing a graph that shows that path through the selected degree plan. Figure 3.4 shows a screenshot of the graph for the Computer Engineering degree plan.

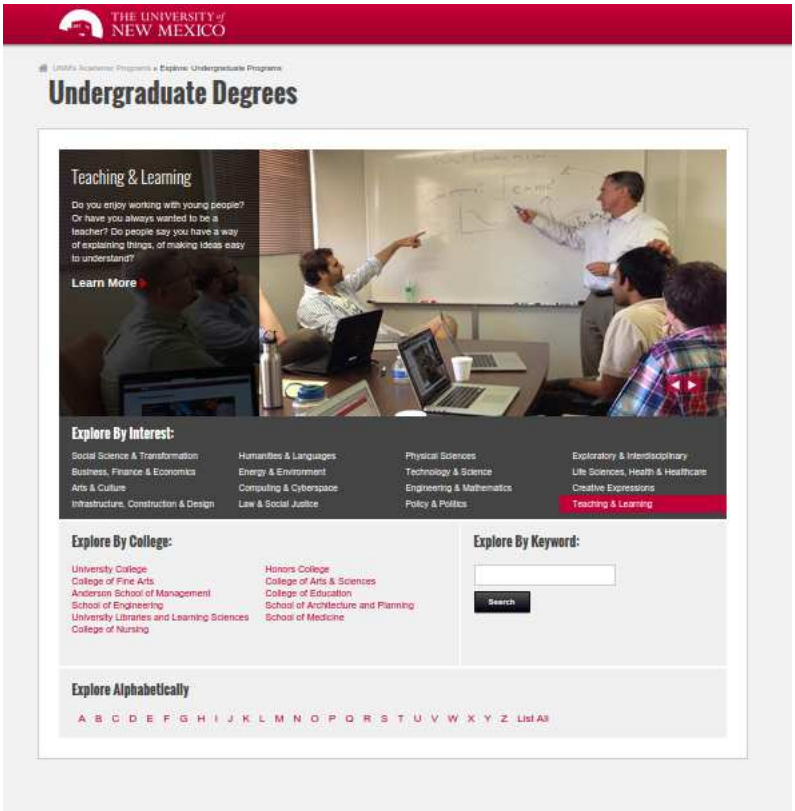


Figure 3.3: UNM Degree Plan Explore Page

3.5.1 Application Demand

The application has been equipped with Google Analytics, so that the number of users could be tracked, as well as how they flow through and interact with the website.

There has been much demand for the degree plans application. According to Google Analytics, the application has had 164,231 distinct Sessions initiated during the time in which the application was first launched on July 15, 2013. Out of the 164,231 sessions, 69.02% were new sessions. There were 113,377 distinct users, with 1,218,289 distinct page views. These users averaged a 4:35 session duration. A bounce rate of 20.84% was present. Figure 3.5 shows these results.

The most requested language was English, with over 95% of the sessions. Spanish,

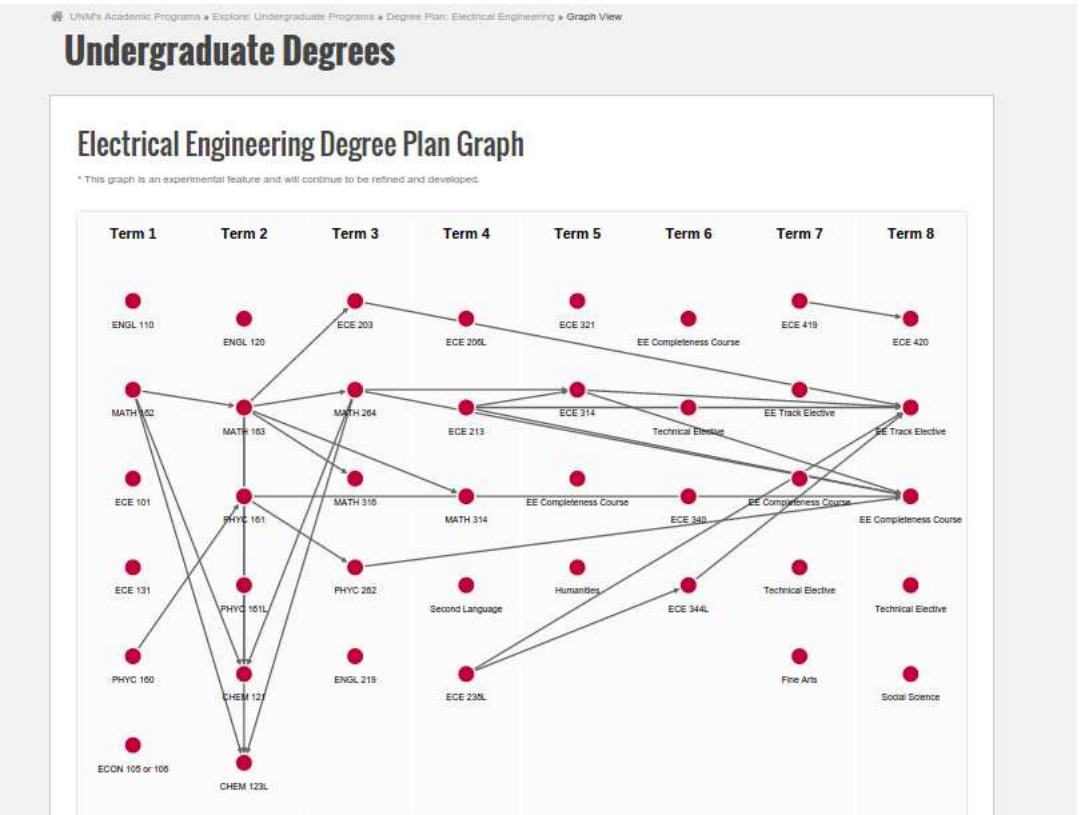


Figure 3.4: UNM Degree Plan Graph View

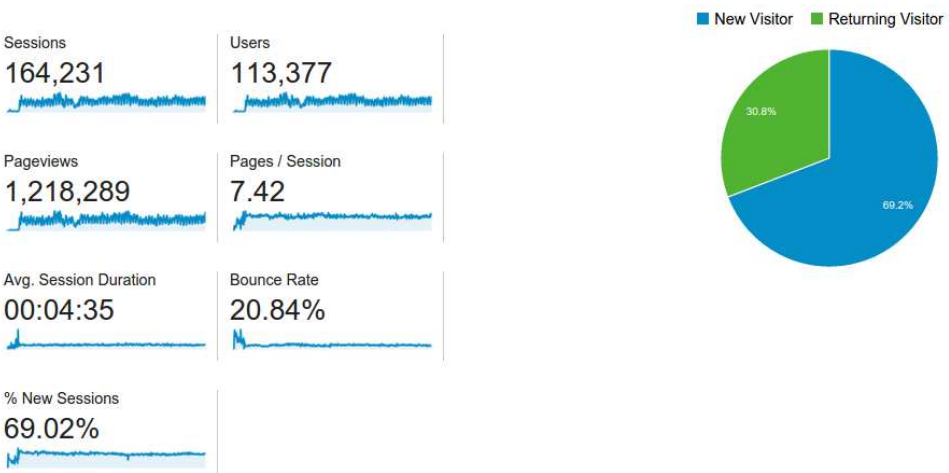


Figure 3.5: Overview of Site Analytics

Chinese, Arabic, Portuguese and French were also requested. This is shown in Figure 3.6

	Language	Sessions	% Sessions
1.	en-us	156,611	95.36%
2.	es	1,326	0.81%
3.	en	1,074	0.65%
4.	zh-cn	858	0.52%
5.	en-gb	813	0.50%
6.	es-es	605	0.37%
7.	pt-br	338	0.21%
8.	ar	286	0.17%
9.	es-419	222	0.14%
10.	fr	215	0.13%

Figure 3.6: Top Ten Languages Requested

The site has also had users from the following countries described in Figure 3.7, with countries with darker shades of blue initiating more session. The United States had the most with 150,548.

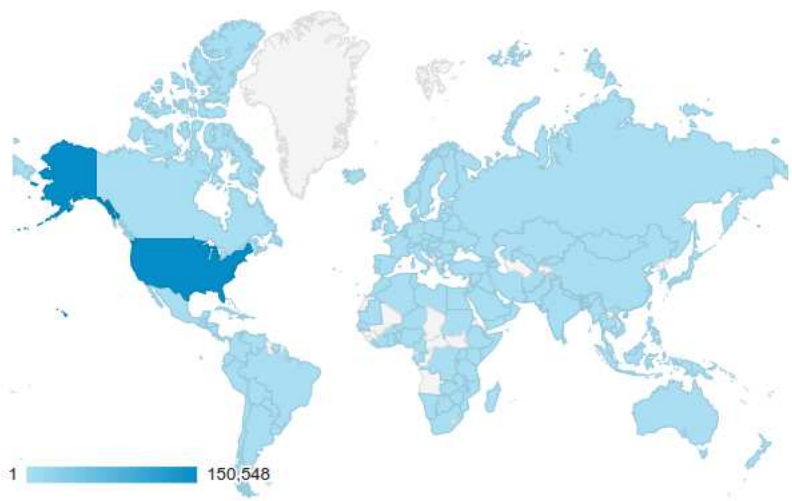


Figure 3.7: Degree Plans Geographic Locations

An interesting piece of information gleaned from the analytics is the cyclic request

nature of the site. The usage would spike in the beginning of the week, and then gradually drop off as the weekend approached. This behavior can be seen in the plot in Figure 3.8

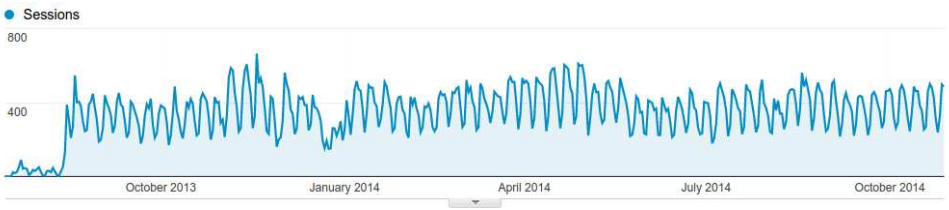


Figure 3.8: Degree Plans Site Usage

3.6 Limitations of Initial Implementation

Below, a description of some of the limitations associated with the implementation of the degree plans web service is described. In Chapter 5, a new design is described that was created to address many of these limitations.

3.6.1 Prerequisite Representation

One entity that was re-factored from the UML diagram and involved significant complexity is the *prerequisite* entity found in the data model in Figure 3.3. This entity was first devised in order to store the prerequisites for a given course. Each object of this type can either be a course, which would contain a foreign key that would point back to a course in the database; or a logical operator, determined by storing a 1 for "or" or a 2 for "and". The prerequisites for the courses were delivered as strings, so data transformations were required. First, the prerequisites were transformed into prefix notation, which allowed the data to be stored without any parentheses, represented by an array. A prefix equation could be easily represented by a binary

expression tree, which could be stored in the database by tracking the node's parent node through a *parent_id*.

Take for example the course MATH 162, with the prerequisite string *MATH 123 and MATH 150*. The prerequisite string converted into a prefix equation represented as an array in Ruby is shown in Figure 3.9.

```
1 ["*", "MATH 123", "MATH 150"]
```

Figure 3.9: Ruby Prefix Equation Array

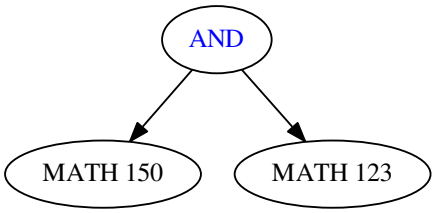


Figure 3.10: Prefix Equation Represented As A Tree

This prefix prerequisite array is then easily represented by a tree, as shown in Figure 3.10. This tree can be modeled in the database as shown by the abbreviated tables in Figure 3.11. Saving the prerequisites in prefix form allowed them to be more easily analyzed by a machine than if they were saved in infix form.

Prerequisites Table

id	course_id	parent_id	prereq_course_id	corequisite	strict_corequisite	connector_type
201	101	NULL	NULL	FALSE	FALSE	2
202	101	201	102	FALSE	FALSE	NULL
203	101	201	103	FALSE	FALSE	NULL

Courses Table

id	number	title
101	MATH 162	Calculus I
102	MATH 123	Trigonometry
103	MATH 150	Pre-Calculus Mathematics

Figure 3.11: Binary Expression Tree Inserted Into Database

The tree structure of Figure 3.10 can be found in the tables in Figure 3.11. The prerequisite with *id* of 201 corresponds to the AND root node of the tree. The prerequisite with *id* of 202 corresponds to the course MATH 123, while the prerequisite with *id* of 203 corresponds to the course MATH 150. The root node of the tree can be deduced from the database row entries as it will be the only entry with a *parent_id* of NULL. From here, the tree is able to be traversed.

Although this scheme could be used to accurately store the prerequisites for a given course, this method was eventually replaced by a simpler method. Once a course is loaded into the database it is static, and does not regularly change. This lessens the need to normalize the prerequisite table. In order to prevent performing unnecessary joins at runtime, the prerequisite array shown in Figure 3.9 was simply stored as is in the database. Rails natively provides mechanisms to save an array to the database. This is accomplished by serializing the array in the YAML format whenever the array is saved, and reassembling it as an array whenever it is retrieved from storage.

3.6.2 Additional Semester Years

When adding new degrees into the application for the 2014-2015 calendar year, some difficulties were encountered. Creating an archive of previous years' academic programs, as well identifying the current degree plans in the system became necessary. In order to resolve this problem, a *year* table was added to the database. Figure 3.12 shows the updated data model.

Adding one entity to the diagram greatly increases the complexity as compared to the diagram in Figure 3.3. The year table aggregates colleges, departments, academic programs, courses, and degree plans. This allows each of these entities to be be searched by year, and then either displayed in the archive or the section of the

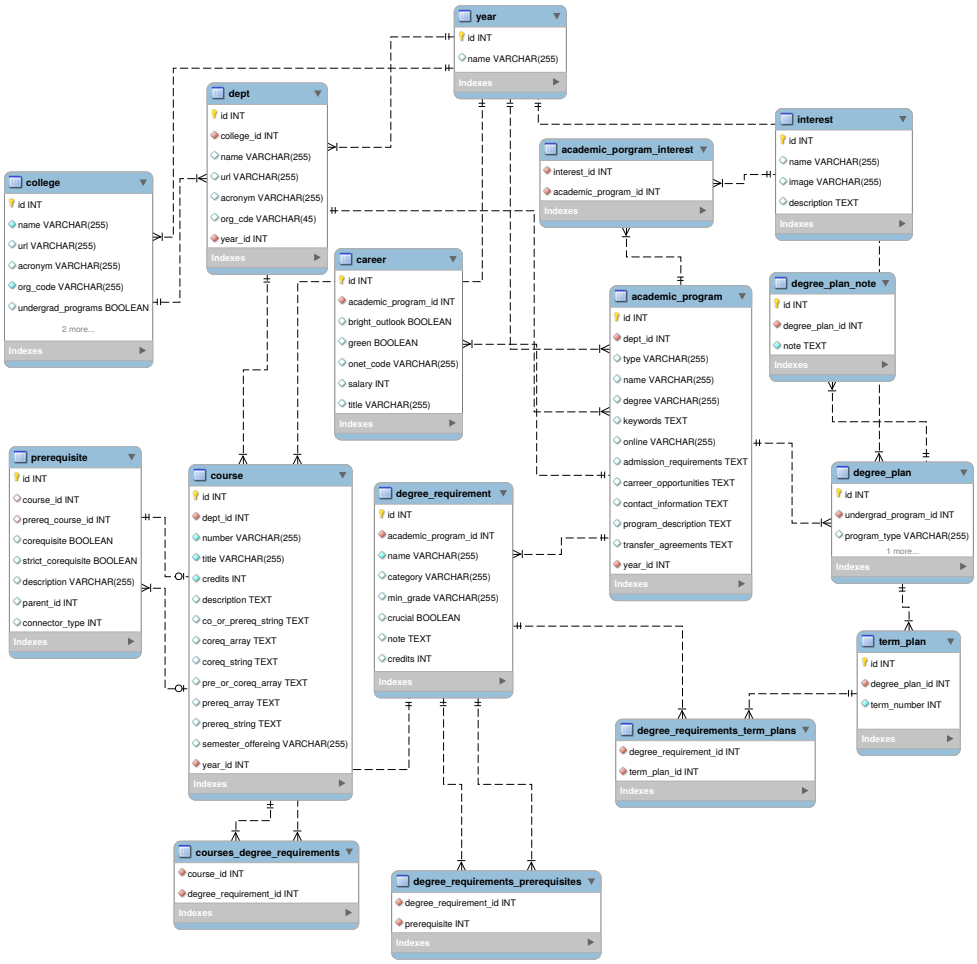


Figure 3.12: Entity Relationship Diagram With Year Entity Added

application that displays the most current degree plan.

While this worked as a solution, it also added some complexity to the database. There were issues of denormalization associated with all of the degree plans that were created. The year table essentially created two sets of all entities originally in the database. Thus, an entity such as a course could exist twice with exactly the same attributes, only tied to a different year. All queries start at the year table, and an entity should only be associated with other entities in the same year. This was to facilitate the site flow and enforce degree plan correctness.

Several join tables could be added to resolve this issue, but this is one of the issues that moving to a modern computational infrastructure can solve.

3.6.3 Data Ingestion

Uploading information to the degree plan website was somewhat complicated. All of the degree plans at the university were converted manually from spreadsheets to files that could be used to populate the database.

The University Information Technologies Office (IT) was able to provide information related to courses at the University of New Mexico as a large JavaScript Object Notation (JSON) file. There were issues concerning how the data was presented, in particular prerequisites. Several scripts needed to be written in order to convert the prerequisites into the format described above in Section 3.6.1. Figure 3.13 shows a representation of how the prerequisites were provided in the JSON file.

One of the issues associated with presenting the prerequisites in this manner is that logic is combined with the saved data. Each element from the prerequisite array can contain courses, parentheses, logical operators, or other prerequisite requirements such as placement exams. Once all of the elements are joined, what is formed is a string that describes the prerequisites. In this case the string would be:

> *(ACT Math minimum score 28 or SAT Mathematics minimum score 640 or MATH 150 or COMPASS - College Algebra minimum score 67) and (MATH 123 or COMPASS - Trigonometry minimum score 60) or (ACT Math minimum score 32 or SAT Mathematics minimum score 700)*

While this is presented in a human readable form, this string was not very useful for analyzing the prerequisites of a course. One of the key problems associated with

```
"prerequisites": [
  {
      "name": "( ACT Math  minimum score 28"
  },
  {
      "name": " or SAT Mathematics  minimum score 640"
  },
  {
      "name": " or MATH 150"
  },
  {
      "name": " or COMPASS - College Algebra  minimum score 67 )"
  },
  {
      "name": " and ( MATH 123"
  },
  {
      "name": " or COMPASS - Trigonometry  minimum score 60 )"
  },
  {
      "name": " or ( ACT Math  minimum score 32"
  },
  {
      "name": " or SAT Mathematics  minimum score 700 )"
  }
]
```

Figure 3.13: Example Prerequisite JSON obtained from IT

the prerequisites delivered by IT was the issue of mixing operators such as “or”, “and”, “(”, and “)” with the operands. Storing them as a prefix equation in an array allowed them to be represented without the use of parentheses. The prerequisites could then be easily parsed using a stack.

After many attempts to parse the actual prerequisite JSON array, it was found to be easier to simply use a regular expression to extract the prerequisites and co-requisites out of the course description found in the JSON file. Many errors were found such as missing operators and parentheses not being correctly closed.

The prerequisites had to undergo additional parsing in order for them to be saved in the database in a useful form, such as that shown in Figure 3.9. This conversion occurred once, and then the resulting array was stored in the database.

Chapter 4

New Technologies in Support of Modern Computational Infrastructure

In order to upgrade the system to a modern computational system that facilitates more sophisticated analytics, many new technologies had to be integrated into the application. The new technologies that are used to construct the new application dealt mainly with data storage and visualization.

4.1 NoSQL Technologies

The main area of improvement in the degree plans application is how data will be stored. The data in the application is currently stored in a relational database. Other persistent data storage models were considered in order to enhance data storage abilities.

NoSQL technology aims to solve many different problems associated with using relational databases in web applications. NoSQL databases represent a large swath of database technologies, and were developed in response to changing needs of the type of data being stored, how the data was accessed, as well as processing needs [12]. The advent of the NoSQL movement allowed developers to select the right database to manage their data, instead of being corralled into the relational "one size fits all" solution.

One of the major benefits that NoSQL databases bring to this project is the fact that they have dynamic schemata. In a relational database system, the schema of the table structure has to be well defined before any data is inserted. Using a NoSQL database, no assumptions about the data being stored need be made beforehand. This characteristic is extremely useful when dealing with datasets found in a university setting. Since not all aspects of the data will be known in advance, a dynamic schema provides an efficient way to deal with this issue.

NoSQL databases are generally described using the following four types. *Key-value stores* are the simplest NoSQL database. Every item in the database is stored as a key mapped to its value. *Document stores* are similar to a key-value store, but each value can be a complex data structure. *Graph based stores* persist data as node-edge pairs. *Column based stores* store common data as columns instead of rows, these stores are optimized for queries over large data sets [4].

The two types of NoSQL databases that were considered for design of the modern architecture were document and graph based-data stores. These two data stores most accurately represent the domain of the data represented by the university curriculum.

ACID versus BASE. Relational SQL and NoSQL databases have different guarantees on the integrity and availability of the data that they store. The two major transactional models used are either ACID or BASE. A play on words from the field

of chemistry, these two systems do offer some parallel assurances on how data will be handled.

As a rule, relational databases support an ACID transactional model. ACID is a set of properties that guarantee reliable database transactions. This concept was initially defined by Jim Gray, but Andreas Reuter and Theo Härder were the ones who coined the term [7].

ACID is an acronym that stands for **A**tomicity, **C**onsistency, **I**solation, **D**urability. Atomicity represents the fact that the transaction that takes place must be all or nothing, if one part fails, it should all fail. Consistency guarantees that the database will remain in a valid state, no matter the transaction. Isolation implies that events must be synchronized, and hidden from other transactions that are running concurrently. Durability implies that once a transaction has taken place, the results will persist in the data store. The concern of a database employing an ACID transactional scheme is one of consistency.

BASE on the other hand concerns itself less with consistency and more with availability. BASE stands for **B**asically **A**vailable, **S**oft state, **E**ventual consistency. Whereas ACID has a pessimistic view on database consistency, BASE takes an optimistic approach and assumes that the database consistency will always contain a certain degree of flux. Allowing for some flux with the consistency of the database allows for a the system to be more partition resistant [13].

A factor to consider in describing what database transactional model to consider is the CAP Theorem introduced by Eric Brewer. Brewer first described this theorem in the context of a web application distributed over geographically distant servers. The concept however is general enough to describe individual databases in a distributed, cloud-based data storage system [6].

CAP is an acronym for **C**onsistency, **A**vailability, **P**artition tolerance. Consis-

tency is defined as the ability for each database server to return the correct response to a given request. Availability is even more core than consistency, and this guarantees that a response will be issued for a request. Partition tolerance actually refers more to the underlying infrastructure than to the actual database servers themselves. What this measures is the ability of the system to be located on many end nodes. If the servers are not able to communicate with one another, or communication is too slow, operations may not be able to be carried out. In visualizing the CAP theorem, a triangle such as the one in Figure 4.1 is used.

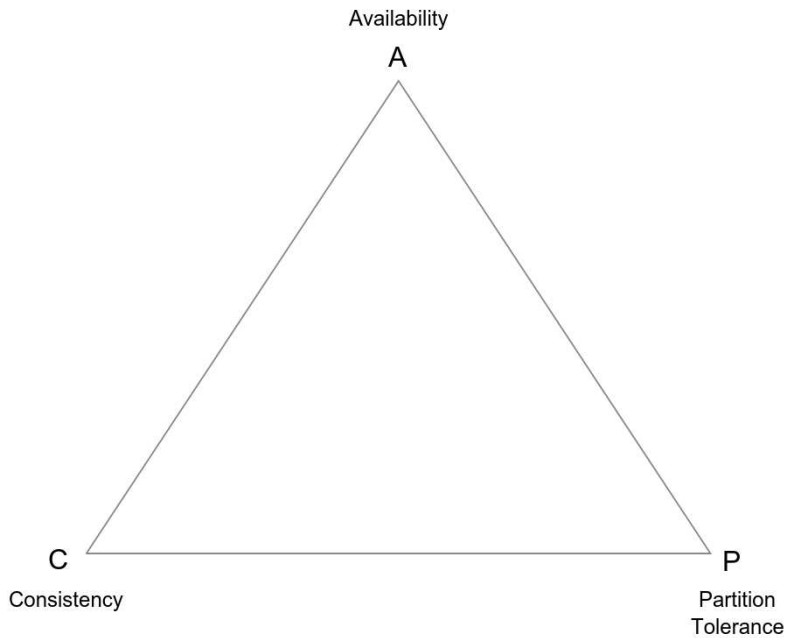


Figure 4.1: CAP Triangle

The CAP Theorem states that not all of the items (CAP) can be satisfied at one time, that there has to be a trade-off with one element. In the context of Figure 4.1, a system would exist between any two of the points of the triangle, and exhibit those two characteristics strongly. The third point is not necessarily non-existent in the system, but usually comes with a higher overhead or latency cost. For example, PostgreSQL is a RDBMS, so it follows the ACID transactional scheme. This places

it between the "C" and "A" points on the triangle, meaning that it is consistent as well as available. Partition tolerance can be achieved in a cloud-deployed relational database system using a two-phase commit system, where all partitioned databases have to agree that a transaction can take place before it is committed. There is obviously some overhead associated with coordinating communication among the databases.

4.1.1 Graph Database

The graph database engine selected to store the information associated with curriculum was Neo4j. One reason for selecting Neo4j was support of ACID transactions. Since this database would hold all of the courses and curriculum associated with the university, it was important that the there be some consistency in this data.

Another data management feature that was important to this selection was the flexible schema that Neo4j provides. No predefined schema is needed to create the database, and nodes can be related to other nodes by any relation type, which can be created dynamically.

Neo4j treats relations as first-level citizens, and thus are embraced as a core aspect of its data model. Relationships between nodes are readily available in a Neo4j, and no joins are necessary in order to expose those relationships. Since these relationships are stored directly in the database, accessing them becomes a constant time operation, and traversing the graph structure becomes a straightforward exercise [20].

Neo4j also has its own query language, cypher. Cypher queries are easily formed and are generally more human readable than SQL queries. Figure 4.2 contains a snippet of cypher code that will return the node representative of MATH 162 Calculus I.

```
1 START root=node:courses_index(number = "MATH 162")
2 RETURN root
```

Figure 4.2: Example Cypher Query

Another important advantage of using this database is that it comes equipped with a web server that allows a user to perform queries using the standard HTTP Protocol. There are also a number of libraries similar to an ORM for an RDBMS that allow easy access to the HTTP Neo4j interface. The Ruby library, or "gem", that was used to connect to the cloud-deployed Neo4j database was Neography [3].

There is also a built-in administrative dashboard that comes standard with Neo4j. This administrative interface gives application developers a console to run cypher commands and a means to explore the contents of the graph visually. Figure 4.3 shows a screen-shot of the Neo4j administrative dashboard.

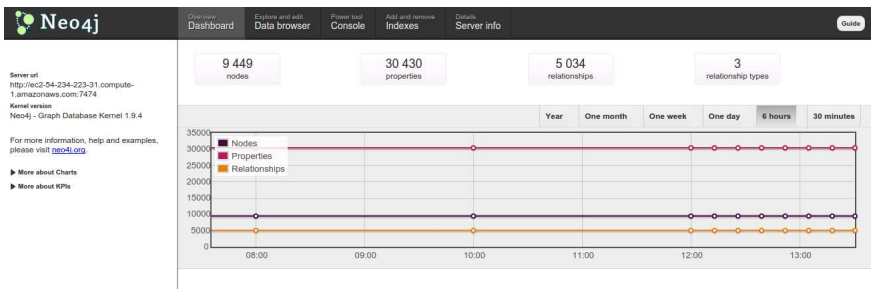


Figure 4.3: Neo4j Administrative Dashboard Screenshot

4.1.2 Document Database

The document database that was chosen for this system was MongoDB, which is an open source document database allowing for high performance and availability as well as automatic scaling [9].

While the graph database had been selected in order to store the data concerning the university structure and the degree requirements, MongoDB had been selected in

order to store the degree plans for each student. The data model that MongoDB uses to store data was a major influence in this decision. MongoDB stores information in collections of documents as opposed to a relational database's tables with rows. Instead, MongoDB design encourages the use of embedded sub-documents [10]. This approach precludes the need to store data in separate tables and use join procedures to reassemble it.

MongoDB generally follows a BASE consistency model, and centers around being consistent and partition resistant. In order to maintain availability, replica sets employ automatic failover in order to provide availability. Replica set members contain the same data, but are otherwise independent. This allows a secondary replica set member to be promoted in the event of a primary member becoming unavailable [11].

MongoDB stores documents in Binary JavaScript Object Notation (BSON) format. BSON is lightweight with low spatial overhead, traversable by structure, and efficiently encoded and decoded. BSON is a basically a schema-less protocol buffer, which makes it very flexible but introduces some overhead in regards to space efficiency [2].

Chapter 5

Modern Computational Infrastructure Design

5.1 Newly Designed Infrastructure

5.1.1 Graph Database

A graph is simply a set of *nodes* and a set of *vertices* that connect them. Graphs accurately model the manner in which many entities exist in the real world [14], especially in the domain of a university. This is apparent with such relations as those between a college and its departments, or a course and its prerequisites.

One of the main components of the system is a Neo4j graph database that contains information on curriculum at the University of New Mexico. This database houses information pertaining to schools and colleges, departments, academic programs, and degree requirements. The database also contains all of the courses offered at the University of New Mexico, and all of the associated prerequisites and co-requisites. A graph-based persistent storage model was chosen due to its closeness to the domain

of the data represented. Courses and their prerequisites, colleges and their associated departments, are all very easily modeled as graphs.

As well as being a close match to the domain of the data, the graph database also has the advantage of being a schema-less persistent storage engine. As mentioned above, without a schema, new attributes can be added to any node or edge without any prior configuration. Also, any node type can be connected to any other node type using any type of edge, which itself may have attributes.

University Organizations. The colleges, departments, and academic programs naturally fit into a hierarchical tree structure. Figure 5.1 shows a representation of the School of Engineering and one of its associated departments and its two degrees offered.

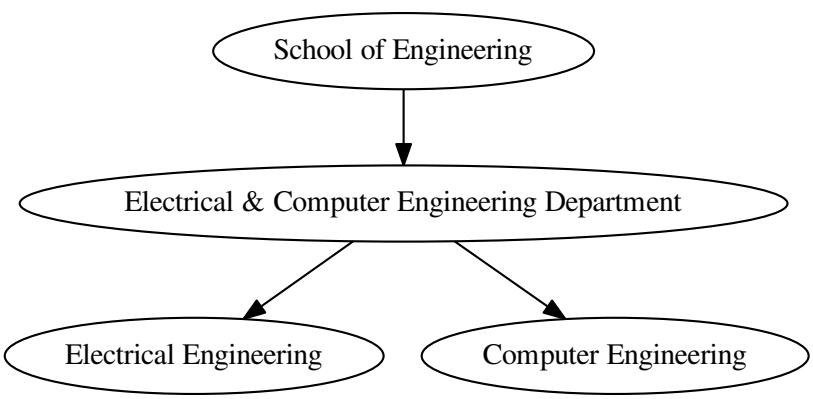


Figure 5.1: University Relationships

Given how the entities relate to one another, modeling them using a graph best fits their natural structure; storing them in Neo4j allows them to be persisted using their natural structure. These entities are not generally queried based on the data stored in the nodes, but rather how they relate to one another. For example, one query may

be to find all of the academic programs that are taught through a specific college. First, the college can be found by querying the *college* nodes using a predefined index, then all of the nodes that are related to the college node would be returned. In Figure 5.1, the query would start at *School of Engineering*, and the academic program nodes that would be returned would be *Electrical Engineering* and *Computer Engineering*.

Courses. The courses at the University of New Mexico also fit a graph data structure very well. The courses generally contain information that describes the course, such as course number, title, and description. There is also a notion of courses being able to relate to one another through prerequisite or co-requisite relationships. This is easily modeled using Neo4j by representing a course as a node with all of its defining attributes, and using edges to represent the relationships between the courses. Figure 5.2 demonstrates the relationships present between the courses MATH 162 Calculus I, MATH 163 Calculus II, PHYC 162 General Physics I, and PHYC 163L General Physics I Lab.

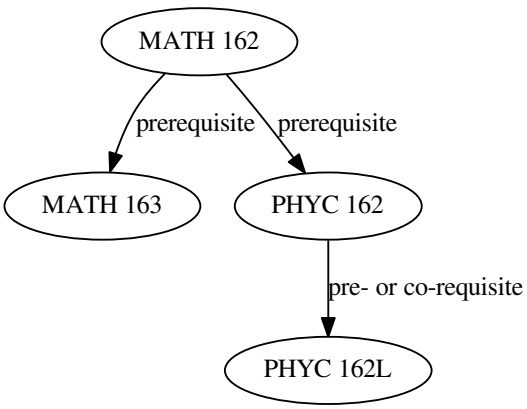


Figure 5.2: Course Relationships

Figure 5.2 shows two of the relationships that can exist between courses at the

University of New Mexico. MATH 162 is shown as a prerequisite for PHYC 162, while PHYC 162 must be taken in the same semester or before PHYC 162L. Neo4j's ability to place attributes not only on the nodes but also the relationships between them is leveraged here.

Degree Requirements. Degree requirements are also stored in the Neo4j database. Degree requirements are also stored as a tree in the graph database. In Figure 5.3, a small section of the tree stored in Neo4j that describes the degree requirements for the *Computer Engineering Degree* is shown.

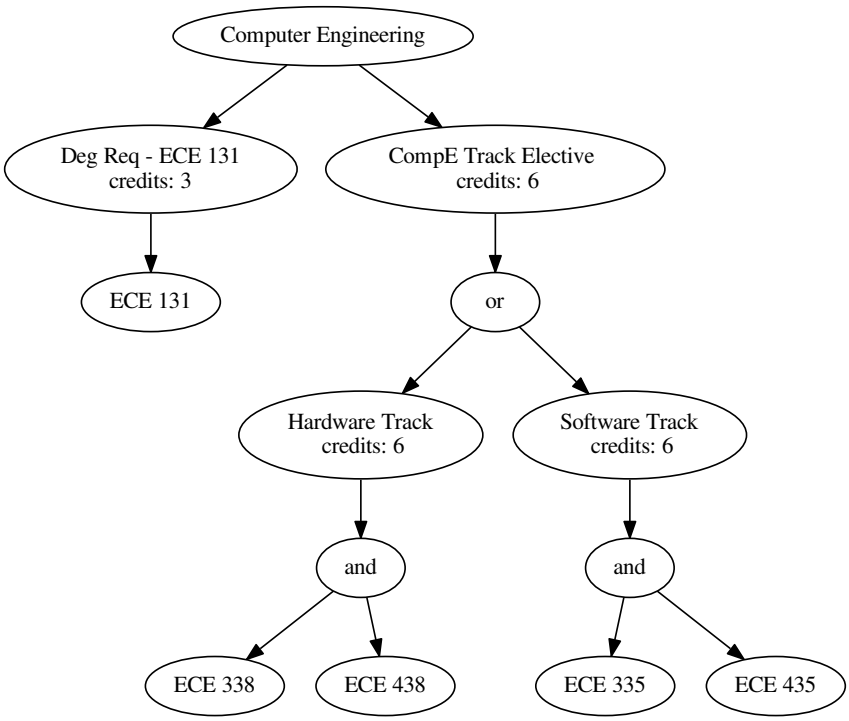


Figure 5.3: Sample of Degree Requirements Tree

In order to accurately represent the degree requirements of a program, many different graph constructs were considered. One of the main issues that was considered

was how departments described their degree requirements. Across all of the colleges and schools, there are two general ways in which degree requirements can be defined: a list of enumerated courses or a free selection of courses, possibly scoped by department or level

In order to create the degree requirements tree, special aggregation nodes were used. These aggregation nodes represented a single degree requirement, and were placed between the nodes representing an academic program (in this case *Computer Engineering*) and the courses that satisfy the specific requirement. The degree requirement nodes also contain information on how many credits are needed to satisfy the degree requirement. For example, the *CompE Track Elective* node in Figure 5.3 has a constraint of six credits. This implies that in order for this requirement to be satisfactorily satisfied, a sufficient number of credits must be taken from the courses that are children of the node in the graph.

5.1.2 Document Database

The second major data persistence component of the system is a document database. The document database provides a convenient manner which to store and deliver degree plans. In this application, the database of record for curriculum (e.g., colleges, departments, degree plans, courses, etc.) is the graph database. The document database is used to store both official degree plans created by departments, and those created by students with the aide of their advisors.

The information about curriculum at the University of New Mexico is relatively static and very seldom changes between semesters. Also, the data that is stored for each degree plan varies by college and department. These properties of the data stored in this database fit MongoDB's concepts of schema-free operation and BASE consistency model.

5.1.3 Departmental Degree Plans

The MongoDB database is used essentially as a large, longer persisting caching database that stores degree plans. The degree plans are constructed using the degree requirements that are stored in the Neo4j database. The degree requirements are retrieved from the graph database, and the user is able to order them into semesters. A degree plan describes the courses to be taken for each semester at the University of New Mexico.

These departmental degree plans are then serialized as BSON and stored in the document database as single documents. This makes retrieving and presenting degree plans to the user a very straightforward proposition. The degree plan can be pulled from the database in one piece, without the need for costly joins that would be associated with a relational data model. Also, since the data is already stored in JSON-like format, no conversion is required before the degree plan is able to be sent over any application interfaces.

5.1.4 Custom Degree Plans

This database can also be used to store custom degree plans for individual students. A student, with the aide of an advisor, would be able to retrieve the current departmental degree plan and then customize it to fit his or her needs. Then, whether or not a student has completed a course, or any notes specific to that student would be able to be stored in a custom degree plan for later retrieval.

5.1.5 Other Stored Data

In addition to storing the degree plans, the MongoDB database can also be utilized to store other information pertinent to curriculum at the University of New Mexico.

One piece of information that could be useful to store would be a document that describes the node-edge relationship of the graph that is used to show prerequisites for all of the courses within a degree plan.

The D3.js (D3) JavaScript library is used in the current application. D3 is a data visualization library used in order to generate the graph view of a degree plan. Being a JavaScript library, D3 requires data to be in JSON format in order for it to be rendered. Currently, the JSON used to render a graph must be created dynamically. Storing the data natively as a BSON document would free up resources from data conversion tasks at run-time.

5.2 Advantages

5.2.1 Graph Database

Flexibility. A main advantage of using the graph database is the ability to add attributes to the data model without having to bother with conforming to a schema. Take for instance the issue of adding an institution level entity to the diagram in Figure 5.1. Using a relational model would require the use of a join table, and there would also be the issue of what types of entities could be joined using that table. Unless some form of polymorphism was used, only entities represented by certain tables could be joined.

In Neo4j, a new node *University of New Mexico* would simply have to be added to the graph and a relationship between the new node and the node *School of Engineering* could be added. While the entities represented by the nodes could very well have a "type," Neo4j is agnostic on this fact, and can relate any two nodes regardless of type. Figure 5.4 shows a diagram of the graph with the added entity.

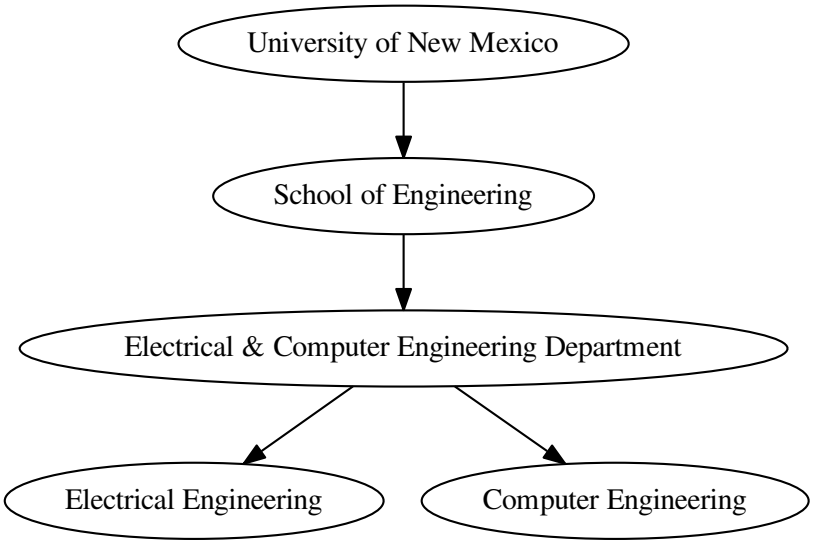


Figure 5.4: University Relationships With Top Level Institution Node

Using this method also leads to a more condensed search criteria, for example a search for all of the academic programs at a university. In order to perform this operation in a relational database, first a join of the given university and all of its departments would need to take place, and then all of the subsequent joins of the colleges to departments and departments to academic program would have to occur. However, this model would not always accurately model all universities in the state. At Central New Mexico Community College, colleges administer degree plans, departments do not even exist. There would have to be certain measures to handle this exception as well.

As stated above, Neo4j does not rely on joins to describe relationships, and accessing them becomes a simple task. In order to find all of the academic programs for a given university, the following cypher query in Figure 5.5 could be used. In this cypher query, `-[*]-` is used to denote any number or type of connections to

AcademicProgram from Institution.

```
1 MATCH (i:Institution { name:"University of New Mexico" })-[*]-(ap:AcademicProgram)
2 RETURN ap
```

Figure 5.5: Cypher Query to Find All Academic Programs at UNM

This script returns all nodes that are of type AcademicProgram that are any number of relationships away from the institution node University of New Mexico. Using this script, the academic programs for any university could be found no matter how many nodes exist between.

Prerequisite Simplification. Storing the university data natively in a graph also allowed the ability to easily find unnecessary prerequisites within a chain of courses. CHEM 121 General Chemistry with Lab has the following prerequisites according to the University of New Mexico catalog:

> *MATH 121 or MATH 123 or MATH 150 or MATH 162 or MATH 163 or MATH 180 or MATH 181 or MATH 264*

Figure 5.6 is a graph representation of all of the courses mentioned above with prerequisite relationships denoted by edges of the graph. For the sake of prerequisite reduction, the "and" and "or" relationships described in the prerequisite chains can be ignored.

Figure 5.7 shows a diagram that illustrates the reduced graph for the prerequisites of CHEM 121. This graph is greatly simplified and shows that taking the course MATH 121 is sufficient to satisfy the prerequisites for CHEM 121.

In order to reduce the prerequisite set of CHEM 121 down to simply MATH 121, a transitive reduction algorithm was used. A transitive reduction of the directed graph G is G'. G' has the smallest number edges such that if a path exists between

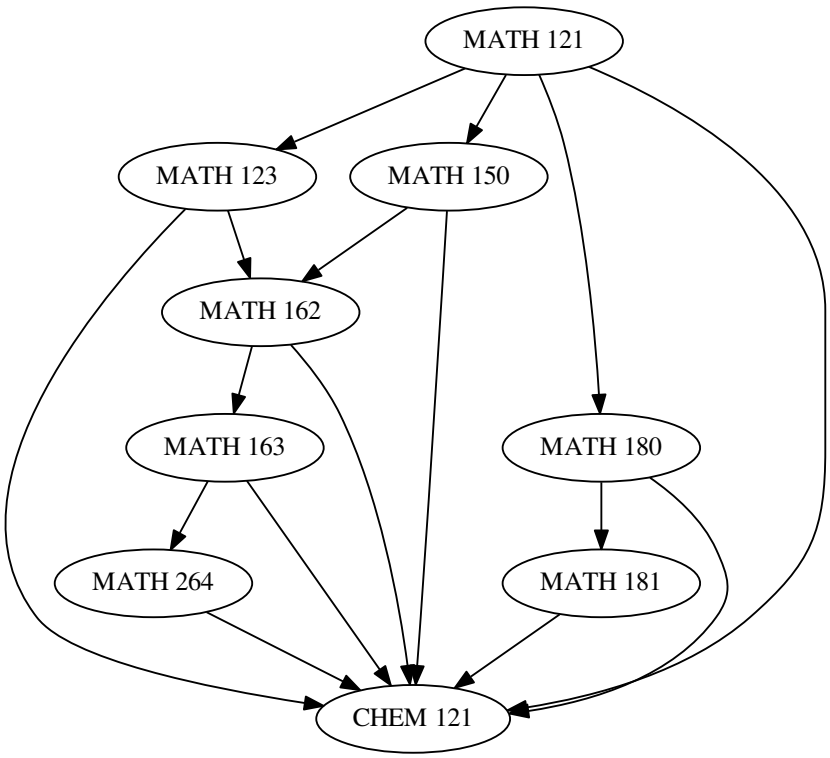


Figure 5.6: CHEM 121 Prerequisite Graph as described in UNM Catalog

two nodes in G, it will also exist in G' [18]. For the graph in Figure 5.6, this has the effect of removing redundant prerequisites. This greatly simplifies the graph of all of the courses at the University of New Mexico and facilitates description of cruciality of courses.

The cruciality of courses took into account two factors of the course. The first factor was the delay factor. A course with a high delay factor was on the critical path, and delays in taking them would delay the student taking courses in subsequent semesters. The second measure was blocking factor, in other words, how many courses could a student take if he or she completed another course. Once the cruciality of the courses are known, decisions can be made that aid students in flowing

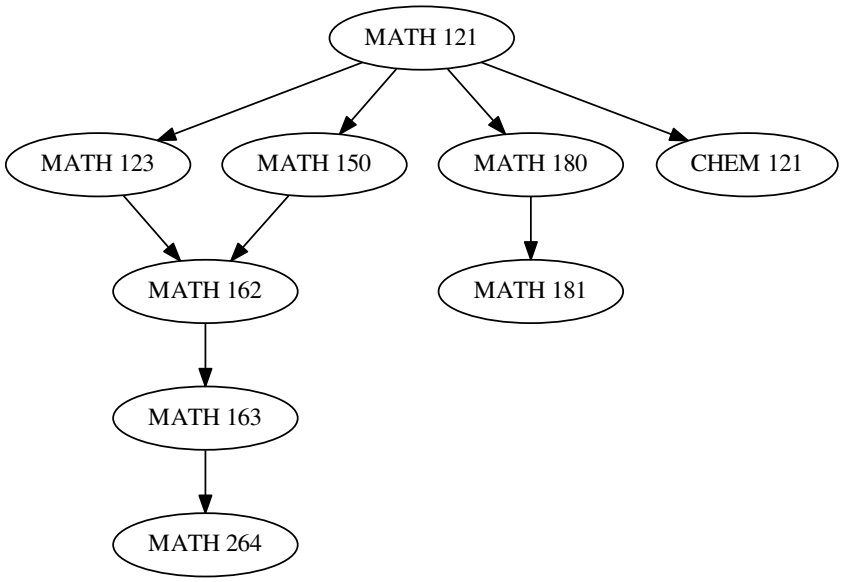


Figure 5.7: Reduced CHEM 121 Prerequisite Graph

through the curriculum [17].

5.2.2 Document Database

The manner in which the requirements are structured in the graph database also makes it hard to perform a 1-to-1 mapping of requirements to terms. Take for instance the "Technical Elective" requirement for Computer Engineering program. This requirement states that a student must take nine credit hours of courses in Electrical & Computer Engineering, Computer Science, Physics, or other engineering departments. Trying to find a way to map this requirement to a specific item in a term would be difficult, this single requirement would have to be spread out among several semesters.

Whenever a new degree plan is created and saved in the document database, it

can be checked against the requirements saved in the graph database in order to ensure that the courses outlined in the plan are sufficient to receive a degree.

Creation of degree plans is only performed at certain points in the year, meaning access to the database would be read-heavy. Once a degree plan is created by a student, they are generally used until he or she graduates. The degree plan may however be read thousands of times before a new degree plan is created. This allows the MongoDB database to act, at least in some way, as a long-term caching database.

As mentioned above, the MongoDB database could also be used in order to store custom degree plans for individual students. If a student wishes to create a custom degree plan, that plan could be stored within the MongoDB database. A departmental degree plan could serve as the template for this, or a set or sets of requirements could be loaded from the Neo4j database and used in order to create the custom plan.

Chapter 6

Futurework/Conclusions

The improvements described in this thesis to the currently deployed system will provide many benefits to both the university and the students it serves. Selecting a data store that best fits the data model of the problem domain is extremely beneficial. For example, modeling the courses as a graph and storing them natively in a Neo4j database allows operations to be performed that would have been difficult had the data been stored in another way. Storing the degree plans as documents in MongoDB also allows for faster access and flexibility in the type of data that can be stored.

One area that will require work in the future is a system that will load degree requirements from the graph database and allow a user to create degree plans. This system will also have to guarantee that a degree plan satisfies a set of degree requirements before it is allowed to be saved. This system that coordinates the two databases will be necessary for the cloud based graph-document data store system to function properly.

The system that is described in this paper is one part of a planned, larger system. Figure 6.1 shows the future vision for this project.

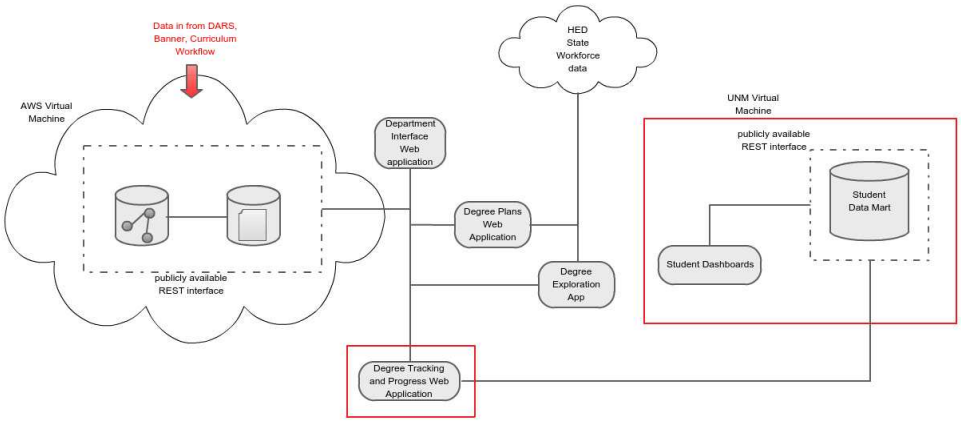


Figure 6.1: Envisioned System

This system is designed to integrate with other systems already at the University of New Mexico, both existing and in production. The cloud based graph-document data store system, located on the left side of the diagram, will serve to be the backbone of the analytics system that will be developed.

The system must also be able to integrate data from the University of New Mexico Office of Institutional Analytics Datamart. It is here that student data can be found, and used to compare with the degree plans of record stored in the cloud based graph-document data store system. This will allow a student's progress to be accurately measured.

References

[1] Active record api. http://api.rubyonrails.org/classes/ActiveRecord/Base.html. Accessed: 2014-10-28.

[2] Bson. http://bsonspec.org/, 2014. Accessed: 2014-10-21.

[3] Max De Marzi. Neography on github. https://github.com/maxdemarzi/neography. Accessed: 2014-09-21.

[4] Exploring the different types of nosql databases. http://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases. Accessed: 2014-10-21.

[5] General ed core course transfer curriculum. http://www.hed.state.nm.us/institutions/general-ed-core-course-transfer-curriculum.aspx. Accessed: 2014-10-01.

[6] Seth Gilbert and Nancy Lynch. Perspectives on the CAP Theorem. *Computer*, 45(2):30–36, 2012.

[7] Theo Häerder and Andreas Reuter. Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4):287–317, 1983.

[8] Heroku dev center. https://devcenter.heroku.com/. Accessed: 2014-10-14.

[9] Introduction to mongodb. http://docs.mongodb.org/manual/core/introduction/, 2014. Accessed: 2014-10-22.

[10] Mongodb data modeling introduction. http://docs.mongodb.org/manual/core/data-modeling-introduction/, 2014. Accessed: 2014-10-21.

[11] Mongodb replica set high availability. http://docs.mongodb.org/manual/core/replica-set-high-availability/, 2014. Accessed: 2014-10-21.

[12] Nosql databases explained. `http://www.mongodb.com/nosql-explained`. Accessed: 2014-10-21.

[13] D Pritchett. Base: An acid alternative. *Queue*, 2008.

[14] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*. O'Reilly Media, Inc., first edition, 2013.

[15] Sam Ruby, Dave Thomas, and David Heinemeier Hansson. *Agile Web Development with Rails*. Pragmatic Programmers, LLC., fourth edition, 2012.

[16] Ruby on rails api. `http://api.rubyonrails.org/`. Accessed: 2014-10-20.

[17] Ahmad Slim, Jarred Kozlick, Gregory L. Heileman, and Chaouki T. Abdallah. The Complexity of University Curricula According to Course Cruciality. *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, pages 242–248, July 2014.

[18] Transitive reduction. `http://xlinux.nist.gov/dads//HTML/transitiveReduction.html`. Accessed: 2014-10-14.

[19] Web application framework. `http://docforge.com/wiki/Web_application_framework`. Accessed: 2014-10-21.

[20] What is a graph database? `http://neo4j.com/developer/graph-database/`, 2014. Accessed: 2014-09-17.

[21] Joseph Yeado, Kati Haycock, Rob Johnstone, and Priyadarshini Chaplot. Learning From High-Performing and Fast-Gaining Institutions. Technical report, Education Trust, 2014.

Appendices

Appendix A

Course JSON Files

The following is an abbreviated section of the JSON files supplied by the University of New Mexico IT.

```
[
    {
        "name": "ECE 101",
        "title": "Intro to Elect & Computer Eng",
        "credits": "1",
        "description ": "Insight into electrical engineering is gained through videos,
             hands-on  experiments, use of computer software to learn basic problem-
             solving skills and a team-oriented design project.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 131",
        "title": "Programming Fundamentals",
        "credits": "3",
        "description ": "Fundamental programming concepts, including consideration of
             abstract machine models with emphasis on the memory hierachy, basic
             programming constructs, functions, parameter passing, pointers and arrays,
             file I/O, bit-level operations and interfacing to external devices.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 203",
        "title": "Circuit Analysis I",
        "credits": "3",
        "description ": "Basic elements and sources.  Energy and power.  Ohm's law and
             Kirchhoff's laws.  Resistive networks, node and loop analysis.  Network
             theorems.  First-order and second-order circuits.  Sinusoidal sources and
```

```
            complex representations: impendance, phasors, complex power. Three-phase
            circuits. Prerequisite: ECE 131 and MATH 163. Pre- or Corequisite: MATH
            316 and PHYC 161.",
25          "department ": "Electrical Computer Engr",
26          "prerequisites": [
27              {
28                  "name": "ECE 131"
29              },
30              {
31                  "name": " and MATH 163"
32              },
33              {
34                  "name": " and PHYC 161(May be Concurrent)"
35              },
36              {
37                  "name": " and MATH 316(May be Concurrent)"
38              }
39          ],
40          "corequisites": [
41              {
42                  "name": "MATH 316"
43              },
44              {
45                  "name": "PHYC 161"
46              }
47          ]
48      },
49      {
50          "name": "ECE 206L",
51          "title": "Instrumentation",
52          "credits": "2",
53          "description ": "Introduction to laboratory practices and the use of test
            equipment. Measurements on basic electrical components, dc and ac circuits
            using ohmmeters, voltmeters, ammeters and oscilloscopes. Circuit simulation.
            Prerequisites: 203L and ENGL 102.",
54          "department ": "Electrical Computer Engr",
55          "prerequisites": [
56              {
57                  "name": "( ECE 203L"
58              },
59              {
60                  "name": " or EECE 203L"
61              },
62              {
63                  "name": " or ECE 203 )"
64              },
65              {
66                  "name": " and ( ENGL 102"
67              },
68              {
69                  "name": " or ACT English  minimum score 29"
70              },
71              {
72                  "name": " or SAT Verbal  minimum score 650 )"
73              },
74              {
75                  "name": " or ENGL 102 Portfolio  minimum score 1"
76              }
```

```
        ],
        "corequisites": []
    },
    {
        "name": "ECE 213",
        "title": "Circuit Analysis II",
        "credits": "3",
        "description ": "General transient analysis of electrical circuits. Laplace
            transform with applications to circuit analysis. State-space equations.
            Fourier series analysis. The network function; convolution; frequency
            response. Prerequisites: 203L and MATH 316. Corequisite: MATH 314.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "( ECE 203L"
            },
            {
                "name": " or ECE 203"
            },
            {
                "name": " or EECE 203L )"
            },
            {
                "name": " and MATH 316"
            }
        ],
        "corequisites": [
            {
                "name": "MATH 314"
            }
        ]
    },
    {
        "name": "ECE 231",
        "title": "Intermediate Programming",
        "credits": "3",
        "description ": "Introducton to elementary data structures, program design and
            computer-based solution of engineering problems. Topics include use of
            pointers, stacks, queues, linked lists, trees, graphs, systems and device-
            level programming and software design methodology. Prerequisite: ECE 131.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 131"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 238L",
        "title": "Computer Logic Design",
        "credits": "4",
        "description ": "Binary number systems. Boolean algebra. Combinational,
            sequential and register transfer logic. VHDL. Arithmetic/logic unit. Memories
            , computer organization. Input-output. Microprocessors.  Prerequisites: ECE
            131.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
```

```
            {
                "name": "ECE 131"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 314",
        "title": "Signals and Systems",
        "credits": "3",
        "description ": "Continuous and discrete time signals and systems; time and
             frequency domain analysis of LTI systems, Fourier series and transforms,
             discrete time Fourier series/transform sampling theorem, block diagrams,
             modulation/demodulation, filters. Prerequisites: 213 and MATH 264.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 213"
            },
            {
                "name": " and MATH 264"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 321L",
        "title": "Electronics I",
        "credits": "4",
        "description ": "Introduction to diodes, bipolar and field-effect transistors.
             Analysis and design of digital circuits, gates, flip-flops and memory
             circuits. Circuits employing operational amplifiers. Analog to digital and
             digital to analog converters. Prerequisite: 213.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 213"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 322L",
        "title": "Electronics II",
        "credits": "4",
        "description ": "Analysis, design, and characterization of linear circuits
             including operational amplifiers. Design of biasing and reference circuits,
             multistage amplifiers, and feedback circuits. Prerequisite: 321L.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 321L"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 330",
```

```
        "title": "Software Design",
        "credits": "3",
        "description ": "Design of software systems using modern modeling techniques.
             Relationship between software design and process, with emphasis on UML and
             its interface application code.  Exposure to design patterns, software
             frameworks, and software architectural paradigms.  Prerequisite:  231.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 231"
            },
            {
                "name": " or EECE 231L"
            },
            {
                "name": " or ECE 231L"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 331",
        "title": "Data Structures & Algorithms",
        "credits": "3",
        "description ": "An introduction to data structures and algorithms. Topics
             include asymptotic notation recurrence relations, sorting, hash tables, basic
              priority queues, balanced search trees and basic graph representation and
             search. Prerequisite: 231 and MATH 327. Corequisite: 340.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 231"
            },
            {
                "name": " and MATH 327"
            }
        ],
        "corequisites": [
            {
                "name": "ECE 340"
            }
        ]
    },
    {
        "name": "ECE 335",
        "title": "Integrated Software Systems",
        "credits": "3",
        "description ": "Course considers design principles, implementation issues, and
             performance evaluation of various software paradigms in an integrated
             computing environment. Topics include performance measurement and evaluation,
              program optimization for the underlying architeccture, integration and
             security for large-scale software systems.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 330"
            },
            {
```

```
                "name": " and ECE 337"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 337",
        "title": "Computer Arch & Organization",
        "credits": "3",
        "description ": "Survey of various levels of computer architecture and design;
             microprogramming and processor architecture, assembly language programming,
             operating system concepts and input/output via the operating system. Three
             lectures, 1 hr. lab. Prerequisites: 231 and 238L. (Spring)",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "( ECE 231"
            },
            {
                "name": " or EECE 231L"
            },
            {
                "name": " or ECE 231L )"
            },
            {
                "name": " and ECE 238L"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 338",
        "title": "Intermediate Logic Design",
        "credits": "3",
        "description ": "Advanced combinational circuits; XOR and transmission gates;
             computer-based optimization methods; RTL and HDL; introduction to computer
             aided design; advanced sequential machines; asynchronous sequential machines;
              timing issues; memory and memory interfacing; programmable logic devices;
             and VLSI concepts. Prerequisite: 238L.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 238L"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 340",
        "title": "Probabilistic Methods in Eng",
        "credits": "3",
        "description ": "Introduction to probability, random variables, random processes,
              probability distribution/density functions, expectation correlation, power
             spectrum, WSS processes, confidence internals, transmission through LIT
             systems, applications of probability. Prerequisite: 314 and MATH 314.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
```

```
                "name": "ECE 314"
            },
            {
                "name": " and MATH 314"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 341",
        "title": "Communication Systems",
        "credits": "3",
        "description ": "Amplitude/frequency modulation, pulse position/amplitude
            modulation, probabilistic noise model, AWGN, Rice representation, figure of
            merit, phase locked loops, digital modulation, introduction to multiple
            access systems. Prerequisite: 314 and 340.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 314"
            },
            {
                "name": " and ECE 340"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 344L",
        "title": "Microprocessors",
        "credits": "4",
        "description ": "Computers and Microprocessors: architecture, assembly language
            programming, input/output and applications. Prerequisite: 206L and 238L and
            321L. Three lectures, 3 hrs. lab. (Fall, Spring)",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 206L"
            },
            {
                "name": " and ECE 238L"
            },
            {
                "name": " and ECE 321L"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 345",
        "title": "Intro to Control Systems",
        "credits": "3",
        "description ": "Introduction to the feedback control problem. Plant modeling,
            transfer function and state-space descriptions. Stability criteria. Nyquist
            and root-locus design. Introduction to analytical design. Z-transforms and
            digital control. Laboratory design project. Prerequisite: 314.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
```

```
            {
                "name": "ECE 314"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 360",
        "title": "Electromagnetic Fields & Waves",
        "credits": "3",
        "description ": "Maxwell s equations, plane wave propagation, waveguides and
            transmission lines, transient pulse propagation and elementary dipole antenna
            . Prerequisites: 213 and PHYC 161 and MATH 264.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 213"
            },
            {
                "name": " and PHYC 161"
            },
            {
                "name": " and MATH 264"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 371",
        "title": "Materials & Devices",
        "credits": "4",
        "description ": "Introduction to quantum mechanics, crystal structures,
            insulators, metals, and semiconductor material properties, bipolar, field
            effect and light emitting devices.  Prerequisite: PHYC 262.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "PHYC 262"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 381",
        "title": " Intro to Power Systems",
        "credits": "3",
        "description ": "Provides in-depth look at various elements of power systems
            including power generation, transformer action, transmission line modeling,
            symmetrical components, pf correction, real/quadrature power calculations,
            load flow analysis and economic considerations in operating systems.
            Prerequisite:  213.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 213"
            }
        ],
        "corequisites": []
```

```
    },
    {
        "name": "ECE 412",
        "title": "Intro to Computer Graphics",
        "credits": "3",
        "description ": "(Also offered as CS 412) Introduction to technical aspects of
             raster algorithms in computer graphics.  Foundational concepts of 2-D and 3-D
              graphics as relate to real-time and offline techniques.  Students develop a
             video game as a final project to demonstrate the algorithms learned in class.
               Prerequisite:  361L or ECE 331.",
        "department ": "Computer Science",
        "prerequisites": [
            {
                "name": "ECE 331"
            },
            {
                "name": " or CS 361L"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 413",
        "title": "Intro to Ray Graphics",
        "credits": "3",
        "description ": "Topics include ray-geometry intersections, viewing, lenses,
             local/global illumination, procedural textures/,models, spline curves and
             surfaces, statistical integration for realistic image synthesis.  Students
             will write a raytracing renderer from scratch, exploring high performance
             implementations and realistic rendering.Prerequisite: 331 or CS 361L.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "CS 361L"
            },
            {
                "name": " or ECE 331"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 419",
        "title": "Senior Design I",
        "credits": "3",
        "description ": "Design methodology and development of professional project-
             oriented skills including communication, team management, economics, and
             engineering ethics. Working in teams, a proposal for a large design is
             prepared in response to an industrial or in-house sponsor. Restriction: ECE
             major, senior standing.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 420",
        "title": "Senior Design II",
        "credits": "3",
```

```
        "description ": "Continuation of 419. Students work in assigned teams to
             implement proposal developed in 419. Prototypes are built and tested to
             sponsor specifications, and oral and written reports made to the project
             sponsor. Prerequisite: 419.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 419"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 421",
        "title": "Analog Electronics",
        "credits": "3",
        "description ": "Design of advanced analog electronic circuits. BJT and MOSFET
             operational amplifiers, current mirrors and output stages. Frequency response
              and compensation. Noise. A/D and D/A converters. Prerequisite: 322L.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 322L"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 424",
        "title": "Digital VLSI Design",
        "credits": "3",
        "description ": "CMOS logic gates and circuits, transistor implementations,
             applications to sequential circuits, VLSI data path and controller design,
             VLSI routing issues and architectures, RTL and VLSI impacts and applications
             to microprocessor design. Prerequisites: 321L and 338.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 321L"
            },
            {
                "name": " and ECE 338"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 432",
        "title": "Intro to Parallel Processing",
        "credits": "3",
        "description ": "(Also offered as CS 442.) Machine taxonomy and introduction to
             parallel programming. Performance issues, speed-up and efficiency.
             Interconnection networks and embeddings. Parallel programming issues and
             models: control parallel, data parallel and data flow. Programming
             assignments on massively parallel machines. Prerequisites: (331 or CS 351L)
             and (337 or CS 341L).",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
```

```
            {
                "name": "( ECE 331"
            },
            {
                "name": " or CS 351L )"
            },
            {
                "name": " and ( ECE 337"
            },
            {
                "name": " or CS 341L )"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 435",
        "title": "Software Engineering",
        "credits": "3",
        "description ": "Management and technical issues including business conduct and
             ethics related to the design of large engineering projects. Student teams
             will address the design, specification, implementation, testing and
             documentation of a large hardware/software project. Prerequisites: 331 and
             335.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 331"
            },
            {
                "name": " and ECE 335"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 437",
        "title": "Computer Operating Systems",
        "credits": "3",
        "description ": "(Also offered as CS 481.) Fundamental principles of modern
             operating systems design, with emphasis on concurrency and resource
             management. Topics include processes, interprocess communication, semaphores,
              monitors, message passing, input/output device, deadlocks memory management,
              files system design. Prerequisites: (330 and 337) or CS 341L.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "( ECE 330"
            },
            {
                "name": " and ECE 337 )"
            },
            {
                "name": " or CS 341L"
            }
        ],
        "corequisites": []
    },
```

```
513     {
514         "name": "ECE 438",
515         "title": "Design of Computers",
516         "credits": "3",
517         "description ": "Computer architecture; design and implementation at HDL level;
                ALU, exception handling and interrupts; addressing; memory; speed issues;
                pipelining; microprogramming; introduction to distributed and parallel
                processing; buses; bus protocols and bus masters. CAD project to include
                written and oral presentations. Prerequisites: 337 and 338 and 344L.",
518         "department ": "Electrical Computer Engr",
519         "prerequisites": [
520             {
521                 "name": "ECE 337"
522             },
523             {
524                 "name": " and ECE 338"
525             },
526             {
527                 "name": " and ECE 344L"
528             }
529         ],
530         "corequisites": []
531     },
532     {
533         "name": "ECE 439",
534         "title": "Intr Digital Signal Processing",
535         "credits": "3",
536         "description ": "Bilateral Z transforms, region of convergence, review of
                sampling theorem, aliasing, the discrete Fourier transform and properties,
                analysis/design of FIR/IIR filters, FFT algorithms spectral analysis using
                FFT. Prerequisite: 314.",
537         "department ": "Electrical Computer Engr",
538         "prerequisites": [
539             {
540                 "name": "ECE 314"
541             }
542         ],
543         "corequisites": []
544     },
545     {
546         "name": "ECE 440",
547         "title": "Computer Networks",
548         "credits": "3",
549         "description ": "(Also offered as CS 485.) Theoretical and practical study of
                computer networks, including network structures and architectures. Principles
                 of digital communications systems. Network topologies, protocols and
                services. TCP/IP protocol suite. Point-to-point networks; broadcast networks;
                 local area networks; routing, error and flow control techniques.
                Prerequisites: 330 and 337. Corequisite: 340.",
550         "department ": "Electrical Computer Engr",
551         "prerequisites": [],
552         "corequisites": [
553             {
554                 "name": "ECE 340"
555             }
556         ]
557     },
558     {
```

```
        "name": "ECE 441",
        "title": "Communication Systems",
        "credits": "3",
        "description ": "Amplitude/frequency modulation, pulse position/amplitude
            modulation, probabilistic noise model, AWGN, Rice representation, figure of
            merit, phase locked loops, digital modulation, introduction to multiple
            access systems. Prerequisites: 314 and 340.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 314"
            },
            {
                "name": " and ECE 340"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 442",
        "title": "Wireless Communication",
        "credits": "3",
        "description ": "The course is an introduction to cellular telephone systems and
            wireless networks, drawing upon a diversity of electrical engineering areas.
            Topics include cellular concepts, radio propagation, modulation methods and
            multiple access techniques. Prerequisite: 314 and 360.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 314"
            },
            {
                "name": " and ECE 360"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 443",
        "title": "Hardware Design with VHDL",
        "credits": "3",
        "description ": "The VHDL hardware description language is used for description
            of digital systems at several levels of complexity, from the system level to
            the gate level. Descriptions provide a mechanism for documentation, for
            simulation and for synthesis. Prerequisite: 338.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 338"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 446",
        "title": "Feedback Control Systems",
        "credits": "3",
```

```
        "description ": "Modeling of continuous and sampled-data control systems. State-
            space representation. Sensitivity, stability and optimization of control
            systems. Design of compensators in the frequency and time domains. Phase-
            plane, describing function design for non-linear systems, and laboratory
            design project.  Prerequisite: 345.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 345"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 448",
        "title": "Fuzzy Logic with Applications",
        "credits": "3",
        "description ": "(Also offered as CE 448.) Theory of fuzzy sets; foundations of
            fuzzy logic. Fuzzy logic is shown to contain evidence, possibility and
            probability logics; course emphasizes engineering applications; control,
            pattern recognition, damage assessment, decisions; hardware/software
            demonstrations. ",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 456",
        "title": "Entrepreneurial Engineering",
        "credits": "3",
        "description ": "Review and application of necessary elements for successfully
            launching technical businesses; focuses upon technology, manufacturing,
            management, marketing, legal and financial aspects. Students work in groups
            developing elements of new businesses and producing business plans.
            Restriction: senior standing.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 460",
        "title": "Microwave Engineering",
        "credits": "3",
        "description ": "This lecture/laboratory course provides essential fundamentals
            for rf, wireless and microwave engineering. Topics include: wave propagation
            in cables, waveguides and free space; impedance matching, standing wave
            ratios, Z- and S- parameters. Prerequisite: 360.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 360"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 463",
        "title": "Advanced Optics I",
```

```
        "credits": "3",
        "description ": "(Also offered as PHYC 463.) Electromagnetic theory of
             geometrical optics, Gaussian ray tracing and matrix methods, finite ray
             tracing, aberrations, interference and diffraction. Prerequisite: PHYC 302.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "PHYC 463"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 464",
        "title": "Laser Physics I",
        "credits": "3",
        "description ": "(Also offered as PHYC 464.) Resonator optics. Rate equations;
             spontaneous and stimuated emission; gas, semiconductor and solid state lasers
             , pulsed and mode-locked laser techniques. Prerequisite: 360 or PHYC 406.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 360"
            },
            {
                "name": " or PHYC 406"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 469",
        "title": "Antennas for Wireless Comm",
        "credits": "3",
        "description ": "Aspects of antenna theory and design; radiation from dipoles,
             loops, apertures, microstrip antennas and antenna arrays. Prerequisite: 360."
             ,
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 360"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 471",
        "title": "Materials & Devices II",
        "credits": "3",
        "description ": "An intermediate study of semiconductor materials, energy band
             structure, p-n junctions, ideal and non-ideal effects in field effect and
             bipolar transistors. Prerequisites: 360 and 371.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 360"
            },
            {
```

```
                "name": " and ECE 371"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 474L",
        "title": "Microelectronics Processing I",
        "credits": "3",
        "description ": "(Also offered as NSMS 574L.) Materials science of semiconductors
             , microelectronics technologies, device/circuit fabrication, parasitics and
             packaging. Lab project features small group design/fabrication/testing of MOS
              circuits.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 475",
        "title": "Optoelectronics",
        "credits": "3",
        "description ": "Basic electro-optics and opto-electronics, with engineering
             applications. Interaction of light with matter. Introduction to optics of
             dielectrics, metals and crystals. Introductory descriptions of electro-optic,
              acousto-optic and magneto-optic effects and related devices. Light sources,
             displays and detectors. Elementary theory and applications of lasers, optical
              waveguides and fibers. Prerequisite: 371.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 371"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 482",
        "title": "Electric Drives & Transformers",
        "credits": "3",
        "description ": "Electromagnetic theory and mechanical considerations are
             employed to develop models for and understanding of Transformers, Induction
             Machines and Synchronous Machines. Additionally, DC Machines are discussed.
             Prerequisite: 203 and 213.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 203"
            },
            {
                "name": " and ECE 213"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 483",
        "title": "Power Electronics",
        "credits": "3",
```

```
        "description ": "Introduces modern power conversion techniques at a lower level,
             dealing with basic structures of power converters and techniques of analyzing
              converter circuits. Students learn to analyze and design suitable circuits
             and subsystems for practical applications. Prerquisite: 321L and 371 and 381.
             ",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 321L"
            },
            {
                "name": " and ECE 371"
            },
            {
                "name": " and ECE 381"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 484",
        "title": "Photovoltaics",
        "credits": "3",
        "description ": "Technical concepts of photovoltaics. Solar cell device level
             operation, packaging, manufacturing, designing phovoltaic system for stand-
             alone or grid-tied operation, some business-case analysis and some real-life
             scenarios of applicability of these solutions. Prerequisite: 381 and MATH
             121.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 381"
            },
            {
                "name": " and MATH 121"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 488",
        "title": "Future Energy Systems",
        "credits": "3",
        "description ": "A detailed study of current and emerging power and energy
             systems and technologies. Including renewable energies, storage, Smart Grid
             concepts, security for power infrastructure. Software modeling of power
             systems and grids. Prerequisite: 381 and 482 and 483 and 484.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [
            {
                "name": "ECE 381"
            },
            {
                "name": " and ECE 482"
            },
            {
                "name": " and ECE 483"
            },
```

```
            {
                "name": " and ECE 484"
            }
        ],
        "corequisites": []
    },
    {
        "name": "ECE 490",
        "title": "Internship",
        "credits": "3",
        "description ": "Professional practice under the guidance of a practicing
            engineer. Assignments include design or analysis of systems or hardware, or
            computer programming. A preliminary proposal and periodic reports are
            required. The engineer evaluates student s work; a faculty monitor assigns
            grade.  Restriction: ECE major, junior standing. (12 hours/week) (24 hours/
            week in summer session). Offered on a CR/NC basis only.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 491",
        "title": "Undergrad Problems",
        "credits": "1 TO 6",
        "description ": "Registration for more than 3 hours requires permission of
            department chairperson.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 493",
        "title": "Honors Seminar",
        "credits": "1 TO 3",
        "description ": "A special seminar open only to honors students. Registration
            requires permission of department chairperson.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 494",
        "title": "Honors Individual Study",
        "credits": "1 TO 6",
        "description ": "Open only to honors students. Registration requires permission
            of the department chairperson and of the supervising professor.",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
    },
    {
        "name": "ECE 495",
        "title": "Special Topics",
        "credits": "1 TO 4",
        "description ": "Restriction: ECE major, senior standing.  ",
        "department ": "Electrical Computer Engr",
        "prerequisites": [],
        "corequisites": []
```

```
844         }
845     ]
```

Appendix B

Course Parsing Scripts

The following code was used to parse the JSON files in Appendix A into a seed file format that was able to be loaded into the database.

```
require 'open-uri'
require 'json'
require 'active_support/all'
require_relative './parse_array.rb'

weasel_depts = ["AFST", "AMST", "ANTH", "ARTH", "ASM", "AS_MULTI_COURSES", "BIOC", "BIOC", "BIOL", "CCS", "CE", "CFA_MULTI_COURSES", "CHEM", "CHNE", "CJ", "COE_MULTI_COURSES", "CS", "DANC", "DOS", "ECE", "ECON", "ELOL", "ENGL", "EPS", "ES", "FITE", "FLL", "GEOG", "HC", "HESS", "HIST", "HSC", "IFCE", "IFDM", "LAW", "LING", "LLSS", "MA", "MATH", "ME", "MEDL", "MIDS", "MUS", "NSMS", "NURS", "PHIL", "PHRM", "PHYC", "POLS", "PSY", "RADS", "RELG", "SAAP", "SHS", "SOC", "SOE_MULTI_COURSES", "SPA", "SPANPORT", "TED", "UC", "UL", "WMST"]
course_array = {"AFST" => [], "AMST" => [], "ANTH" => [], "ARTH" => [], "ASM" => [], "AS_MULTI_COURSES" => [], "BIOC" => [], "BIOC" => [], "BIOL" => [], "CCS" => [], "CE" => [], "CFA_MULTI_COURSES" => [], "CHEM" => [], "CHNE" => [], "CJ" => [], "COE_MULTI_COURSES" => [], "CS" => [], "DANC" => [], "DOS" => [], "ECE" => [], "ECON" => [], "ELOL" => [], "ENGL" => [], "EPS" => [], "ES" => [], "FITE" => [], "FLL" => [], "GEOG" => [], "HC" => [], "HESS" => [], "HIST" => [], "HSC" => [], "IFCE" => [], "IFDM" => [], "LAW" => [], "LING" => [], "LLSS" => [], "MA" => [], "MATH" => [], "ME" => [], "MEDL" => [], "MIDS" => [], "MUS" => [], "NSMS" => [], "NURS" => [], "PHIL" => [], "PHRM" => [], "PHYC" => [], "POLS" => [], "PSY" => [], "RADS" => [], "RELG" => [], "SAAP" => [], "SHS" => [], "SOC" => [], "SOE_MULTI_COURSES" => [], "SPA" => [], "SPANPORT" => [], "TED" => [], "UC" => [], "UL" => [], "WMST" => []}
catalog_depts = {"*DM (MD) Program" => "HSC", "*Interdisciplinary: A.S." => "AS_MULTI_COURSES", "*Interdisciplinary: Education" => "COE_MULTI_COURSES", "*Interdisciplinary: Fine Arts" => "CFA_MULTI_COURSES", "*Interdisciplinary:Engineering" => "SOE_MULTI_COURSES", "AS American Studies" => "AMST", "AS Anthropology" => "ANTH", "AS Biology" => "BIOL", "AS CHMS Program" => "CCS", "AS Economics" => "ECON", "AS Linguistics" => "LING", "African American Studies" => "AFST", "Air Force ROTC" => nil, "Anderson Schol Management ASM" => "ASM", "Art Art History" => "ARTH", "Biomedical Sci Grad Prg BSGP" => "HSC", "CE Professional Development" => nil, "Chemical Nuclear
```

```
    Engineering" => "CHNE", "Chemistry" => "CHEM", "Civil Engineering Civil Engr" => "CE"
    , "College of Nursing" => "NURS", "College of Pharmacy" => "PHRM", "Communication
    Journalism" => "CJ", "Computer Science" => "CS", "Earth & Planetary Sciences" => "EPS
    ", "Educ Leader Orgn Learning ELOL" => "ELOL", "Educational Specialties Ed Sp" => "ES
    ", "Electrical Computer Engr" => "ECE", "Emergency Medicine" => "HSC", "English" => "
    ENGL", "FCM Masters in Public Health" => "HSC", "FCM Physicians Assistand Pgm" => "
    HSC", "Foreign Languages Literatures" => "FLL", "Geography" => "GEOG", "History" => "
    HIST", "Individual Fam Comm Educ IFCE" => "IFCE", "International Programs Studies" =>
     nil, "Landscape Architecture" => nil, "Lang Literacy Sociocultural LL" => "LLSS", "
    LosAlamos Branch" => nil, "Mathematics Statistics" => "MATH", "Media Arts" => "MA", "
    Military Science & Leadership" => nil, "Music" => "MUS", "NSMS Nano Science & Micro
    Syst" => "NSMS", "Native American Studies" => "UC", "Naval Science" => nil, "Orgn
    Learning Instruct Develop" => "ELOL", "Orthopaedics Physical Therapy" => "HSC", "
    Pediatrics Occupational Ther" => nil, "Philosophy" => "PHIL", "Phys Perform Dev Phy
    Perf Dev" => "HESS", "Physical Ed (Non-Professional)" => "HESS", "Physics Astronomy"
    => "PHYC", "Political Science" => "POLS", "Provost Branch Campuses" => nil, "
    Psychology" => "PSY", "Radiology" => "RADS", "Religious Studies Prgm" => "RELG", "SOE
     Mechanical Engineering" => "ME", "SOM Clinical Departments" => "HSC", "SOM Pathology
     Medical Lab Sci" => "MEDL", "School Architecture Planning" => "SAAP", "School of Law
     Administration" => "LAW", "School of Public Admin" => "SPA", "Sociology" => "SOC", "
    Spanish Portuguese" => "SPANPORT", "Speech & Hearing Sciences" => "HSC", "Surgery
    Dental Services" => "HSC", "Teacher Education" => "TED", "Theatre & Dance" => "DANC",
     "UC Administration" => nil, "UC Advisement Center" => nil, "UC Chicano Studies
    Program" => nil, "UC Departments" => "UC", "UC Student Academic Choices" => "UC", "UC
     Water Resources Program" => "UC", "UNM Honors Program" => "HC", "Univ Lbry Deans
    Support" => nil, "Women Studies" => "WMST"}
9
10
11
12
13  file = File.open("eecs_catalog.json", "rb")
14  json_string = file.read
15  file.close
16
17  parsed_json = JSON.parse(json_string)
18
19  parsed_json.each do |pj|
20    if catalog_depts[pj["department "]]
21      course_name = pj["name"]
22      course_title = pj["title"]
23
24      #takes the highest amount of credit possible
25      course_credits = pj["credits"].split.last.to_i
26      #can be altered in the future
27
28      #read raw course description
29      course_description = pj["description "]
30
31      #load co or prereq field
32      temp_string = course_description.match(/[Pp]re[-]{0,1}\s[Oo]?[Rr]?\s?[Cc]o[-]{0,1}
            requisite[s]{0,1}:\s[^\.]+\./).to_s
33      course_co_or_prereq_string = temp_string.gsub(/[Pp]re[-]{0,1}\s[Oo]?[Rr]?\s?[Cc]o
            [-]{0,1}requisite[s]{0,1}:/, "").strip
34      course_description = course_description.gsub(/[Pp]re[-]{0,1}\s[Oo]?[Rr]?\s?[Cc]o
            [-]{0,1}requisite[s]{0,1}:\s[^\.]+\./, '').strip
35
36      #load prereq field
```

```
37      temp_string = course_description.match(/[Pp]re[-]{0,1}\s?requisite[s]{0,1}:
            [^\.]+\./).to_s
38      course_prereq_string = temp_string.gsub(/[Pp]re[-]{0,1}\s?requisite[s]{0,1}:/, '').
            strip
39      course_description = course_description.gsub(/[Pp]re[-]{0,1}\s?requisite[s]{0,1}:
            [^\.]+\./, '').strip
40
41      #load coreq field
42      temp_string = course_description.match(/[Cc]o[-]?\s?requisite[s]?: [^\.]+\./).to_s
43      course_coreq_string = temp_string.gsub(/[Cc]o[-]?\s?requisite[s]?:/, '').strip
44      course_description = course_description.gsub(/[Cc]o[-]?\s?requisite[s]?: [^\.]+\./, '
            ').strip
45
46      #load which semesters courses are offered
47      course_semester_offering = ""
48      temp_string = course_description.match(/\(([Ss]ummer|[Ff]all|[Ss]pring){0,1},{0,1}
            {0,1}([Ss]ummer|[Ff]all|[Ss]pring){0,1},{0,1} {0,1}([Ss]ummer|[Ff]all|[Ss]pring)
            {0,1},{0,1} {0,1}\)/) { |m|
49        course_semester_offering += m.to_s
50      }
51      course_semester_offering = course_semester_offering.gsub('(','').gsub(')', '')
52      course_description = course_description.gsub(/\(([Ss]ummer|[Ff]all|[Ss]pring)
            {0,1},{0,1} {0,1}([Ss]ummer|[Ff]all|[Ss]pring){0,1},{0,1} {0,1}([Ss]ummer|[Ff]all
            |[Ss]pring){0,1},{0,1} {0,1}\)/, '').strip
53
54
55
56      prereqs_array = parse_array(course_prereq_string, course_name.match(/[A-Z]{2,4}/),
            course_name)
57      coreqs_array = parse_array(course_coreq_string, course_name.match(/[A-Z]{2,4}/),
            course_name)
58      pre_or_coreqs_array = parse_array(course_co_or_prereq_string, course_name.match(/[A-Z
            ]{2,4}/), course_name)
59
60      # course_prereq_string = prereq_string(prereqs_array)
61      # course_coreq_string = prereq_string(coreqs_array)
62      # course_co_or_prereq_string = prereq_string(pre_or_coreqs_array)
63
64      # prereqs_array = prereq_prefix_eqn(prereqs_array)
65      # coreqs_array = prereq_prefix_eqn(coreqs_array)
66      # pre_or_coreqs_array = prereq_prefix_eqn(pre_or_coreqs_array)
67
68      #I DID THIS FIRST
69      course_description_string = "\t\{\n\t\tnumber: \"#{course_name}\",\n\t\ttitle: \"#{
            course_title}\",\n\t\tcredits: #{course_credits},\n\t\tdescription: \"#{
            course_description}\",\n\t\tprereq_string: \"#{course_prereq_string}\",\n\t\
            tcoreq_string: \"#{course_coreq_string}\",\n\t\tco_or_prereq_string: \"#{
            course_co_or_prereq_string}\",\n\t\tsemester_offering: \"#{
            course_semester_offering}\",\n\t\tprereq_array: #{prereqs_array.to_s},\n\t\
            tcoreq_array: #{coreqs_array.to_s},\n\t\tpre_or_coreq_array: #{
            pre_or_coreqs_array.to_s}\n\t\},\n\n"
70
71      course_array[catalog_depts[pj["department "]]] << course_description_string
72    end
73  end
74
75  #all courses are loaded
76  #make sure all courses are uniq and sorted
```

```
77  weasel_depts.each do |weasel|
78    course_array[weasel].sort!
79    course_array[weasel].uniq!
80
81    unless course_array[weasel].empty?
82      course_array[weasel].last.chomp!(",\n\n")
83    end
84  end
85
86  puts "writing out courses"
87  #write 'em out to file
88  weasel_depts.each do |weasel|
89    unless course_array[weasel].empty?
90      File.open("./courses/#{weasel}courses.rb", 'w') do |f|
91  #   File.open("../Wild-Weasel/db/data/courses/#{weasel}courses.rb", 'w') do |f|
92        f.write "courses = Course.create([\n\n"
93        course_array[weasel].each do |course|
94          f.write(course)
95        end
96        f.write("\n\n])\n\nDept.where(acronym: '#{weasel}').first.courses = Array.new(
              courses)")
97      end
98    end
99  end
```

```
1   require_relative './change_to_prefix.rb'
2
3   def reduce_parens(prereq_array)
4     infix_array = Array.new(prereq_array)
5
6     current_operator = nil
7     stack = Array.new
8     last_index = infix_array.size - 1
9
10    infix_array.each_with_index do |element, index|
11
12      if is_open_paren? element
13        stack.push index
14
15      elsif is_operator? element
16        current_operator = element
17
18      elsif is_close_paren? element
19        if (infix_array[index + 1] == current_operator and is_operand? infix_array[index +
              2])
20          infix_array[stack.pop] = nil
21          infix_array[index] = nil
22
23        elsif index == last_index && is_close_paren?(element)
24          if stack.pop == 0
25            infix_array[index] = nil
26            infix_array[0] = nil
27          end
28
29        else
30          stack.pop
31
32        end
```

```
33      end
34    end
35
36    infix_array.delete(nil)
37    infix_array.reverse!
38
39    current_operator = nil
40    stack = Array.new
41    last_index = infix_array.size - 1
42
43    infix_array.each_with_index do |element, index|
44
45      if is_close_paren? element
46        stack.push index
47
48      elsif is_operator? element
49        current_operator = element
50
51      elsif is_open_paren? element
52        if (infix_array[index + 1] == current_operator and is_operand? infix_array[index +
              2])
53          infix_array[stack.pop] = nil
54          infix_array[index] = nil
55
56        else
57          stack.pop
58
59        end
60      end
61    end
62
63    infix_array.delete(nil)
64    infix_array.reverse!
65  end
66
67  def prereq_string(prereq_array)
68    reduce_parens(prereq_array).join(" ").gsub("( ", "(").gsub(" )", ")")
69  end
70
71
72
73  def parse_array(string, current_dept, course_name)
74    string = string.gsub(/[0-9]+-[0-9]+/, '').gsub(/\s+/, ' ')
75
76    string = string.gsub('IS-M', 'ISM').gsub(/\s+/, ' ')
77
78    string = string.gsub(/\bMath\b/, 'MATH').gsub(/\s+/, ' ')
79
80    #assume commas are an "and" condition
81    string = string.gsub(',', ' and ').gsub(/\s+/, ' ')
82
83    #remove all special characters except for parenthesis
84    string = string.gsub(/[^A-Za-z0-9() ]/, ' ').gsub(/\s+/, ' ')
85
86    string = string.gsub("(", "( ").gsub(/\s+/, ' ')
87
88    string = string.gsub(")", " )").gsub(/\s+/, ' ')
89
```

```
#remove all capitalized words
string = string.gsub(/\b[A-Z]{1}[a-z]\b/, ' ').gsub(/\s+/, ' ')

#remove ACT
string = string.gsub(/\bACT\s[0-9]?/, ' ').gsub(/\s+/, ' ')

#remove ISM
string = string.gsub(/\bISM\s[0-9]?/, ' ').gsub(/\s+/, ' ')

#remove SAT
string = string.gsub(/\bSAT\s[0-9]?/, ' ').gsub(/\s+/, ' ')

#remove stray 1 or 2 digit number
string = string.gsub(/\b[0-9]{1,2}\b/, ' ').gsub(/\s+/, ' ')

string = string.gsub(/\b[Oo][Rr]\b/, '+').gsub(/\b[Aa][Nn][Dd]\b/, '*')

array = string.split

array.delete_if{ |element|
  not(element.match(/[()*+]/)) and not(element.match(/[A-Z]{2,4}/)) and not(element.
      match(/[0-9]{3}[A-Za-z]?/))
}

string = array.join(' ')
string = string.gsub("( )", '')
string = string.gsub(/\([+* ]+\)/, '')
# puts string

while string.match(/\([+* ]+([A-Z]{2,4} [0-9]{3}[A-Za-z]?) [+* ]+\)/)
  string.match(/\([+* ]+([A-Z]{2,4} [0-9]{3}[A-Za-z]?) [+* ]+\)/){ |m|
    string.gsub!(string.match(/\([+* ]+([A-Z]{2,4} [0-9]{3}[A-Za-z]?) [+* ]+\)/).to_s,
        m[1])
  }
end

array = string.split

while array.first == '+' or array.first == '*'
  array.shift
end

while array.last == '+' or array.last == '*'
  array.pop
end

final_array = Array.new
last_element = ""

array.each do |element|
  if element.match(/[A-Z]{2,4}/)
    last_element = element
  elsif element.match(/[0-9]{3}[A-Za-z]?/)
    if last_element.match(/[A-Z]{2,4}/)
        final_array << "#{last_element}_#{element}"
      else
        final_array << "#{current_dept}_#{element}"
      end
```

```
146     else
147       final_array << element
148     end
149     last_element = element
150   end
151
152
153   i = 0
154   loop_iteration = true
155   last_element_op = false
156
157   unless final_array.empty?
158     while loop_iteration
159       current_element_op = (final_array[i].match(/[+*]/) ? true : false)
160
161       if current_element_op and last_element_op
162         final_array.delete_at(i-1)
163         i = 0
164         last_element_op = false
165       else
166         last_element_op = current_element_op
167         i +=1
168       end
169
170       loop_iteration = (i < final_array.size)
171
172     end
173   end
174
175   string = final_array.join(' ')
176   string = string.gsub(/[*+]\s\(\s*\)/, ' ').gsub(/\s+/, ' ')
177   temp_array = string.split(" ")
178
179   final_array = Array.new
180   temp_array.each do |element|
181     final_array << element.gsub("_", " ")
182   end
183   return prereq_prefix_eqn(final_array, course_name)
184   # return Array.new(final_array)
185 end
```

```
1  def is_open_paren?(string)
2    string == '('
3  end
4
5  def is_close_paren?(string)
6    string == ')'
7  end
8
9  def is_paren?(string)
10   string == '(' or string == ')'
11 end
12
13 def is_operator?(string)
14   string == '+' or string == "or" or string == "Or" or string == "OR" or string == '*' or
         string == "and" or string == "And" or string == "AND"
15 end
16
```

```
17  def is_operator_or_paren?(string)
18    is_paren?(string) or is_operator?(string)
19  end
20
21  def is_operand?(string)
22    not(is_operator_or_paren?(string))
23  end
24
25  def to_symbolic_operator(string)
26    if string == '+' or string == "or" or string == "Or" or string == "OR"
27      return '+'
28    elsif string == '*' or string == "and" or string == "And" or string == "AND"
29      return '*'
30    end
31  end
32
33  def prereq_prefix_eqn(prereq_array, course_name)
34    # DEBUG
35    static_array = Array.new(prereq_array)
36    # DEBUG
37
38    operator_stack = Array.new
39    prefix_eqn = Array.new
40    current_eqn = prereq_array.reverse
41
42    current_eqn.each do |an_element|
43      if is_operator?(an_element)
44        operator_stack.push to_symbolic_operator(an_element)
45
46      elsif is_close_paren?(an_element)
47        operator_stack.push an_element
48
49      elsif is_open_paren?(an_element)
50
51        this_operator = operator_stack.pop
52
53        while !is_close_paren?(this_operator)
54          # DEBUG
55          puts course_name
56          # DEBUG
57          prefix_eqn.push this_operator
58          this_operator = operator_stack.pop
59
60        end
61
62
63      else
64        prefix_eqn.push an_element
65
66      end
67
68    end
69
70      unless operator_stack.empty?
71        prefix_eqn += operator_stack.reverse
72      end
73
74    prefix_eqn.reverse
```

```
75  end
76
77  Dir[File.dirname(__FILE__) + "/courses/**/*.rb"].each  do |file|
78    # DEBUG
79    puts file
80    # DEBUG
81    lines_of_file = []
82
83    File.open(file, 'r').each do |line|
84
85      if line.index "_array"
86        matched_array = line.match(/\[(.+)\]/)
87
88        if matched_array
89           prefix_array = prereq_prefix_eqn(matched_array[1].gsub("\"", "").split(',').each
                 {|e| e.strip!})
90
91           lines_of_file << line.gsub(/\[(.+)\]/, prefix_array.inspect)
92        else
93          lines_of_file << line
94        end
95
96      else
97        lines_of_file << line
98      end
99
100   end
101
102   File.open(file, 'w') do |f|
103     lines_of_file.each do |a_line|
104       f.write a_line
105     end
106   end
107 end
```

Appendix C

Database Seed Files

Below is a sample of Rails seed file that is able to load all of the courses offered in Electrical & Computer Engineering

```
courses = Course.create([

  {
    number: "ECE 101",
    title: "Intro to Elect & Computer Eng",
    credits: 1,
    description: "Insight into electrical engineering is gained through videos,  hands-on
          experiments, use of computer software to learn basic problem-solving skills and
         a team-oriented design project.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 131",
    title: "Programming Fundamentals",
    credits: 3,
    description: "Fundamental programming concepts, including consideration of abstract
        machine models with emphasis on the memory hierachy, basic programming constructs
        , functions, parameter passing, pointers and arrays, file I/O, bit-level
        operations and interfacing to external devices.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
```

```
27      coreq_array: [],
28      pre_or_coreq_array: []
29    },
30
31    {
32      number: "ECE 203",
33      title: "Circuit Analysis I",
34      credits: 3,
35      description: "Basic elements and sources.  Energy and power.  Ohm's law and Kirchhoff
            's laws.  Resistive networks, node and loop analysis.  Network theorems.  First-
            order and second-order circuits.  Sinusoidal sources and complex representations:
              impendance, phasors, complex power.  Three-phase circuits.",
36      prereq_string: "ECE 131 and MATH 163.",
37      coreq_string: "",
38      co_or_prereq_string: "MATH 316 and PHYC 161.",
39      semester_offering: "",
40      prereq_array: ["*", "ECE 131", "MATH 163"],
41      coreq_array: [],
42      pre_or_coreq_array: ["*", "MATH 316", "PHYC 161"]
43    },
44
45    {
46      number: "ECE 206L",
47      title: "Instrumentation",
48      credits: 2,
49      description: "Introduction to laboratory practices and the use of test equipment.
            Measurements on basic electrical components, dc and ac circuits using ohmmeters,
            voltmeters, ammeters and oscilloscopes. Circuit simulation.",
50      prereq_string: "203L and ENGL 102.",
51      coreq_string: "",
52      co_or_prereq_string: "",
53      semester_offering: "",
54      prereq_array: ["*", "ECE 203L", "ENGL 102"],
55      coreq_array: [],
56      pre_or_coreq_array: []
57    },
58
59    {
60      number: "ECE 213",
61      title: "Circuit Analysis II",
62      credits: 3,
63      description: "General transient analysis of electrical circuits. Laplace transform
            with applications to circuit analysis. State-space equations.  Fourier series
            analysis. The network function; convolution; frequency response.",
64      prereq_string: "203L and MATH 316.",
65      coreq_string: "MATH 314.",
66      co_or_prereq_string: "",
67      semester_offering: "",
68      prereq_array: ["*", "ECE 203L", "MATH 316"],
69      coreq_array: ["MATH 314"],
70      pre_or_coreq_array: []
71    },
72
73    {
74      number: "ECE 231",
75      title: "Intermediate Programming",
76      credits: 3,
```

```
77      description: "Introducton to elementary data structures, program design and computer-
            based solution of engineering problems. Topics include use of pointers, stacks,
            queues, linked lists, trees, graphs, systems and device-level programming and
            software design methodology.",
78      prereq_string: "ECE 131.",
79      coreq_string: "",
80      co_or_prereq_string: "",
81      semester_offering: "",
82      prereq_array: ["ECE 131"],
83      coreq_array: [],
84      pre_or_coreq_array: []
85    },
86
87    {
88      number: "ECE 238L",
89      title: "Computer Logic Design",
90      credits: 4,
91      description: "Binary number systems. Boolean algebra. Combinational, sequential and
            register transfer logic. VHDL. Arithmetic/logic unit. Memories, computer
            organization. Input-output. Microprocessors.",
92      prereq_string: "ECE 131.",
93      coreq_string: "",
94      co_or_prereq_string: "",
95      semester_offering: "",
96      prereq_array: ["ECE 131"],
97      coreq_array: [],
98      pre_or_coreq_array: []
99    },
100
101   {
102     number: "ECE 314",
103     title: "Signals and Systems",
104     credits: 3,
105     description: "Continuous and discrete time signals and systems; time and frequency
            domain analysis of LTI systems, Fourier series and transforms, discrete time
            Fourier series/transform sampling theorem, block diagrams, modulation/
            demodulation, filters.",
106     prereq_string: "213 and MATH 264.",
107     coreq_string: "",
108     co_or_prereq_string: "",
109     semester_offering: "",
110     prereq_array: ["*", "ECE 213", "MATH 264"],
111     coreq_array: [],
112     pre_or_coreq_array: []
113   },
114
115   {
116     number: "ECE 321L",
117     title: "Electronics I",
118     credits: 4,
119     description: "Introduction to diodes, bipolar and field-effect transistors. Analysis
            and design of digital circuits, gates, flip-flops and memory circuits. Circuits
            employing operational amplifiers. Analog to digital and digital to analog
            converters.",
120     prereq_string: "213.",
121     coreq_string: "",
122     co_or_prereq_string: "",
123     semester_offering: "",
```

```
    prereq_array: ["ECE 213"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 322L",
    title: "Electronics II",
    credits: 4,
    description: "Analysis, design, and characterization of linear circuits including
        operational amplifiers. Design of biasing and reference circuits, multistage
        amplifiers, and feedback circuits.",
    prereq_string: "321L.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 321L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 330",
    title: "Software Design",
    credits: 3,
    description: "Design of software systems using modern modeling techniques.
        Relationship between software design and process, with emphasis on UML and its
        interface application code.  Exposure to design patterns, software frameworks,
        and software architectural paradigms.",
    prereq_string: "231.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 231"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 331",
    title: "Data Structures & Algorithms",
    credits: 3,
    description: "An introduction to data structures and algorithms. Topics include
        asymptotic notation recurrence relations, sorting, hash tables, basic priority
        queues, balanced search trees and basic graph representation and search.",
    prereq_string: "231 and MATH 327.",
    coreq_string: "340.",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 231", "MATH 327"],
    coreq_array: ["ECE 340"],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 335",
    title: "Integrated Software Systems",
    credits: 3,
```

```
    description: "Course considers design principles, implementation issues, and
        performance evaluation of various software paradigms in an integrated computing
        environment. Topics include performance measurement and evaluation, program
        optimization for the underlying architeccture, integration and security for large
        -scale software systems.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 337",
    title: "Computer Arch & Organization",
    credits: 3,
    description: "Survey of various levels of computer architecture and design;
        microprogramming and processor architecture, assembly language programming,
        operating system concepts and input/output via the operating system. Three
        lectures, 1 hr. lab.",
    prereq_string: "231 and 238L.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "Spring",
    prereq_array: ["*", "ECE 231", "ECE 238L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 338",
    title: "Intermediate Logic Design",
    credits: 3,
    description: "Advanced combinational circuits; XOR and transmission gates; computer-
        based optimization methods; RTL and HDL; introduction to computer aided design;
        advanced sequential machines; asynchronous sequential machines; timing issues;
        memory and memory interfacing; programmable logic devices; and VLSI concepts.",
    prereq_string: "238L.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 238L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 340",
    title: "Probabilistic Methods in Eng",
    credits: 3,
    description: "Introduction to probability, random variables, random processes,
        probability distribution/density functions, expectation correlation, power
        spectrum, WSS processes, confidence internals, transmission through LIT systems,
        applications of probability.",
    prereq_string: "314 and MATH 314.",
    coreq_string: "",
```

```
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 314", "MATH 314"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 341",
    title: "Communication Systems",
    credits: 3,
    description: "Amplitude/frequency modulation, pulse position/amplitude modulation,
        probabilistic noise model, AWGN, Rice representation, figure of merit, phase
        locked loops, digital modulation, introduction to multiple access systems.",
    prereq_string: "314 and 340.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 314", "ECE 340"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 344L",
    title: "Microprocessors",
    credits: 4,
    description: "Computers and Microprocessors: architecture, assembly language
        programming, input/output and applications.  Three lectures, 3 hrs. lab.",
    prereq_string: "206L and 238L and 321L.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "Fall, Spring",
    prereq_array: ["*", "*", "ECE 206L", "ECE 238L", "ECE 321L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 345",
    title: "Intro to Control Systems",
    credits: 3,
    description: "Introduction to the feedback control problem. Plant modeling, transfer
        function and state-space descriptions. Stability criteria. Nyquist and root-locus
         design. Introduction to analytical design. Z-transforms and digital control.
        Laboratory design project.",
    prereq_string: "314.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 314"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 360",
    title: "Electromagnetic Fields & Waves",
```

```
    credits: 3,
    description: "Maxwell s equations, plane wave propagation, waveguides and
        transmission lines, transient pulse propagation and elementary dipole antenna.",
    prereq_string: "213 and PHYC 161 and MATH 264.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "*", "ECE 213", "PHYC 161", "MATH 264"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 371",
    title: "Materials & Devices",
    credits: 4,
    description: "Introduction to quantum mechanics, crystal structures, insulators,
        metals, and semiconductor material properties, bipolar, field effect and light
        emitting devices.",
    prereq_string: "PHYC 262.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["PHYC 262"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 381",
    title: " Intro to Power Systems",
    credits: 3,
    description: "Provides in-depth look at various elements of power systems including
        power generation, transformer action, transmission line modeling, symmetrical
        components, pf correction, real/quadrature power calculations, load flow analysis
         and economic considerations in operating systems.",
    prereq_string: "213.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 213"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 413",
    title: "Intro to Ray Graphics",
    credits: 3,
    description: "Topics include ray-geometry intersections, viewing, lenses, local/
        global illumination, procedural textures/,models, spline curves and surfaces,
        statistical integration for realistic image synthesis.  Students will write a
        raytracing renderer from scratch, exploring high performance implementations and
        realistic rendering.",
    prereq_string: "331 or CS 361L.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
```

```
    prereq_array: ["+", "ECE 331", "CS 361L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 419",
    title: "Senior Design I",
    credits: 3,
    description: "Design methodology and development of professional project-oriented
        skills including communication, team management, economics, and engineering
        ethics. Working in teams, a proposal for a large design is prepared in response
        to an industrial or in-house sponsor. Restriction: ECE major, senior standing.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 420",
    title: "Senior Design II",
    credits: 3,
    description: "Continuation of 419. Students work in assigned teams to implement
        proposal developed in 419. Prototypes are built and tested to sponsor
        specifications, and oral and written reports made to the project sponsor.",
    prereq_string: "419.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 419"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 421",
    title: "Analog Electronics",
    credits: 3,
    description: "Design of advanced analog electronic circuits. BJT and MOSFET
        operational amplifiers, current mirrors and output stages. Frequency response and
         compensation. Noise. A/D and D/A converters.",
    prereq_string: "322L.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 322L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 424",
    title: "Digital VLSI Design",
    credits: 3,
```

```
    description: "CMOS logic gates and circuits, transistor implementations, applications
        to sequential circuits, VLSI data path and controller design, VLSI routing
        issues and architectures, RTL and VLSI impacts and applications to microprocessor
        design.",
    prereq_string: "321L and 338.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 321L", "ECE 338"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 432",
    title: "Intro to Parallel Processing",
    credits: 3,
    description: "(Also offered as CS 442.) Machine taxonomy and introduction to parallel
        programming. Performance issues, speed-up and efficiency. Interconnection
        networks and embeddings. Parallel programming issues and models: control parallel
        , data parallel and data flow. Programming assignments on massively parallel
        machines.",
    prereq_string: "(331 or CS 351L) and (337 or CS 341L).",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "+", "ECE 331", "CS 351L", "+", "ECE 337", "CS 341L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 435",
    title: "Software Engineering",
    credits: 3,
    description: "Management and technical issues including business conduct and ethics
        related to the design of large engineering projects. Student teams will address
        the design, specification, implementation, testing and documentation of a large
        hardware/software project.",
    prereq_string: "331 and 335.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 331", "ECE 335"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 437",
    title: "Computer Operating Systems",
    credits: 3,
    description: "(Also offered as CS 481.) Fundamental principles of modern operating
        systems design, with emphasis on concurrency and resource management. Topics
        include processes, interprocess communication, semaphores, monitors, message
        passing, input/output device, deadlocks memory management, files system design.",
    prereq_string: "(330 and 337) or CS 341L.",
    coreq_string: "",
```

```
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["+", "*", "ECE 330", "ECE 337", "CS 341L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 438",
    title: "Design of Computers",
    credits: 3,
    description: "Computer architecture; design and implementation at HDL level; ALU,
        exception handling and interrupts; addressing; memory; speed issues; pipelining;
        microprogramming; introduction to distributed and parallel processing; buses; bus
         protocols and bus masters. CAD project to include written and oral presentations
        .",
    prereq_string: "337 and 338 and 344L.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "*", "ECE 337", "ECE 338", "ECE 344L"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 439",
    title: "Intr Digital Signal Processing",
    credits: 3,
    description: "Bilateral Z transforms, region of convergence, review of sampling
        theorem, aliasing, the discrete Fourier transform and properties, analysis/design
         of FIR/IIR filters, FFT algorithms spectral analysis using FFT.",
    prereq_string: "314.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 314"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 440",
    title: "Computer Networks",
    credits: 3,
    description: "(Also offered as CS 485.) Theoretical and practical study of computer
        networks, including network structures and architectures. Principles of digital
        communications systems. Network topologies, protocols and services. TCP/IP
        protocol suite. Point-to-point networks; broadcast networks; local area networks;
         routing, error and flow control techniques.",
    prereq_string: "330 and 337.",
    coreq_string: "340.",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 330", "ECE 337"],
    coreq_array: ["ECE 340"],
    pre_or_coreq_array: []
  },
```

```
464
465   {
466     number: "ECE 441",
467     title: "Communication Systems",
468     credits: 3,
469     description: "Amplitude/frequency modulation, pulse position/amplitude modulation,
          probabilistic noise model, AWGN, Rice representation, figure of merit, phase
          locked loops, digital modulation, introduction to multiple access systems.",
470     prereq_string: "314 and 340.",
471     coreq_string: "",
472     co_or_prereq_string: "",
473     semester_offering: "",
474     prereq_array: ["*", "ECE 314", "ECE 340"],
475     coreq_array: [],
476     pre_or_coreq_array: []
477   },
478
479   {
480     number: "ECE 442",
481     title: "Wireless Communication",
482     credits: 3,
483     description: "The course is an introduction to cellular telephone systems and
          wireless networks, drawing upon a diversity of electrical engineering areas.
          Topics include cellular concepts, radio propagation, modulation methods and
          multiple access techniques.",
484     prereq_string: "314 and 360.",
485     coreq_string: "",
486     co_or_prereq_string: "",
487     semester_offering: "",
488     prereq_array: ["*", "ECE 314", "ECE 360"],
489     coreq_array: [],
490     pre_or_coreq_array: []
491   },
492
493   {
494     number: "ECE 443",
495     title: "Hardware Design with VHDL",
496     credits: 3,
497     description: "The VHDL hardware description language is used for description of
          digital systems at several levels of complexity, from the system level to the
          gate level. Descriptions provide a mechanism for documentation, for simulation
          and for synthesis.",
498     prereq_string: "338.",
499     coreq_string: "",
500     co_or_prereq_string: "",
501     semester_offering: "",
502     prereq_array: ["ECE 338"],
503     coreq_array: [],
504     pre_or_coreq_array: []
505   },
506
507   {
508     number: "ECE 446",
509     title: "Feedback Control Systems",
510     credits: 3,
511     description: "Modeling of continuous and sampled-data control systems. State-space
          representation. Sensitivity, stability and optimization of control systems.
          Design of compensators in the frequency and time domains. Phase-plane, describing
```

```
        function design for non-linear systems, and laboratory design project.",
    prereq_string: "345.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 345"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 448",
    title: "Fuzzy Logic with Applications",
    credits: 3,
    description: "(Also offered as CE 448.) Theory of fuzzy sets; foundations of fuzzy
        logic. Fuzzy logic is shown to contain evidence, possibility and probability
        logics; course emphasizes engineering applications; control, pattern recognition,
         damage assessment, decisions; hardware/software demonstrations.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 456",
    title: "Entrepreneurial Engineering",
    credits: 3,
    description: "Review and application of necessary elements for successfully launching
         technical businesses; focuses upon technology, manufacturing, management,
        marketing, legal and financial aspects. Students work in groups developing
        elements of new businesses and producing business plans. Restriction: senior
        standing.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 460",
    title: "Microwave Engineering",
    credits: 3,
    description: "This lecture/laboratory course provides essential fundamentals for rf,
        wireless and microwave engineering. Topics include: wave propagation in cables,
        waveguides and free space; impedance matching, standing wave ratios, Z- and S-
        parameters.",
    prereq_string: "360.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 360"],
```

```
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 463",
    title: "Advanced Optics I",
    credits: 3,
    description: "(Also offered as PHYC 463.) Electromagnetic theory of geometrical
        optics, Gaussian ray tracing and matrix methods, finite ray tracing, aberrations,
         interference and diffraction.",
    prereq_string: "PHYC 302.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["PHYC 302"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 464",
    title: "Laser Physics I",
    credits: 3,
    description: "(Also offered as PHYC 464.) Resonator optics. Rate equations;
        spontaneous and stimuated emission; gas, semiconductor and solid state lasers,
        pulsed and mode-locked laser techniques.",
    prereq_string: "360 or PHYC 406.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["+", "ECE 360", "PHYC 406"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 469",
    title: "Antennas for Wireless Comm",
    credits: 3,
    description: "Aspects of antenna theory and design; radiation from dipoles, loops,
        apertures, microstrip antennas and antenna arrays.",
    prereq_string: "360.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 360"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 471",
    title: "Materials & Devices II",
    credits: 3,
    description: "An intermediate study of semiconductor materials, energy band structure
        , p-n junctions, ideal and non-ideal effects in field effect and bipolar
        transistors.",
```

```
    prereq_string: "360 and 371.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 360", "ECE 371"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 474L",
    title: "Microelectronics Processing I",
    credits: 3,
    description: "(Also offered as NSMS 574L.) Materials science of semiconductors,
        microelectronics technologies, device/circuit fabrication, parasitics and
        packaging. Lab project features small group design/fabrication/testing of MOS
        circuits.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 475",
    title: "Optoelectronics",
    credits: 3,
    description: "Basic electro-optics and opto-electronics, with engineering
        applications. Interaction of light with matter. Introduction to optics of
        dielectrics, metals and crystals. Introductory descriptions of electro-optic,
        acousto-optic and magneto-optic effects and related devices. Light sources,
        displays and detectors. Elementary theory and applications of lasers, optical
        waveguides and fibers.",
    prereq_string: "371.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 371"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 482",
    title: "Electric Drives & Transformers",
    credits: 3,
    description: "Electromagnetic theory and mechanical considerations are employed to
        develop models for and understanding of Transformers, Induction Machines and
        Synchronous Machines. Additionally, DC Machines are discussed.",
    prereq_string: "203 and 213.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 203", "ECE 213"],
    coreq_array: [],
```

```
    pre_or_coreq_array: []
  },

  {
    number: "ECE 483",
    title: "Power Electronics",
    credits: 3,
    description: "Introduces modern power conversion techniques at a lower level, dealing
        with basic structures of power converters and techniques of analyzing converter
        circuits. Students learn to analyze and design suitable circuits and subsystems
        for practical applications. Prerquisite: 321L and 371 and 381.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 484",
    title: "Photovoltaics",
    credits: 3,
    description: "Technical concepts of photovoltaics. Solar cell device level operation,
        packaging, manufacturing, designing phovoltaic system for stand-alone or grid-
        tied operation, some business-case analysis and some real-life scenarios of
        applicability of these solutions.",
    prereq_string: "381 and MATH 121.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 381", "MATH 121"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 488",
    title: "Future Energy Systems",
    credits: 3,
    description: "A detailed study of current and emerging power and energy systems and
        technologies. Including renewable energies, storage, Smart Grid concepts,
        security for power infrastructure. Software modeling of power systems and grids."
        ,
    prereq_string: "381 and 482 and 483 and 484.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "*", "*", "ECE 381", "ECE 482", "ECE 483", "ECE 484"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 490",
    title: "Internship",
    credits: 3,
```

```
    description: "Professional practice under the guidance of a practicing engineer.
        Assignments include design or analysis of systems or hardware, or computer
        programming. A preliminary proposal and periodic reports are required. The
        engineer evaluates student s work; a faculty monitor assigns grade.  Restriction:
         ECE major, junior standing. (12 hours/week) (24 hours/week in summer session).
        Offered on a CR/NC basis only.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 491",
    title: "Undergrad Problems",
    credits: 6,
    description: "Registration for more than 3 hours requires permission of department
        chairperson.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 493",
    title: "Honors Seminar",
    credits: 3,
    description: "A special seminar open only to honors students. Registration requires
        permission of department chairperson.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 494",
    title: "Honors Individual Study",
    credits: 6,
    description: "Open only to honors students. Registration requires permission of the
        department chairperson and of the supervising professor.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
```

```
  },

  {
    number: "ECE 495",
    title: "Special Topics",
    credits: 4,
    description: "Restriction: ECE major, senior standing.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 500",
    title: "Theory of Linear Systems",
    credits: 3,
    description: "State space representation of dynamical systems. Analysis and design of
        linear models in control systems and signal processing. Continuous, discrete and
        sampled representations. This course is fundamental for students in the system
       areas.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 506",
    title: "Optimization Theory",
    credits: 3,
    description: "Introduction to the topic of optimization by the computer. Linear and
        nonlinear programming. The simplex method, Karmakar method, gradient, conjugate
        gradient and quasi-Newton methods, Fibonacci/Golden search, Quadratic and Cubic
        fitting methods, Penalty and Barrier methods.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 510",
    title: "Medical Imaging",
    credits: 3,
    description: "This course will introduce the student to medical imaging modalities (e
        .g. MRI, Nuclear Imagine, Ultrasound) with an emphasis on a signals and systems
        approach.  Topics include hardware, signal formation, image reconstruction, and
        application.",
```

```
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 511",
    title: "fMRI Analysis Methods",
    credits: 3,
    description: "This course will be an introduction to signal and image processing
        methods for functional magnetic resonance imaging (fMRI) of the brain.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 512",
    title: "Advanced Image Synthesis",
    credits: 3,
    description: "Course covers image synthesis techniques from perspective of high-end
        scanline rendering including physically-based rendering algorithms.  Topics:
        radiometry, stachastic ray tracing, variance reduction, photon mapping,
        reflection models, participating media, advanced algorithms for light transport."
        ,
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 513",
    title: "Real-time Graphics",
    credits: 3,
    description: "Course covers advanced algorithms in real-time rendering and graphics
        hardware, bringing students up to speed with cutting edge real-time graphics.
        Topics: advanced GPU algorithms for graphics and non-graphics applications.  Term
         project required.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },
```

```
856
857   {
858     number: "ECE 514",
859     title: "Nonlinear Control",
860     credits: 3,
861     description: "Linearization of nonlinear systems. Phase-plane analysis. Lyapunov
          stability analysis. Hyperstability and Popov stability criterion. Adaptive
          control systems. Adaptive estimation. Stability of adaptive control systems,
          backstepping and nonlinear designs.",
862     prereq_string: "500.",
863     coreq_string: "",
864     co_or_prereq_string: "",
865     semester_offering: "",
866     prereq_array: ["ECE 500"],
867     coreq_array: [],
868     pre_or_coreq_array: []
869   },
870
871   {
872     number: "ECE 516",
873     title: "Computer Vision",
874     credits: 3,
875     description: "(Also offered as CS 532.) Theory and practice of feature extraction,
          including edge, texture and shape measures. Picture segmentation; relaxation.
          Data structures for picture description. Matching and searching as models of
          association and knowledge learning. Formal models of picture languages.",
876     prereq_string: "",
877     coreq_string: "",
878     co_or_prereq_string: "",
879     semester_offering: "",
880     prereq_array: [],
881     coreq_array: [],
882     pre_or_coreq_array: []
883   },
884
885   {
886     number: "ECE 517",
887     title: "Pattern Recognition",
888     credits: 3,
889     description: "(Also offered as CS 531) Decision functions and dichotomization;
          prototype classification and clustering; statistical classification and Bayes
          theory; trainable deterministic and statistical classifiers. Feature
          transformations and selection.",
890     prereq_string: "",
891     coreq_string: "",
892     co_or_prereq_string: "",
893     semester_offering: "",
894     prereq_array: [],
895     coreq_array: [],
896     pre_or_coreq_array: []
897   },
898
899   {
900     number: "ECE 518",
901     title: "Synthesis of Nanostructures",
902     credits: 3,
903     description: "(Also offered as CHNE, NSM 518.) Underlying physical and chemical
          principles (optics, organic and inorganic chemistry, colloid chemistry, surface
```

```
        and materials science) for nanostructure formation using 'top-down' lithography (
        patterned optical exposure of photosensitive materials) and 'bottom-up' self-
        assembly. Labs will synthesize samples.",
    prereq_string: "510.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "Spring",
    prereq_array: ["ECE 510"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 519",
    title: "MEMS Transducer Devices & Tech",
    credits: 3,
    description: "(Also offered as NSMS 519, ME 419, ME 519.) Lectures and laboratory
        projects on physical theory, design, analysis, fabrication, and characterization
        of micro and nanosystems. Special attention given to scaling effects involved
        with operation of devices at nano and microscale.Restriction: senior standing.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 520",
    title: "VLSI Design",
    credits: 3,
    description: "Advanced topics include: lC technologies, CAD tools, gate arrays,
        standard cells and full custom designs. Design of memories, PLA, I/0 and random
        logic circuit. Design for testability.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 522",
    title: " Hrdwr Sftwr Codesign FPGA's",
    credits: 3,
    description: "This course provides an introduction to the design of electronic
        systems that incorporate both hardware and software components.",
    prereq_string: "433.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 433"],
    coreq_array: [],
    pre_or_coreq_array: []
```

```
  },

  {
    number: "ECE 523",
    title: "Analog Electronics",
    credits: 3,
    description: "Design of advanced analog electronics circuits. BJT and MOSFET
        operational amplifiers, current mirrors and output stages. Frequency response and
         compensation. Noise. A/D and D/A converters.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 524",
    title: "Collaborative Interdisc Teach",
    credits: 3,
    description: "(Also offered as ANTH 624, BIOL 524, CS 524, STAT 524) Course designed
        to develop the methods content and assessment of effective interdisciplinary
        biological courses; Students will develop and teach an undergraduate
        interdisciplinary topics course. Topics vary. Restriction: permission of
        instructor.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 525",
    title: "Microelectronics Test Eng",
    credits: 3,
    description: "This course provides an introduction to hardware security and trust
        primitives and their application to secure and trustworthy hardware systems.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 528",
    title: "Embedded Systems Architecture",
    credits: 3,
    description: "Development and analysis of techniques and algorithms for use in
        embedded processor systems. Application of tools implementing solutions to
        control and data applications involving standard processing paradigms and
```

```
        programmable logic systems.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 533",
    title: "Digital Image Processing",
    credits: 3,
    description: "Fundamentals of 2D signals and systems. Introduction to
        multidimensional signal processing. Applications in digital image processing.
        Image formation, representation and display. Linear and nonlinear operators in
        multiple dimensions. Orthogonal transforms representation and display. Image
        analysis, enhancement, restoration and coding. Students will carry out image
        processing projects.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 534",
    title: "Plasma Physics I",
    credits: 3,
    description: "(Also offered as ASTR, PHYC, CHNE 534.) Plasma parameters, adiabatic
        invariants, orbit theory, plasma oscillations, hydromagnetic waves, plasma
        transport, stability, kinetic theory, nonlinear effects, applications.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 536",
    title: "Computer System Software",
    credits: 3,
    description: "Course considers design principles, implementation issues and
        performance evaluation of system software in advanced computing environments.
        Topics include resource allocation and scheduling, information service provider
        and manipulation, multithreading and concurrency, security for parallel and
        distributed systems.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
```

```
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 537",
    title: "Foundations of Computing",
    credits: 3,
    description: "Computational aspects of engineering problems. Topics include machine
        models and computability, classification and performance analysis of algorithms,
        advanced data structures, approximation algorithms, introduction to complexity
        theory and complexity classes.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 538",
    title: "Advanced Computer Architecture",
    credits: 3,
    description: "Course provides an in-depth analysis of computer architecture
        techniques. Topics include high speed computing techniques, memory systems,
        pipelining, vector machines, parallel processing, multiprocessor systems, high-
        level language machines and data flow computers.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 539",
    title: "Digital Signal Processing I",
    credits: 3,
    description: "Hilbert spaces, orthogonal basis, generalized sampling theorem,
        multirate systems, filterbanks, quantization, structures for LTI systems, finite
        word-length effects, linear prediction, min/max phase systems, multiresolution
        signal analysis.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 540",
```

```
    title: "Advanced Networking",
    credits: 3,
    description: "Research, design and implementation of high-performance computer
        networks and distributed systems. High speed networking technologies, multimedia
        networks, enterprise network security and management, client/server database
        applications, mobile communications and state-of-the-art internetworking
        solutions.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 541",
    title: "Probab and Stochastic Process",
    credits: 3,
    description: "Axiomatic probability theory, projection theorem for Hilbert spaces,
        conditioned expectations, modes of stochastic convergence, Markov chains, mean-
        square calculus, Wiener filtering, optimal signal estimation, prediction
        stationarity, ergodicity, transmission through linear and nonlinear systems,
        sampling.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 542",
    title: "Digital Communication Theory",
    credits: 3,
    description: "Elements of information theory and source coding, digital modulation
        techniques, signal space representation, optimal receivers for coherent/non-
        coherent detection in AWGN channels, error probability bounds, channel capacity,
        elements of block and convolutional coding, fading, equalization signal design.",
    prereq_string: "541.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 541"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 546",
    title: "Multivariable Control Theory",
    credits: 3,
    description: "Hermite, Smith and Smith-McMillan canonic forms for polynomial and
        rational matrices. Coprime matrix-fraction representations for rational matrices.
         Bezout identity. Poles and zeros for multivariable systems. Matrix-fraction
```

```
        approach to feedback system design. Optimal linear-quadratic-Gaussian (LQG)
        control. Multivariable Nyquist stability criteria.",
    prereq_string: "500.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 500"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 547",
    title: "Neural Networks",
    credits: 3,
    description: "(Also offered as CS 547.) A study of biological and artificial neuron
        models, basic neural architectures and parallel and distributed processing.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 548",
    title: "Fuzzy Logic with Applications",
    credits: 3,
    description: "(Also offered as CE 548.) Theory of fuzzy sets; foundations of fuzzy
        logic. Fuzzy logic is shown to contain evidence, possibility and probability
        logics; course emphasizes engineering applications; control, pattern recognition,
         damage assessment, decisions; hardware/software demonstrations.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 549",
    title: "Inform Thry&Coding",
    credits: 3,
    description: "An introduction to information theory. Fundamental concepts such as
        entropy, mutual information, and the asymptotic equipartition property are
        introduced. Additional topics include data compression, communication over noisy
        channels, algorithmic information theory, and applications.",
    prereq_string: "340 or equivalent.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 340"],
    coreq_array: [],
    pre_or_coreq_array: []
```

```
  },

  {
    number: "ECE 550",
    title: "Soc & Eth Iss Nanotechnology",
    credits: 3,
    description: "(Also offered as CHNE, NSMS 550.) In this course, students will examine
        issues arising from this emerging technology, includ-ing those of privacy,
       health and safety, the environment, public perception and human enhancement.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 551",
    title: "Problems",
    credits: 6,
    description: "",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 553L",
    title: "Exp Techniques Plasma Physics",
    credits: 3,
    description: "Theory and practice of plasma generation and diagnostics, coordinated
       lectures and experiments, emphasis on simple methods of plasma production and
       selection of appropriate diagnostic techniques, applications to plasma processing
        and fusion.",
    prereq_string: "534.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 534"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 554",
    title: "Advanced Optics II",
    credits: 3,
    description: "(Also offered as PHYC 554.) Diffractions theory, coherence theory,
       coherent objects, and incoherent imaging, and polarization.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
```

```
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 555",
    title: "Foundations of Engineering EM",
    credits: 3,
    description: "Mathematical foundations for engineering electromagnetics: linear
        analysis and method of moments, complex analysis and Kramers-Kronig relations,
        Green's functions, spectral representation method and electromagnetic sources.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 556",
    title: "Entrepreneurial Engineering",
    credits: 3,
    description: "(Also offered as ME 556.) Review and application of necessary elements
        for successfully launching technical businesses; focuses upon technology,
        manufacturing, management, marketing, legal and financial aspects. Students work
        in groups developing elements of new businesses and producing business plans.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 557",
    title: "Pulsed Pwr & Char Partic Accel",
    credits: 3,
    description: "Principles of pulsed power circuits, components, systems and their
        relationship to charged particle acceleration and transport. Energy storage,
        voltage multiplication, pulse shaping, insulation and breakdown and switching.
        Single particle dynamics and accelerator configurations.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 558",
```

```
    title: "Charged Partical Beams",
    credits: 3,
    description: "(Also offered as CHNE 546.) Overview of physics of particle beams and
        applications at high-current and high-energy. Topics include review of collective
         physics, beam emittance, space-charge forces, transport at high power levels,
        and application to high power microwave generation.",
    prereq_string: "557 and CHNE 545.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["*", "ECE 557", "CHNE 545"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 559",
    title: "Intern: Optical Science & Eng",
    credits: 3,
    description: "(Also offered as PHYC 559.) Students do research and/or development
        work at a participating industry or government laboratory in any area of optical
        science and engineering.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 560",
    title: "Microwave Engineering",
    credits: 3,
    description: "This lecture/laboratory course provides essential fundamentals for rf,
        wireless and microwave engineering. Topics include: wave propagation in cables,
        waveguides and free space; impedance matching, standing wave ratios, Z- and S-
        parameters.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 561",
    title: "Electrodynamics",
    credits: 3,
    description: "Maxwell's equations, electromagnetic interaction with materials, the
        wave equation, plane wave propagation, wave reflection and transmission, vector
        potentials and radiation equations, electromagnetic field theorems, wave
        propagation in anisotropic media and metamaterials, period structures, dielectric
         slab waveguides.",
    prereq_string: "555.",
```

```
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 555"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 562",
    title: "Electronics RF Design",
    credits: 3,
    description: "Course will cover rf design techniques using transmission lines, strip
        lines and solid state devices. It will include the design of filters and matching
         elements required for realizable high frequency design. Amplifiers, oscillators
        and phase lock loops are covered from a rf perspective.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 563",
    title: "Comp Methods Electromagnetics",
    credits: 3,
    description: "Computational techniques for partial differential and integral
        equations: finite-difference, finite-element, method of moments. Applications
        include transmission lines, resonators, waveguides, integrated circuits, solid-
        state device modeling, electromagnetic scattering and antennas.",
    prereq_string: "561.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 561"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 564",
    title: "Guided Wave Optics",
    credits: 3,
    description: "Optical propagation in free space, colored dielectrics, metals,
        semiconductors, crystals, graded index media. Radiation and guided modes in
        complex structures. Input and output coupling, cross-coupling mode conversion.
        Directional couplers, modulators, sources and detectors.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },
```

```
[

  {
    number: "ECE 565",
    title: "Optical Comm Components Subsys",
    credits: 3,
    description: "Optical waveguides, optical fiber attenuation and dispersion, power
        launching and coupling of light, mechanical and fiber lifetime issues,
        photoreceivers, digital on-off keying, modulation methods, SNR and BER, QAM and M
        -QAM, modulation methods, SNR, and BER, intersymbol interference (impact on SNR),
         clock and data recovery issues, point-to-point digital links, optical amplifiers
         theory and design (SOA, EDFA, and SRA), simple WDM system concepts, WDM
        components.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 566",
    title: "Advanced Optical Networks",
    credits: 3,
    description: "External modulators WDM system design, other multiple access techniques
         design issues, analog transmission systems nonlinear processes in optical fibers
         and their impact on system performance, optical networks, photonic packet
        switching, coherent lightwave systems, basic principles for homodyne and
        heterodyne detection, noise reduction, relevant digital modulation formats: PSK,
        ASK, FSK, DPSK. Practical implementation, performance of synchronous and
        asynchronous heterodyne systems, phase noise, polarization mismatch.",
    prereq_string: "565.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 565"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 569",
    title: "Antennas for Wireless Comm",
    credits: 3,
    description: "Aspects of antenna theory and design; radiation from dipoles, loops,
        apertures, microstrip antennas and antenna arrays.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 570",
```

```
    title: "Semicon Materials & Devices",
    credits: 3,
    description: "Theory and operation of optoelectronic semiconductor devices;
        semiconductor alloys, epitaxial growth, relevant semiconductor physics (
        recombination processes, heterojunctions, noise, impact ionization), analysis of
        the theory and practice of important OE semiconductor devices (LEDs, Lasers,
        Photodetectors, Solar Cells).",
    prereq_string: "471 or 572.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["+", "ECE 471", "ECE 572"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 572",
    title: "Physics of Semiconductors",
    credits: 3,
    description: "(Also offered as NSMS 572.) Crystal properties, symmetry and
        imperfections. Energy bands, electron dynamics, effective mass tensor, concept
        and properties of holes. Equilibrium distributions, density of states, Fermi
        energy and transport proper-ties including Boltzmann s equation. Continuity
        equation, diffusion and drift of carriers.",
    prereq_string: "471.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 471"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 574L",
    title: "Microelectronics Processing I",
    credits: 3,
    description: "(Also offered as NSMS 574L.) Materials science of semiconductors,
        microelectronics technologies, device/circuit fabrication, parasitics and
        packaging. Lab project features small group design/fabrication/testing of MOS
        circuits.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 576",
    title: "Modern VLSI Devices",
    credits: 3,
    description: "Review of the evolution of VLSI technology and basic device physics.
        Detailed analysis of MOSFET devices, CMOS device design including device scaling
        concepts.",
```

```
    prereq_string: "471 or 572.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["+", "ECE 471", "ECE 572"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 577",
    title: "Fundmts of Semic Lasers & LEDs",
    credits: 3,
    description: "Carrier generation and recombination, photon generation and loss in
        laser cavities, density of optical modes and blackbody radiation, radiative and
        non-radiative processes, optical gain, spontaneous and stimulated emission, Fermi
         s golden rule, gain and current relations, characterizing real diode lasers,
        dynamic effects, rate equation; small signal and large signal analysis, radiative
         intensity noise and linewidth.",
    prereq_string: "572.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 572"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 578",
    title: "Advanced Semiconductor Lasers",
    credits: 3,
    description: "Scattering matrix theory, S and T matrices, gratings, DBR and DFB
        lasers, perturbation and coupled-mode theory, photonic integrated circuits,
        tunable lasers, directional couplers.",
    prereq_string: "577.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 577"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 580",
    title: "Advanced Plasma Physics",
    credits: 3,
    description: "(Also offered as PHYC 580, CHNE 580.)",
    prereq_string: "534 or PHYC 534.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["+", "ECE 534", "PHYC 534"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

```

```
{
  number: "ECE 581",
  title: "Coll Nanocrystal Biomed Appl",
  credits: 3,
  description: "Intended for students planning careers combining engineering, materials
       science, and biomedical sciences. Covers synthesis, nanocrystals
      characterization, biofunctionalization, biomedical nanosensors, FRET-based
      nanosensing, molecular-level sensing/imaging, and applications in cell biology,
      cancer diagnostics and therapy, neuroscience, and drug delivery.",
  prereq_string: "",
  coreq_string: "",
  co_or_prereq_string: "",
  semester_offering: "",
  prereq_array: [],
  coreq_array: [],
  pre_or_coreq_array: []
},

{
  number: "ECE 582",
  title: "Electric Drives & Transformers",
  credits: 3,
  description: "Electromagnetic theory and mechanical considerations are employed to
      develop models for and understanding of Transformers, Induction Machines and
      Synchronous Machines. Additionally, DC Machines are discussed.",
  prereq_string: "",
  coreq_string: "",
  co_or_prereq_string: "",
  semester_offering: "",
  prereq_array: [],
  coreq_array: [],
  pre_or_coreq_array: []
},

{
  number: "ECE 583",
  title: "Power Electronics",
  credits: 3,
  description: "Introduces modern power conversion techniques at a lower level, dealing
       with basic structures of power converters and techniques of analyzing converter
      circuits. Students learn to analyze and design suitable circuits and subsystems
      for practical applications.",
  prereq_string: "",
  coreq_string: "",
  co_or_prereq_string: "",
  semester_offering: "",
  prereq_array: [],
  coreq_array: [],
  pre_or_coreq_array: []
},

{
  number: "ECE 584",
  title: "Photovoltaics",
  credits: 3,
  description: "Technical concepts of photovoltaics. Solar cell device level operation,
       packaging, manufacturing, designing phovoltaic system for stand-alone or grid-
      tied operation, some business-case analysis and some real-life scenarios of
```

```
        applicability of these solutions.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 585",
    title: "Modern Manufacturing Methods",
    credits: 3,
    description: "(Also offered as ME 585.) Study of business of manufacturing,
        emphasizing modern approaches. Topics include: U.S. manufacturing dilemma; JIT,
        kanban, pull manufacturing, quality; modeling; design for production;
        manufacturing economics; management issues; DIM; case studies.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 586",
    title: "Design for Manufacturability",
    credits: 3,
    description: "(Also offered as ME 586.) Introduction to methods of design for
        manufacturability (DFM). Emphasis is on teamwork and designing your customers
        needs. This is achieved through statistical methods and computer based systems.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 588",
    title: "Future Energy Systems",
    credits: 3,
    description: "A detailed study of current and emerging power and energy systems and
        technologies. Including renewable energies, storage, Smart Grid concepts,
        security for power infrastructure. Software modeling of power systems and grids."
        ,
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
```

```
  },

  {
    number: "ECE 590",
    title: "Graduate Seminar",
    credits: 1,
    description: "Offered on a CR/NC basis only.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 591",
    title: "INCBN Seminar",
    credits: 1,
    description: "Graduate seminar on Integrating Nanotechnology with Cell Biology and
        Neuroscience.  Grades based on active participation, including oral presentation.
        ",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 594",
    title: "Complex Systems Theory",
    credits: 3,
    description: "Advanced topics in complex systems including but not limited to
        biological systems social and technological networks, and complex dynamics.",
    prereq_string: "graduate standing.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 595",
    title: "Special Topics",
    credits: 4,
    description: "",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
```

```
    pre_or_coreq_array: []
  },

  {
    number: "ECE 599",
    title: "Masters Thesis",
    credits: 6,
    description: "Offered on a CR/NC basis only.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 620",
    title: "T: Interdis Bio and Biomed Sci",
    credits: 3,
    description: "(Also offered as ANTH 620, BIOL 520, CS 520, STAT 520). Varying
        interdisciplinary topics taught by collaborative scientists from UNM, SFI, and
        LANL.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 633",
    title: "T: Image Processing",
    credits: 3,
    description: "Advanced topics including but not limited to computational,
        mathematical, multi-scale, and spatial statistical methods for multi-dimensional
        signal processing, multi-spectral imagery, image and video processing.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 637",
    title: "Topics-Algorithms",
    credits: 3,
    description: "Advanced topics including parallel and high-performance computing,
        multimedia, virtual reality, real-time systems and robotics, encryption and
        security, information technology, applied algorithmics and computational science
        algorithms and applications.",
    prereq_string: "537.",
```

```
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 537"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 638",
    title: "Topics - Architecture & Systems",
    credits: 3,
    description: "Advanced topics including advanced computer architecture, networks,
        distributed computing, large-scale resource management, high-performance
        computing and grid-based computing.",
    prereq_string: "538.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 538"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 642",
    title: "Detection & Estimation Theory",
    credits: 3,
    description: "Hypothesis testing; Karhunen-Loeve representation; optimal detection of
         discrete- and continuous-time signals; ML, MMSE, and MAP estimation; sufficient
        statistics, estimation error bounds; Wiener and Kalman-Bucy filtering; detection/
        receivers for multiuser and multipath fading channels.",
    prereq_string: "541.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 541"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 651",
    title: "Problems",
    credits: 6,
    description: "",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 661",
    title: "Topics - Electromagnetics",
```

```
    credits: 3,
    description: "Topics include advanced antenna theory, electromagnetic scattering and
        propagation, electromagnetic compatibility, low temperature plasma science,
        advanced plasma physics, and other subjects in applied electromagnetics.",
    prereq_string: "561.",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: ["ECE 561"],
    coreq_array: [],
    pre_or_coreq_array: []
  },

  {
    number: "ECE 699",
    title: "Dissertation",
    credits: 12,
    description: "Offered on a CR/NC basis only.",
    prereq_string: "",
    coreq_string: "",
    co_or_prereq_string: "",
    semester_offering: "",
    prereq_array: [],
    coreq_array: [],
    pre_or_coreq_array: []
  }

])

Dept.where(acronym: 'ECE').first.courses = Array.new(courses)
```