

University of New Mexico
UNM Digital Repository

Electrical and Computer Engineering ETDs

Engineering ETDs

1-28-2015

Development of a Modern Computational Infrastructure Around University Curricula

Jarred Kozlick

Follow this and additional works at: https://digitalrepository.unm.edu/ece_etds

Recommended Citation

Kozlick, Jarred. "Development of a Modern Computational Infrastructure Around University Curricula." (2015).
https://digitalrepository.unm.edu/ece_etds/143

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

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

©2014, Jarred Kozlick

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	x
Glossary	xii
1 Introduction	1
2 High Level Design	3
2.1 Motivation	3
2.2 Advantages	5
2.3 Curriculum Description	8
2.3.1 Organizational Description	8
2.3.2 Requirements Description	13
3 Current Degree Plan Website	15
3.1 Web Applications Framework	15
3.2 Deployment	16

Contents

3.3	Relational Database Management System	17
3.4	Data Model	17
3.5	Application Interface	19
3.5.1	Application Demand	21
3.6	Limitations of Initial Implementation	24
3.6.1	Prerequisite Representation	24
3.6.2	Additional Semester Years	26
3.6.3	Data Ingestion	28
4	New Technologies in Support of Modern Computational Infrastructure	30
4.1	NoSQL Technologies	30
4.1.1	Graph Database	34
4.1.2	Document Database	35
5	Modern Computational Infrastructure Design	37
5.1	Newly Designed Infrastructure	37
5.1.1	Graph Database	37
5.1.2	Document Database	41
5.1.3	Departmental Degree Plans	42
5.1.4	Custom Degree Plans	42
5.1.5	Other Stored Data	42

Contents

5.2 Advantages	43
5.2.1 Graph Database	43
5.2.2 Document Database	47
6 Futurework/Conclusions	49
References	51
Appendices	53
A Course JSON Files	54
B Course Parsing Scripts	73
C Database Seed Files	82

List of Figures

2.1	Sample Degree Plan Spreadsheet	4
2.2	Description of Problem Domain	9
2.3	Requirements Breakdown	13
3.1	UNM Degree Plan Entity Relationship Diagram	18
3.2	UNM Degree Plan Landing Page	20
3.3	UNM Degree Plan Explore Page	21
3.4	UNM Degree Plan Graph View	22
3.5	Overview of Site Analytics	22
3.6	Top Ten Languages Requested	23
3.7	Degree Plans Geographic Locations	23
3.8	Degree Plans Site Usage	24
3.9	Ruby Prefix Equation Array	25
3.10	Prefix Equation Represented As A Tree	25
3.11	Binary Expression Tree Inserted Into Database	25

List of Figures

3.12	Entity Relationship Diagram With Year Entity Added	27
3.13	Example Prerequisite JSON obtained from IT	29
4.1	CAP Triangle	33
4.2	Example Cypher Query	35
4.3	Neo4j Administrative Dashboard Screenshot	35
5.1	University Relationships	38
5.2	Course Relationships	39
5.3	Sample of Degree Requirements Tree	40
5.4	University Relationships With Top Level Institution Node	44
5.5	Cypher Query to Find All Academic Programs at UNM	45
5.6	CHEM 121 Prerequisite Graph as described in UNM Catalog	46
5.7	Reduced CHEM 121 Prerequisite Graph	47
6.1	Envisioned System	50

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.

Glossary

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

Chapter 1. Introduction

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.

Chapter 2. High Level Design

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	Course Subject and Title	Cr. Hrs.	Major	Minor/2nd Major	Core	UD	Min Grade	Notes
Semester One:								Semester Two:							
ENGL 101 Composition I	3					C		ENGL 102 Composition 2	3					C	
MATH 152: Calculus I	4	4				C		MATH 153: Calculus II	4	4				C	
ECE 101: Intro to ECE Dept	1	1				C		PHYC 161: General Physics II	3	3				C	
ECE 131: Intro to Programming	3	3				C		PHYC 161L: General Physics II Lab	1	1				C	
PHYC 160: General Physics	3					C		ECE 231: Intermediate Programming	3	3				C	
ECON 105 or 106: Macro or Micro Econ	3					C		Humanities	3					C	
Total:	17							Total:	17						
Freshman Advisement anytime after 10th week - How to use the Degree Audit								Sophomore Advisement Enhanced Degree Audit skills							
Semester Three:								Semester Four:							
ECE 203: Circuit Analysis I	3	3				C		ECE 206L: EE Lab I	2	2				C	
ECE 238L: Comp Logic Design	4	4				C		ECE 213: Circuit Analysis II	3	3				C	
MATH 316: Diff Eqs	3	3				C		Math Elective	3	3				C	
Science with Lab	4	4				C		MATH 264L: Calculus III	4	4				C	
ENGL 219: Tech Writing	3					C		ECE 330: Software Design	3	3				C	
Total	17							Total	15						
Semester Once Grades are In...								Departmental Orientation within first 6 weeks							
Semester Five:								Semester Six:							
ECE 321: Electronics I	4	4				C		ECE Track Elective	3	3				C	
MATH 327: Discrete Structures	3	3				C		ECE 331: Data Structure Algorithms	3	3				C	
ECE 314: Signals and Systems	3	3				C		ECE 344L: Microprocessors	4	4				C	
ECE 331L: Intro Computer Arch Dig	3	3				C		ECE 340: Probabilistic Methods	3	3				C	
Second Language	3					C		Social Science	3					C	
Total	16							Total	16						
Visit Career Services								Apply for degree After 4th week							
Semester Seven:								Semester Eight:							
ECE 419: Senior Design I	3	3				C		ECE 420: Senior Design II	3	3				C	
ECE Track Elective	3	3				C		ECE 440: Computer Networks	3	3				C	
ECE 437L: Operating Systems	3	3				C		Technical Elective	3	3				C	
Technical Elective	3	3				C		Fine Arts	3					C	
Technical Elective	3	3				C		Humanities	3					C	
Total	15							Total	15						
Senior Visit - Advisement								Senior Visit Advisement							
The University of New Mexico Core Curriculum (37 units)								School of Engineering Minimum Requirements							
Writing and Speaking: (3-9 units)								Total credit hours = varies among degrees							
Mathematics: (3 units)								300/400 level credit hours = 56* (Could include 200 level math courses for Diff Eqs and Linear Algebra - 6 credits)							
Physical and Natural Sciences: (7 units)								Minimum credit hours taught in A&S = 96							
Social and Behavioral Sciences: (6 units)															
Humanities: (6 units)															
Foreign Language: (non-English language; 3 units)															
Fine Arts: (3 units)															
University Residence Requirements								Minimum graduation GPA = 2.00							
a. Minimum hours = 30								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 cumulative minimum.							
b. Senior standing = 15 past 92								For more information see the catalog at www.unm.edu .							
c. In major = One half															
d. In minor = One quarter															
Career Opportunities and Pathways								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							
➤															
Contact Information															
General Engineering Advisor:		Email:		Website:											
ESS		ess@unm.edu		http://openunm.edu											
Major Advisor:		Email:		Website:											
Christina Garcia		cgarcia@ece.unm.edu		http://ece.unm.edu											
College Advisor:		Email:		Website:											

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

Chapter 2. High Level Design

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

Chapter 2. High Level Design

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

Chapter 2. High Level Design

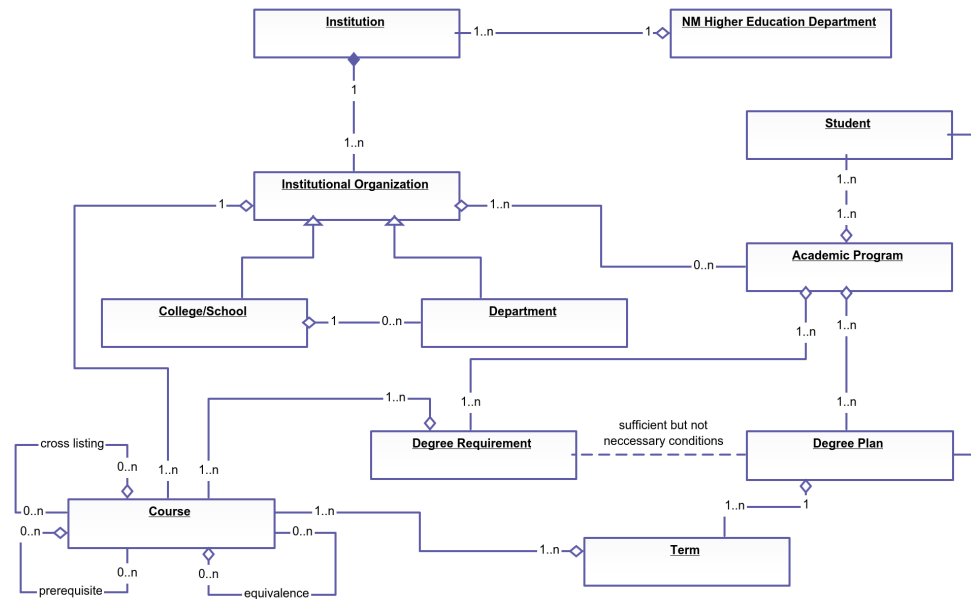


Figure 2.2: Description of Problem Domain

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

Chapter 2. High Level Design

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

Chapter 2. High Level Design

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

Chapter 2. High Level Design

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 in 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.

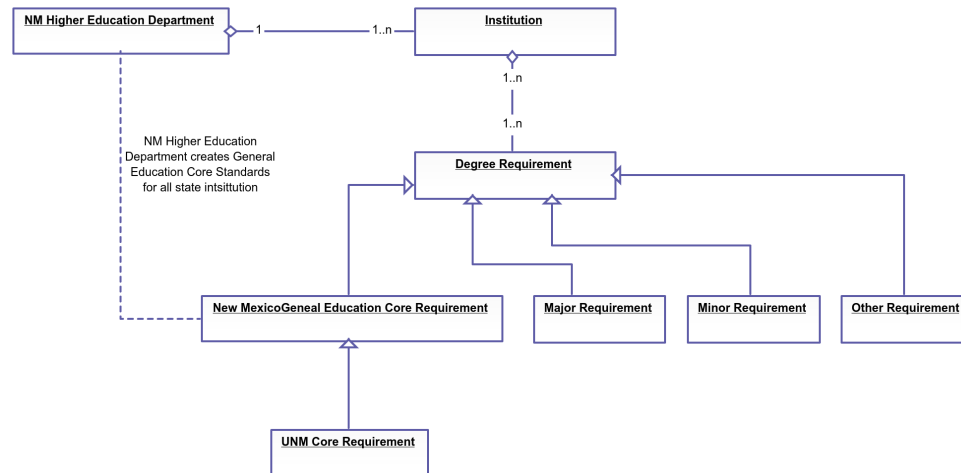


Figure 2.3: Requirements Breakdown

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

Chapter 2. High Level Design

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-

Chapter 3. Current Degree Plan Website

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

Chapter 3. Current Degree Plan Website

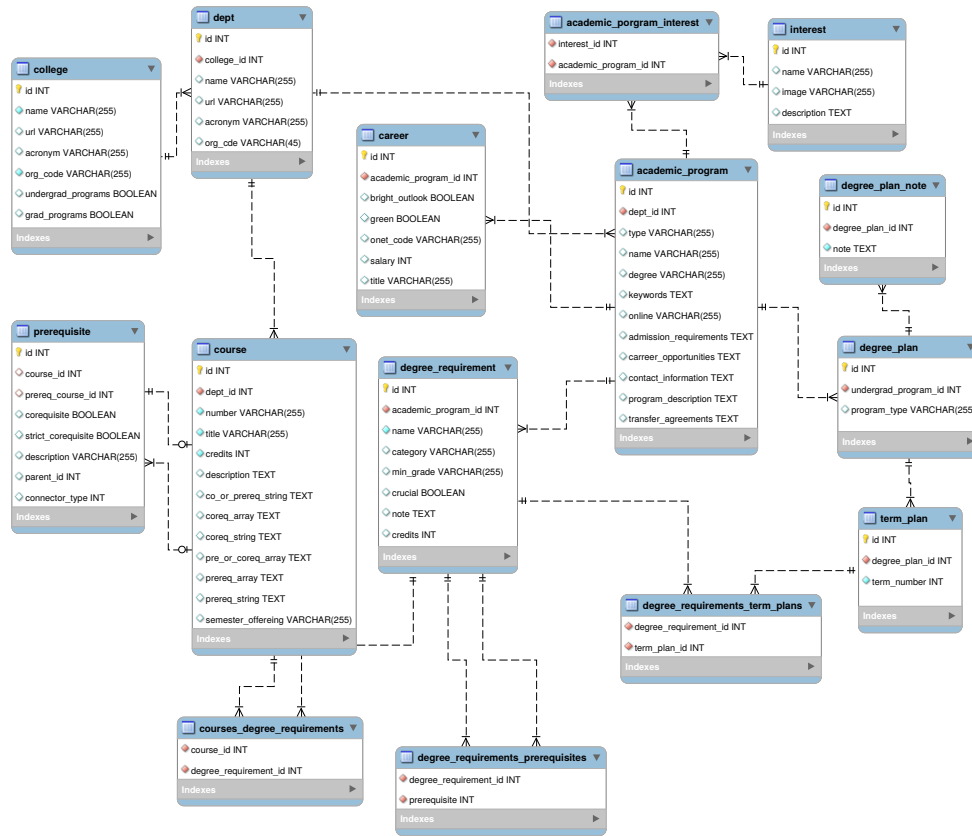


Figure 3.1: UNM Degree Plan Entity Relationship Diagram

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 area 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

Chapter 3. Current Degree Plan Website

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

Chapter 3. Current Degree Plan Website

3.2 shows a screen capture of the landing page.

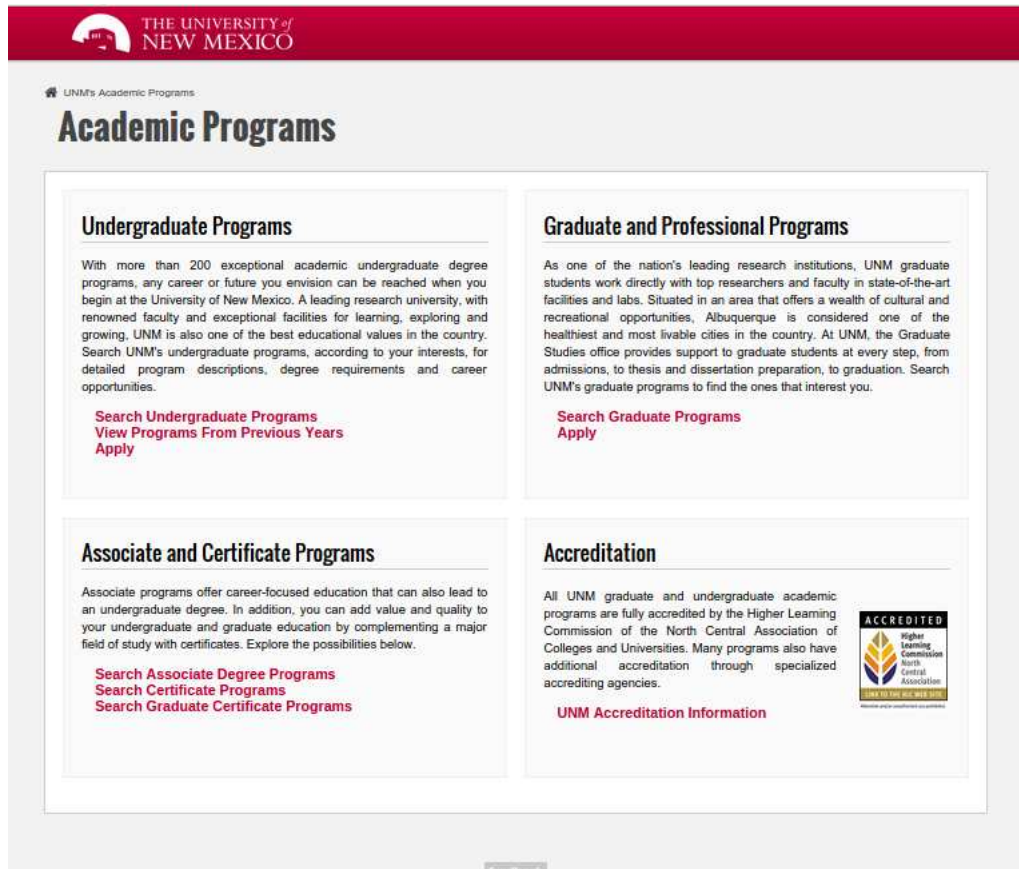


Figure 3.2: UNM Degree Plan Landing Page

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.

Chapter 3. Current Degree Plan Website

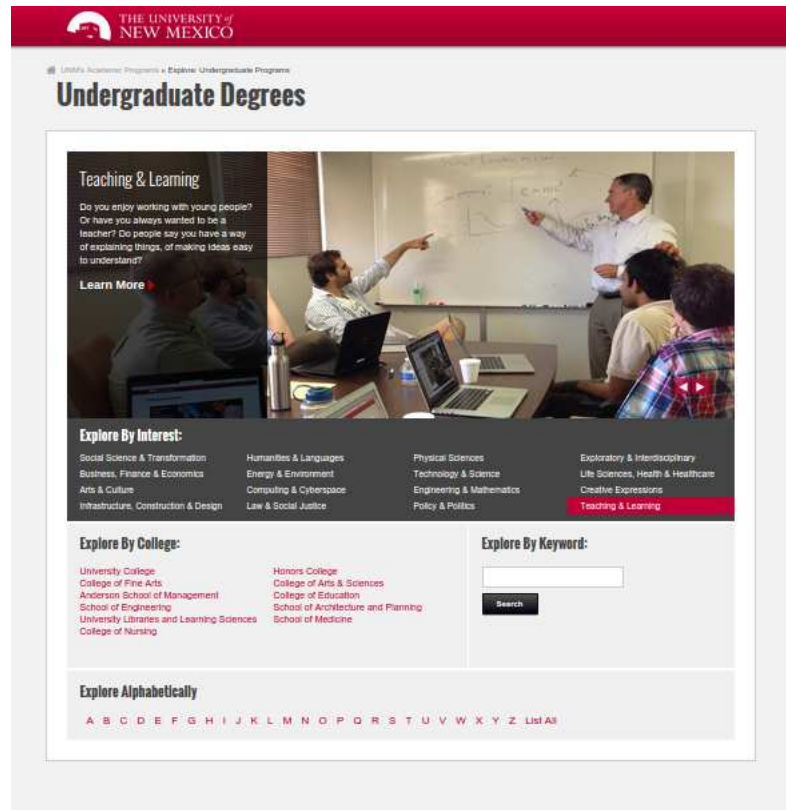


Figure 3.3: UNM Degree Plan Explore Page

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,

Chapter 3. Current Degree Plan Website

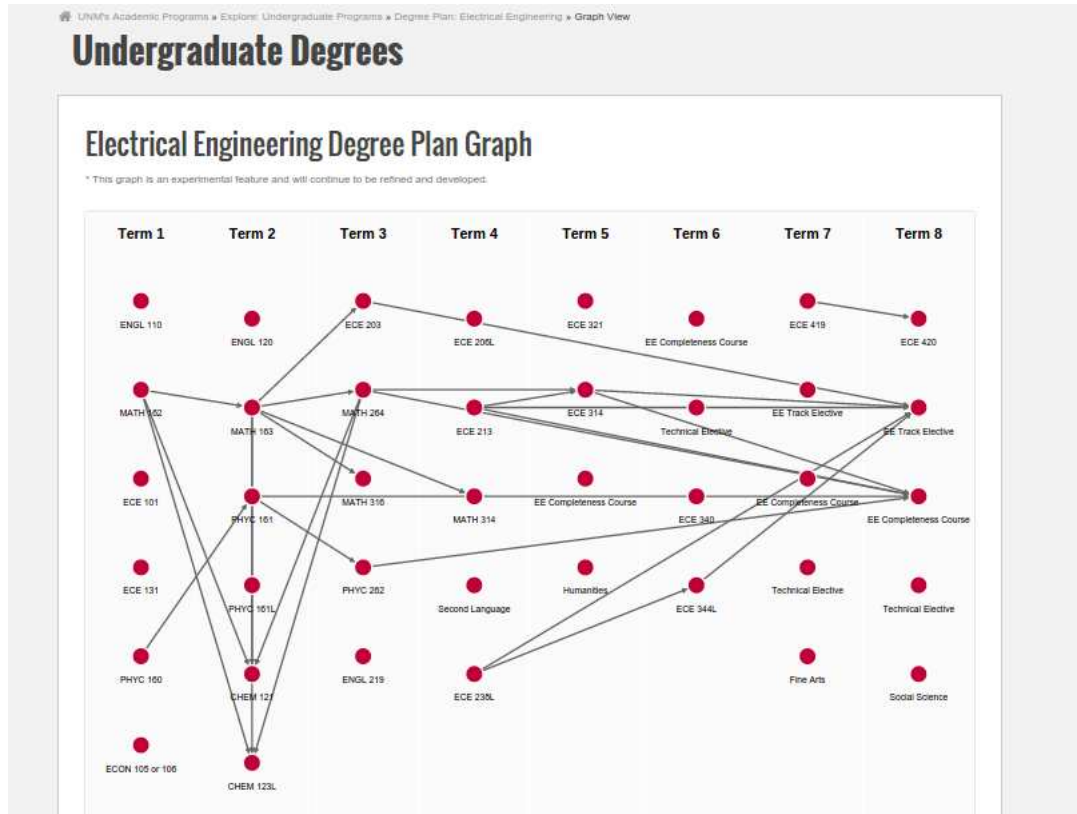


Figure 3.4: UNM Degree Plan Graph View

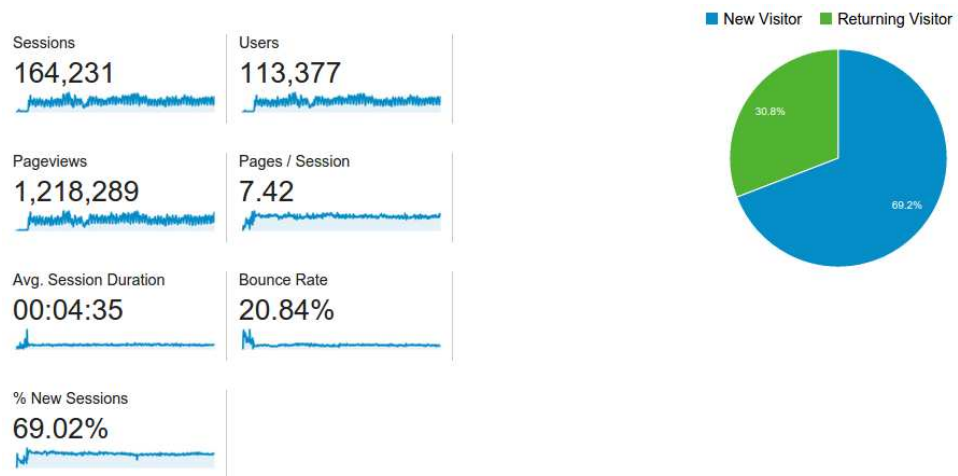


Figure 3.5: Overview of Site Analytics

Chapter 3. Current Degree Plan Website

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.

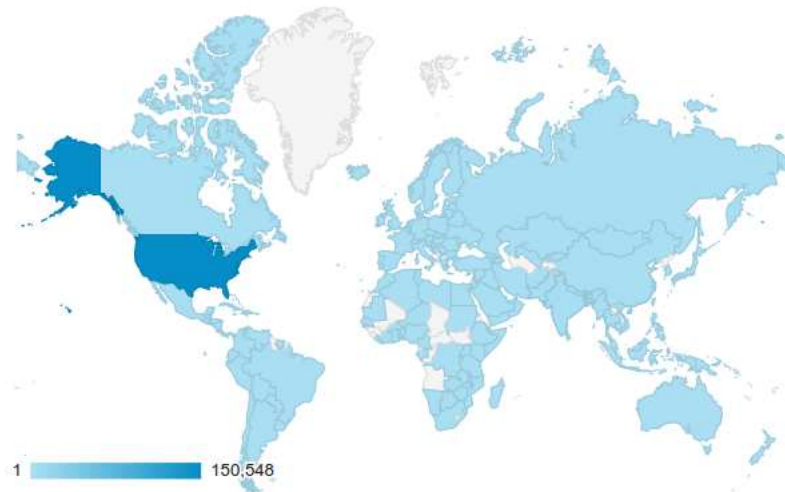


Figure 3.7: Degree Plans Geographic Locations

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

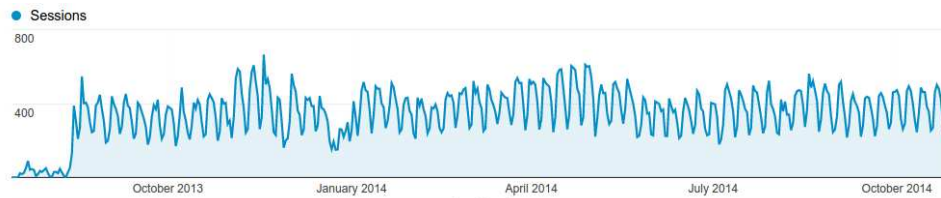


Figure 3.8: Degree Plans Site Usage

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

Chapter 3. Current Degree Plan Website

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

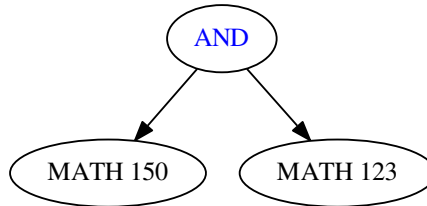


Figure 3.10: Prefix Equation Represented As A Tree

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

Chapter 3. Current Degree Plan Website

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 searched by year, and then either displayed in the archive or the section of the

Chapter 3. Current Degree Plan Website

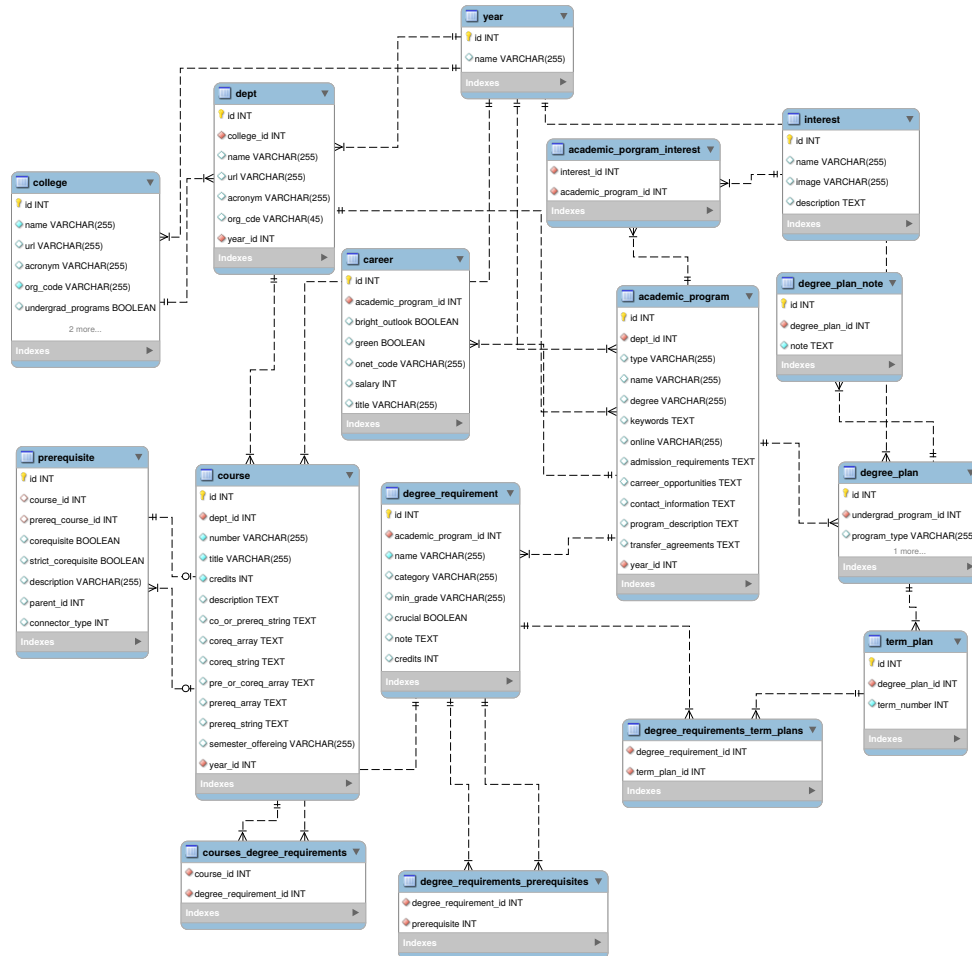


Figure 3.12: Entity Relationship Diagram With Year Entity Added

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

Chapter 3. Current Degree Plan Website

```
1  "prerequisites": [  
2    {  
3      "name": "( ACT Math  minimum score 28 "  
4    },  
5    {  
6      "name": " or SAT Mathematics  minimum score 640 "  
7    },  
8    {  
9      "name": " or MATH 150 "  
10   },  
11   {  
12     "name": " or COMPASS - College Algebra  minimum score 67 )"  
13   },  
14   {  
15     "name": " and ( MATH 123 "  
16   },  
17   {  
18     "name": " or COMPASS - Trigonometry  minimum score 60 )"  
19   },  
20   {  
21     "name": " or ( ACT Math  minimum score 32 "  
22   },  
23   {  
24     "name": " or SAT Mathematics  minimum score 700 )"  
25   }  
26 ]
```

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.

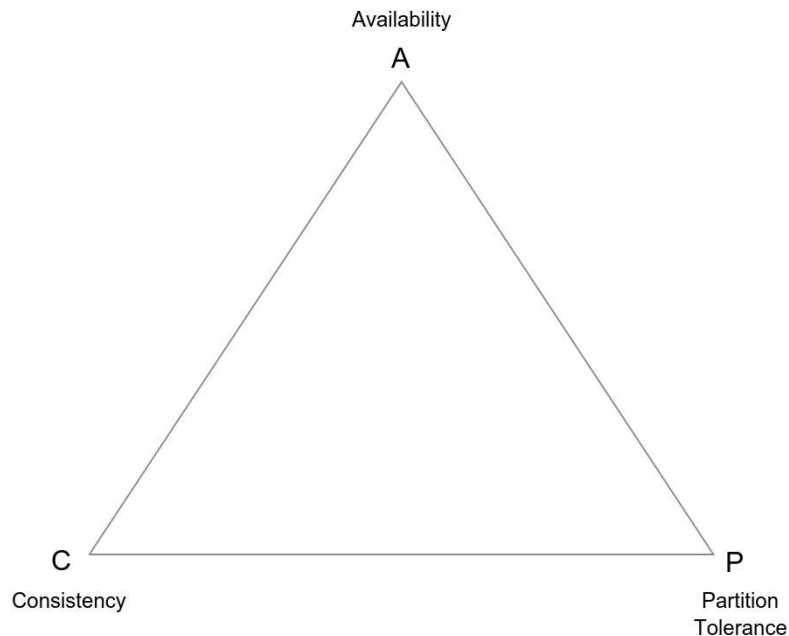


Figure 4.1: CAP Triangle

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 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.

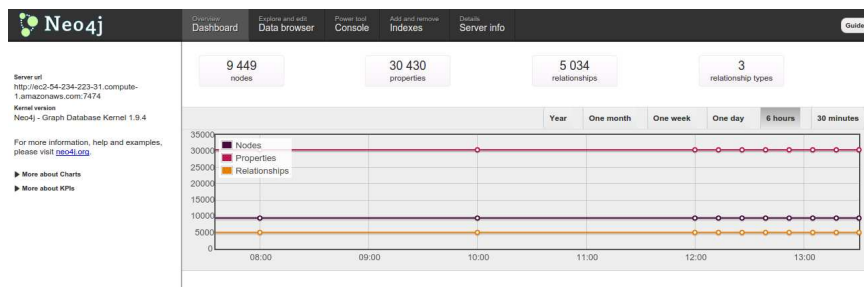


Figure 4.3: Neo4j Administrative Dashboard Screenshot

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.

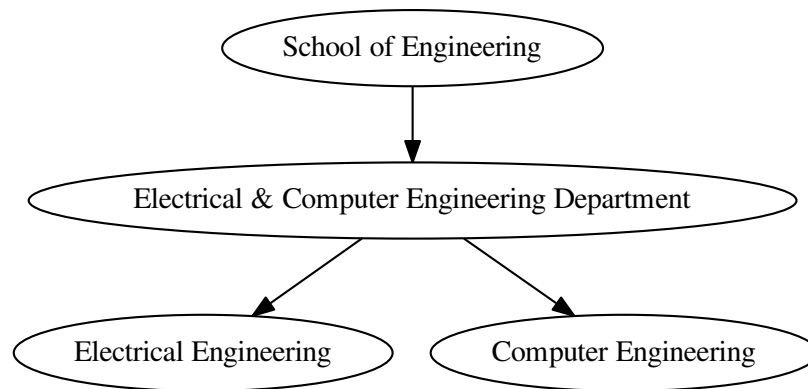


Figure 5.1: University Relationships

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.

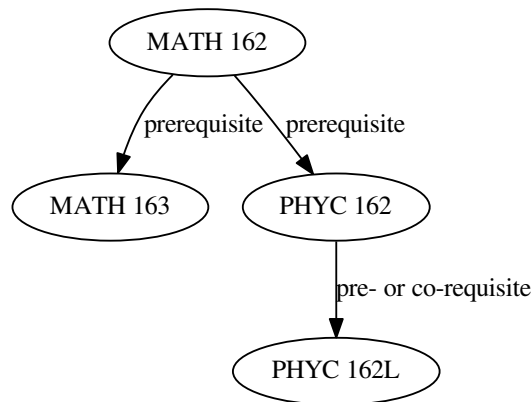


Figure 5.2: Course Relationships

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.

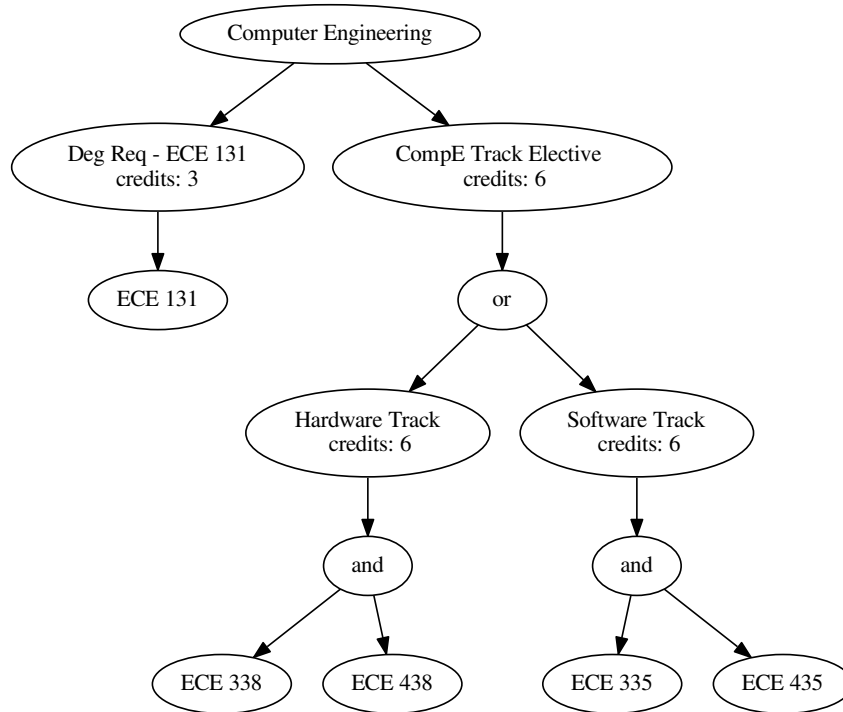


Figure 5.3: Sample of Degree Requirements Tree

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.

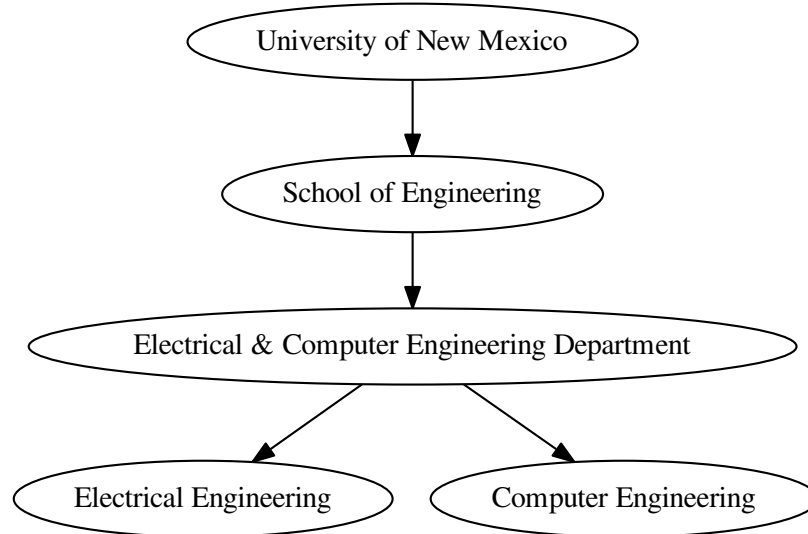


Figure 5.4: University Relationships With Top Level Institution Node

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

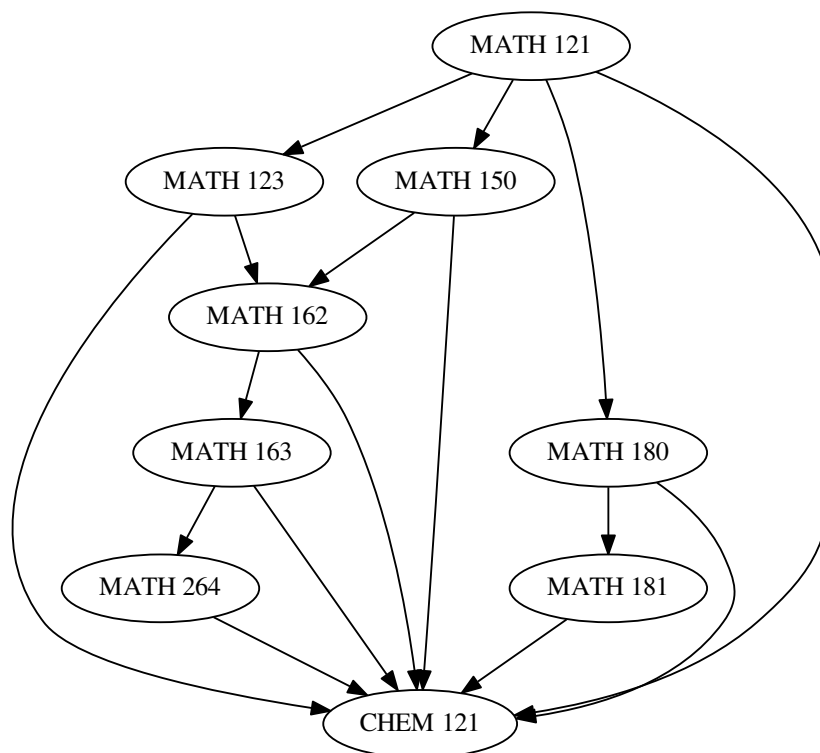


Figure 5.6: CHEM 121 Prerequisite Graph as described in UNM Catalog

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

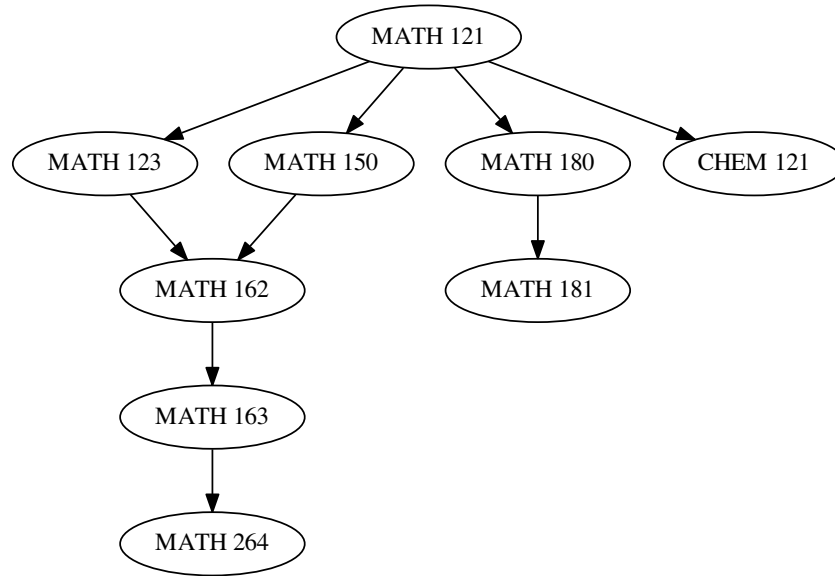


Figure 5.7: Reduced CHEM 121 Prerequisite Graph

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

Chapter 5. Modern Computational Infrastructure Design

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.

Chapter 6. Futurework/Conclusions

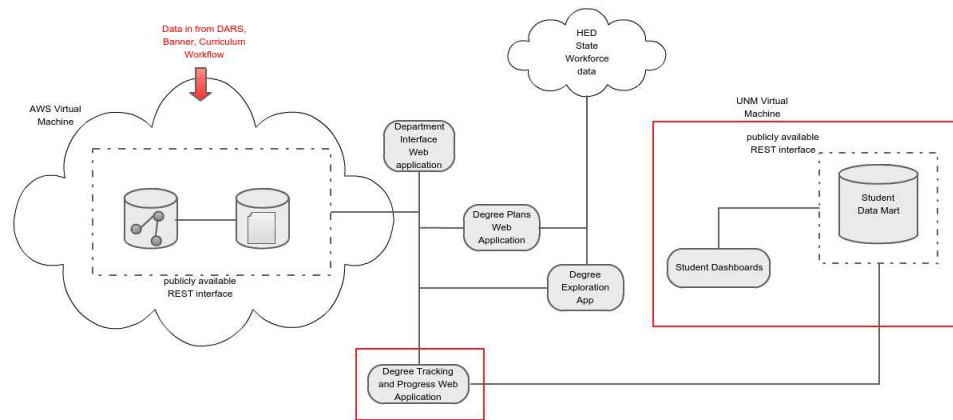


Figure 6.1: Envisioned System

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 Härder 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.

References

- [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

A Course JSON Files	54
B Course Parsing Scripts	72
C Database Seed Files	81

Appendix A

Course JSON Files

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

```
1  [
2    {
3      "name": "ECE 101",
4      "title": "Intro to Elect & Computer Eng",
5      "credits": "1",
6      "description": "Insight into electrical engineering is gained through videos,
7                    hands-on experiments, use of computer software to learn basic problem-
8                    solving skills and a team-oriented design project.",
9      "department": "Electrical Computer Engr",
10     "prerequisites": [],
11     "corequisites": []
12   },
13   {
14     "name": "ECE 131",
15     "title": "Programming Fundamentals",
16     "credits": "3",
17     "description": "Fundamental programming concepts, including consideration of
18                   abstract machine models with emphasis on the memory hierarchy, basic
19                   programming constructs, functions, parameter passing, pointers and arrays,
20                   file I/O, bit-level operations and interfacing to external devices.",
21     "department": "Electrical Computer Engr",
22     "prerequisites": [],
23     "corequisites": []
24   },
25   {
26     "name": "ECE 203",
27     "title": "Circuit Analysis I",
28     "credits": "3",
29     "description": "Basic elements and sources. Energy and power. Ohm's law and
30                   Kirchhoff's laws. Resistive networks, node and loop analysis. Network
31                   theorems. First-order and second-order circuits. Sinusoidal sources and
```

Appendix A. Course JSON Files

```
    complex representations: impedance, 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         }
77     ]
78 }
```

Appendix A. Course JSON Files

```
77     ],
78     "corequisites": []
79   },
80   {
81     "name": "ECE 213",
82     "title": "Circuit Analysis II",
83     "credits": "3",
84     "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.",
85     "department": "Electrical Computer Engr",
86     "prerequisites": [
87       {
88         "name": "( ECE 203L"
89       },
90       {
91         "name": " or ECE 203"
92       },
93       {
94         "name": " or EECE 203L )"
95       },
96       {
97         "name": " and MATH 316"
98       }
99     ],
100    "corequisites": [
101      {
102        "name": "MATH 314"
103      }
104    ]
105  },
106  {
107    "name": "ECE 231",
108    "title": "Intermediate Programming",
109    "credits": "3",
110    "description": "Introduction 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.",
111    "department": "Electrical Computer Engr",
112    "prerequisites": [
113      {
114        "name": "ECE 131"
115      }
116    ],
117    "corequisites": []
118  },
119  {
120    "name": "ECE 238L",
121    "title": "Computer Logic Design",
122    "credits": "4",
123    "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.",
124    "department": "Electrical Computer Engr",
125    "prerequisites": [
```

Appendix A. Course JSON Files

```
126         {
127             "name": "ECE 131"
128         }
129     ],
130     "corequisites": []
131 },
132 {
133     "name": "ECE 314",
134     "title": "Signals and Systems",
135     "credits": "3",
136     "description": "Continuous and discrete time signals and systems; time and
137         frequency domain analysis of LTI systems, Fourier series and transforms,
138         discrete time Fourier series/transform sampling theorem, block diagrams,
139         modulation/demodulation, filters. Prerequisites: 213 and MATH 264.",
140     "department": "Electrical Computer Engr",
141     "prerequisites": [
142         {
143             "name": "ECE 213"
144         },
145         {
146             "name": " and MATH 264"
147         }
148     ],
149     "corequisites": []
150 },
151 {
152     "name": "ECE 321L",
153     "title": "Electronics I",
154     "credits": "4",
155     "description": "Introduction to diodes, bipolar and field-effect transistors.
156         Analysis and design of digital circuits, gates, flip-flops and memory
157         circuits. Circuits employing operational amplifiers. Analog to digital and
158         digital to analog converters. Prerequisite: 213.",
159     "department": "Electrical Computer Engr",
160     "prerequisites": [
161         {
162             "name": "ECE 213"
163         }
164     ],
165     "corequisites": []
166 },
167 {
168     "name": "ECE 322L",
169     "title": "Electronics II",
170     "credits": "4",
171     "description": "Analysis, design, and characterization of linear circuits
172         including operational amplifiers. Design of biasing and reference circuits,
173         multistage amplifiers, and feedback circuits. Prerequisite: 321L.",
174     "department": "Electrical Computer Engr",
175     "prerequisites": [
176         {
177             "name": "ECE 321L"
178         }
179     ],
180     "corequisites": []
181 },
182 {
183     "name": "ECE 330",
```

Appendix A. Course JSON Files

```
176     "title": "Software Design",
177     "credits": "3",
178     "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.",
179     "department": "Electrical Computer Engr",
180     "prerequisites": [
181         {
182             "name": "ECE 231"
183         },
184         {
185             "name": " or EECE 231L"
186         },
187         {
188             "name": " or ECE 231L"
189         }
190     ],
191     "corequisites": []
192 },
193 {
194     "name": "ECE 331",
195     "title": "Data Structures & Algorithms",
196     "credits": "3",
197     "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.",
198     "department": "Electrical Computer Engr",
199     "prerequisites": [
200         {
201             "name": "ECE 231"
202         },
203         {
204             "name": " and MATH 327"
205         }
206     ],
207     "corequisites": [
208         {
209             "name": "ECE 340"
210         }
211     ]
212 },
213 {
214     "name": "ECE 335",
215     "title": "Integrated Software Systems",
216     "credits": "3",
217     "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 architecture, integration and
                    security for large-scale software systems.",
218     "department": "Electrical Computer Engr",
219     "prerequisites": [
220         {
221             "name": "ECE 330"
222         },
223         {
```


Appendix A. Course JSON Files

```
224         "name": " and ECE 337 "
225     }
226 ],
227     "corequisites": []
228 },
229 {
230     "name": "ECE 337",
231     "title": "Computer Arch & Organization",
232     "credits": "3",
233     "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)",
234     "department": "Electrical Computer Engr",
235     "prerequisites": [
236         {
237             "name": "( ECE 231 "
238         },
239         {
240             "name": " or EECE 231L"
241         },
242         {
243             "name": " or ECE 231L )"
244         },
245         {
246             "name": " and ECE 238L"
247         }
248     ],
249     "corequisites": []
250 },
251 {
252     "name": "ECE 338",
253     "title": "Intermediate Logic Design",
254     "credits": "3",
255     "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.",
256     "department": "Electrical Computer Engr",
257     "prerequisites": [
258         {
259             "name": "ECE 238L"
260         }
261     ],
262     "corequisites": []
263 },
264 {
265     "name": "ECE 340",
266     "title": "Probabilistic Methods in Eng",
267     "credits": "3",
268     "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.",
269     "department": "Electrical Computer Engr",
270     "prerequisites": [
271         {
```

Appendix A. Course JSON Files

```
272         "name": "ECE 314"
273     },
274     {
275         "name": " and MATH 314"
276     }
277 ],
278 "corequisites": []
279 },
280 {
281     "name": "ECE 341",
282     "title": "Communication Systems",
283     "credits": "3",
284     "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.",
285     "department": "Electrical Computer Engr",
286     "prerequisites": [
287         {
288             "name": "ECE 314"
289         },
290         {
291             "name": " and ECE 340"
292         }
293     ],
294     "corequisites": []
295 },
296 {
297     "name": "ECE 344L",
298     "title": "Microprocessors",
299     "credits": "4",
300     "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)",
301     "department": "Electrical Computer Engr",
302     "prerequisites": [
303         {
304             "name": "ECE 206L"
305         },
306         {
307             "name": " and ECE 238L"
308         },
309         {
310             "name": " and ECE 321L"
311         }
312     ],
313     "corequisites": []
314 },
315 {
316     "name": "ECE 345",
317     "title": "Intro to Control Systems",
318     "credits": "3",
319     "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.",
320     "department": "Electrical Computer Engr",
321     "prerequisites": [
```

Appendix A. Course JSON Files

```
322     {
323         "name": "ECE 314"
324     }
325 ],
326 "corequisites": []
327 },
328 {
329     "name": "ECE 360",
330     "title": "Electromagnetic Fields & Waves",
331     "credits": "3",
332     "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.",
333     "department": "Electrical Computer Engr",
334     "prerequisites": [
335         {
336             "name": "ECE 213"
337         },
338         {
339             "name": " and PHYC 161"
340         },
341         {
342             "name": " and MATH 264"
343         }
344     ],
345     "corequisites": []
346 },
347 {
348     "name": "ECE 371",
349     "title": "Materials & Devices",
350     "credits": "4",
351     "description": "Introduction to quantum mechanics, crystal structures,
        insulators, metals, and semiconductor material properties, bipolar, field
        effect and light emitting devices. Prerequisite: PHYC 262.",
352     "department": "Electrical Computer Engr",
353     "prerequisites": [
354         {
355             "name": "PHYC 262"
356         }
357     ],
358     "corequisites": []
359 },
360 {
361     "name": "ECE 381",
362     "title": " Intro to Power Systems",
363     "credits": "3",
364     "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.",
365     "department": "Electrical Computer Engr",
366     "prerequisites": [
367         {
368             "name": "ECE 213"
369         }
370     ],
371     "corequisites": []
```

Appendix A. Course JSON Files

```
372     },
373     {
374         "name": "ECE 412",
375         "title": "Intro to Computer Graphics",
376         "credits": "3",
377         "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.",
378         "department": "Computer Science",
379         "prerequisites": [
380             {
381                 "name": "ECE 331"
382             },
383             {
384                 "name": " or CS 361L"
385             }
386         ],
387         "corequisites": []
388     },
389     {
390         "name": "ECE 413",
391         "title": "Intro to Ray Graphics",
392         "credits": "3",
393         "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.",
394         "department": "Electrical Computer Engr",
395         "prerequisites": [
396             {
397                 "name": "CS 361L"
398             },
399             {
400                 "name": " or ECE 331"
401             }
402         ],
403         "corequisites": []
404     },
405     {
406         "name": "ECE 419",
407         "title": "Senior Design I",
408         "credits": "3",
409         "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.",
410         "department": "Electrical Computer Engr",
411         "prerequisites": [],
412         "corequisites": []
413     },
414     {
415         "name": "ECE 420",
416         "title": "Senior Design II",
417         "credits": "3",
```

Appendix A. Course JSON Files

```
418     "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.",
419     "department " : "Electrical Computer Engr",
420     "prerequisites": [
421         {
422             "name": "ECE 419"
423         }
424     ],
425     "corequisites": []
426 },
427 {
428     "name": "ECE 421",
429     "title": "Analog Electronics",
430     "credits": "3",
431     "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.",
432     "department " : "Electrical Computer Engr",
433     "prerequisites": [
434         {
435             "name": "ECE 322L"
436         }
437     ],
438     "corequisites": []
439 },
440 {
441     "name": "ECE 424",
442     "title": "Digital VLSI Design",
443     "credits": "3",
444     "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.",
445     "department " : "Electrical Computer Engr",
446     "prerequisites": [
447         {
448             "name": "ECE 321L"
449         },
450         {
451             "name": " and ECE 338 "
452         }
453     ],
454     "corequisites": []
455 },
456 {
457     "name": "ECE 432",
458     "title": "Intro to Parallel Processing",
459     "credits": "3",
460     "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).",
461     "department " : "Electrical Computer Engr",
462     "prerequisites": [
```

Appendix A. Course JSON Files

```
463     {
464         "name": "( ECE 331 "
465     },
466     {
467         "name": " or CS 351L )"
468     },
469     {
470         "name": " and ( ECE 337 "
471     },
472     {
473         "name": " or CS 341L )"
474     }
475 ],
476 "corequisites": []
477 },
478 {
479     "name": "ECE 435",
480     "title": "Software Engineering",
481     "credits": "3",
482     "description": "Management and technical issues including business conduct and
483         ethics related to the design of large engineering projects. Student teams
484         will address the design, specification, implementation, testing and
485         documentation of a large hardware/software project. Prerequisites: 331 and
486         335.",
487     "department": "Electrical Computer Engr",
488     "prerequisites": [
489         {
490             "name": "ECE 331"
491         },
492         {
493             "name": " and ECE 335 "
494         }
495     ],
496     "corequisites": []
497 },
498 {
499     "name": "ECE 437",
500     "title": "Computer Operating Systems",
501     "credits": "3",
502     "description": "(Also offered as CS 481.) Fundamental principles of modern
503         operating systems design, with emphasis on concurrency and resource
504         management. Topics include processes, interprocess communication, semaphores,
505         monitors, message passing, input/output device, deadlocks memory management,
506         files system design. Prerequisites: (330 and 337) or CS 341L.",
507     "department": "Electrical Computer Engr",
508     "prerequisites": [
509         {
510             "name": "( ECE 330 "
511         },
512         {
513             "name": " and ECE 337 )"
514         },
515         {
516             "name": " or CS 341L"
517         }
518     ],
519     "corequisites": []
520 },
```

Appendix A. Course JSON Files

```
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     {
```

Appendix A. Course JSON Files

```
559     "name": "ECE 441",
560     "title": "Communication Systems",
561     "credits": "3",
562     "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.",
563     "department": "Electrical Computer Engr",
564     "prerequisites": [
565         {
566             "name": "ECE 314"
567         },
568         {
569             "name": " and ECE 340"
570         }
571     ],
572     "corequisites": [],
573 },
574 {
575     "name": "ECE 442",
576     "title": "Wireless Communication",
577     "credits": "3",
578     "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.",
579     "department": "Electrical Computer Engr",
580     "prerequisites": [
581         {
582             "name": "ECE 314"
583         },
584         {
585             "name": " and ECE 360"
586         }
587     ],
588     "corequisites": [],
589 },
590 {
591     "name": "ECE 443",
592     "title": "Hardware Design with VHDL",
593     "credits": "3",
594     "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.",
595     "department": "Electrical Computer Engr",
596     "prerequisites": [
597         {
598             "name": "ECE 338"
599         }
600     ],
601     "corequisites": [],
602 },
603 {
604     "name": "ECE 446",
605     "title": "Feedback Control Systems",
606     "credits": "3",
```


Appendix A. Course JSON Files

```
607     "description " : "Modeling of continuous and sampled-data control systems. State-
608         space representation. Sensitivity, stability and optimization of control
609         systems. Design of compensators in the frequency and time domains. Phase-
610         plane, describing function design for non-linear systems, and laboratory
611         design project. Prerequisite: 345.",
612     "department " : "Electrical Computer Engr",
613     "prerequisites" : [
614         {
615             "name": "ECE 345"
616         }
617     ],
618     "corequisites" : []
619 },
620 {
621     "name": "ECE 448",
622     "title": "Fuzzy Logic with Applications",
623     "credits": "3",
624     "description " : "(Also offered as CE 448.) Theory of fuzzy sets; foundations of
625         fuzzy logic. Fuzzy logic is shown to contain evidence, possibility and
626         probability logics; course emphasizes engineering applications; control,
627         pattern recognition, damage assessment, decisions; hardware/software
628         demonstrations. ",
629     "department " : "Electrical Computer Engr",
630     "prerequisites" : [],
631     "corequisites" : []
632 },
633 {
634     "name": "ECE 456",
635     "title": "Entrepreneurial Engineering",
636     "credits": "3",
637     "description " : "Review and application of necessary elements for successfully
638         launching technical businesses; focuses upon technology, manufacturing,
639         management, marketing, legal and financial aspects. Students work in groups
640         developing elements of new businesses and producing business plans.
641         Restriction: senior standing.",
642     "department " : "Electrical Computer Engr",
643     "prerequisites" : [],
644     "corequisites" : []
645 },
646 {
647     "name": "ECE 460",
648     "title": "Microwave Engineering",
649     "credits": "3",
650     "description " : "This lecture/laboratory course provides essential fundamentals
651         for rf, wireless and microwave engineering. Topics include: wave propagation
652         in cables, waveguides and free space; impedance matching, standing wave
653         ratios, Z- and S- parameters. Prerequisite: 360.",
654     "department " : "Electrical Computer Engr",
655     "prerequisites" : [
656         {
657             "name": "ECE 360"
658         }
659     ],
660     "corequisites" : []
661 },
662 {
663     "name": "ECE 463",
664     "title": "Advanced Optics I",
```

Appendix A. Course JSON Files

```
650     "credits": "3",
651     "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.",
652     "department": "Electrical Computer Engr",
653     "prerequisites": [
654         {
655             "name": "PHYC 463"
656         }
657     ],
658     "corequisites": [],
659 },
660 {
661     "name": "ECE 464",
662     "title": "Laser Physics I",
663     "credits": "3",
664     "description": "(Also offered as PHYC 464.) Resonator optics. Rate equations;
        spontaneous and stimulated emission; gas, semiconductor and solid state lasers
        , pulsed and mode-locked laser techniques. Prerequisite: 360 or PHYC 406.",
665     "department": "Electrical Computer Engr",
666     "prerequisites": [
667         {
668             "name": "ECE 360"
669         },
670         {
671             "name": " or PHYC 406"
672         }
673     ],
674     "corequisites": [],
675 },
676 {
677     "name": "ECE 469",
678     "title": "Antennas for Wireless Comm",
679     "credits": "3",
680     "description": "Aspects of antenna theory and design; radiation from dipoles,
        loops, apertures, microstrip antennas and antenna arrays. Prerequisite: 360.",
681     "department": "Electrical Computer Engr",
682     "prerequisites": [
683         {
684             "name": "ECE 360"
685         }
686     ],
687     "corequisites": [],
688 },
689 {
690     "name": "ECE 471",
691     "title": "Materials & Devices II",
692     "credits": "3",
693     "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.",
694     "department": "Electrical Computer Engr",
695     "prerequisites": [
696         {
697             "name": "ECE 360"
698         },
699         {
```

Appendix A. Course JSON Files

```
700         "name": " and ECE 371"
701     }
702 ],
703     "corequisites": []
704 },
705 {
706     "name": "ECE 474L",
707     "title": "Microelectronics Processing I",
708     "credits": "3",
709     "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.",
710     "department": "Electrical Computer Engr",
711     "prerequisites": [],
712     "corequisites": []
713 },
714 {
715     "name": "ECE 475",
716     "title": "Optoelectronics",
717     "credits": "3",
718     "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.",
719     "department": "Electrical Computer Engr",
720     "prerequisites": [
721         {
722             "name": "ECE 371"
723         }
724     ],
725     "corequisites": []
726 },
727 {
728     "name": "ECE 482",
729     "title": "Electric Drives & Transformers",
730     "credits": "3",
731     "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.",
732     "department": "Electrical Computer Engr",
733     "prerequisites": [
734         {
735             "name": "ECE 203"
736         }
737     ],
738     "corequisites": [
739         {
740             "name": " and ECE 213"
741         }
742     ],
743     "corequisites": []
744 },
745 {
746     "name": "ECE 483",
747     "title": "Power Electronics",
748     "credits": "3",
```

Appendix A. Course JSON Files

```
747     "description " : "Introduces modern power conversion techniques at a lower level ,
748         dealing with basic structures of power converters and techniques of analyzing
749         converter circuits. Students learn to analyze and design suitable circuits
750         and subsystems for practical applications. Prerequisite: 321L and 371 and 381.
751     " ,
752     "department " : "Electrical Computer Engr" ,
753     "prerequisites" : [
754         {
755             "name" : "ECE 321L"
756         } ,
757         {
758             "name" : " and ECE 371"
759         } ,
760         {
761             "name" : " and ECE 381"
762         }
763     ] ,
764     "corequisites" : []
765 } ,
766 {
767     "name" : "ECE 484" ,
768     "title" : "Photovoltaics" ,
769     "credits" : "3" ,
770     "description " : "Technical concepts of photovoltaics. Solar cell device level
771         operation , packaging , manufacturing , designing phovoltaic system for stand -
772         alone or grid-tied operation , some business-case analysis and some real-life
773         scenarios of applicability of these solutions. Prerequisite: 381 and MATH
774         121." ,
775     "department " : "Electrical Computer Engr" ,
776     "prerequisites" : [
777         {
778             "name" : "ECE 381"
779         } ,
780         {
781             "name" : " and MATH 121"
782         }
783     ] ,
784     "corequisites" : []
785 } ,
786 {
787     "name" : "ECE 488" ,
788     "title" : "Future Energy Systems" ,
789     "credits" : "3" ,
790     "description " : "A detailed study of current and emerging power and energy
791         systems and technologies. Including renewable energies , storage , Smart Grid
792         concepts , security for power infrastructure. Software modeling of power
793         systems and grids. Prerequisite: 381 and 482 and 483 and 484." ,
794     "department " : "Electrical Computer Engr" ,
795     "prerequisites" : [
796         {
797             "name" : "ECE 381"
798         } ,
799         {
800             "name" : " and ECE 482"
801         } ,
802         {
803             "name" : " and ECE 483"
804         }
805     ] ,
```

Appendix A. Course JSON Files

```
794     {
795         "name": " and ECE 484"
796     }
797 ],
798 "corequisites": []
799 },
800 {
801     "name": "ECE 490",
802     "title": "Internship",
803     "credits": "3",
804     "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.",
805     "department ": "Electrical Computer Engr",
806     "prerequisites": [],
807     "corequisites": []
808 },
809 {
810     "name": "ECE 491",
811     "title": "Undergrad Problems",
812     "credits": "1 TO 6",
813     "description ": "Registration for more than 3 hours requires permission of
                        department chairperson.",
814     "department ": "Electrical Computer Engr",
815     "prerequisites": [],
816     "corequisites": []
817 },
818 {
819     "name": "ECE 493",
820     "title": "Honors Seminar",
821     "credits": "1 TO 3",
822     "description ": "A special seminar open only to honors students. Registration
                        requires permission of department chairperson.",
823     "department ": "Electrical Computer Engr",
824     "prerequisites": [],
825     "corequisites": []
826 },
827 {
828     "name": "ECE 494",
829     "title": "Honors Individual Study",
830     "credits": "1 TO 6",
831     "description ": "Open only to honors students. Registration requires permission
                        of the department chairperson and of the supervising professor.",
832     "department ": "Electrical Computer Engr",
833     "prerequisites": [],
834     "corequisites": []
835 },
836 {
837     "name": "ECE 495",
838     "title": "Special Topics",
839     "credits": "1 TO 4",
840     "description ": "Restriction: ECE major, senior standing. ",
841     "department ": "Electrical Computer Engr",
842     "prerequisites": [],
843     "corequisites": []
```

Appendix A. Course JSON Files

```
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.

```
1 require 'open-uri'
2 require 'json'
3 require 'active_support/all'
4 require_relative './parse_array.rb'
5
6 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"]
7
8 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" => []}
9
10 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
```

Appendix B. Course Parsing Scripts

```

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

```


Appendix B. Course Parsing Scripts

```

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\t\tdescription: \"#{
    course_description}\",\n\t\t\tprereq_string: \"#{course_prereq_string}\",\n\t\t
    \tcoreq_string: \"#{course_coreq_string}\",\n\t\t\tco_or_prereq_string: \"#{
    course_co_or_prereq_string}\",\n\t\t\tsemester_offering: \"#{
    course_semester_offering}\",\n\t\t\tprereq_array: #{prereqs_array.to_s},\n\t\t
    \tcoreq_array: #{coreqs_array.to_s},\n\t\t\tpre_or_coreq_array: #{
    pre_or_coreqs_array.to_s}\n\t\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

```

Appendix B. Course Parsing Scripts

```
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 +
20        2])
21        infix_array[stack.pop] = nil
22        infix_array[index] = nil
23
24      elsif index == last_index && is_close_paren?(element)
25        if stack.pop == 0
26          infix_array[index] = nil
27          infix_array[0] = nil
28        end
29
30      else
31        stack.pop
32
33    end
34  end
```

Appendix B. Course Parsing Scripts

```
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 +
53         2])
54         infix_array[stack.pop] = nil
55         infix_array[index] = nil
56       else
57         stack.pop
58       end
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
```

Appendix B. Course Parsing Scripts

```
90 #remove all capitalized words
91 string = string.gsub(/\b[A-Z]{1}[a-z]\b/, ' ').gsub(/\s+/, ' ')
92
93 #remove ACT
94 string = string.gsub(/\bACT\s[0-9]?/, ' ').gsub(/\s+/, ' ')
95
96 #remove ISM
97 string = string.gsub(/\bISM\s[0-9]?/, ' ').gsub(/\s+/, ' ')
98
99 #remove SAT
100 string = string.gsub(/\bSAT\s[0-9]?/, ' ').gsub(/\s+/, ' ')
101
102 #remove stray 1 or 2 digit number
103 string = string.gsub(/\b[0-9]{1,2}\b/, ' ').gsub(/\s+/, ' ')
104
105 string = string.gsub(/\b[Oo][Rr]\b/, '+').gsub(/\b[Aa][Nn][Dd]\b/, '*')
106
107 array = string.split
108
109 array.delete_if { |element|
110   not(element.match(/[(\)*+]/)) and not(element.match(/[A-Z]{2,4}/)) and not(element.
111     match(/[0-9]{3}[A-Za-z]?/))
112 }
113
114 string = array.join(' ')
115 string = string.gsub("( )", ' ')
116 string = string.gsub(/\s+/, ' ')
117 # puts string
118
119 while string.match(/\s+([A-Z]{2,4} [0-9]{3}[A-Za-z]?)+\s+/)
120   string.gsub!(string.match(/\s+([A-Z]{2,4} [0-9]{3}[A-Za-z]?)+\s+/).to_s,
121     m[1])
122 end
123
124 array = string.split
125
126 while array.first == '+' or array.first == '*'
127   array.shift
128 end
129
130 while array.last == '+' or array.last == '*'
131   array.pop
132 end
133
134 final_array = Array.new
135 last_element = ""
136
137 array.each do |element|
138   if element.match(/[A-Z]{2,4}/)
139     last_element = element
140   elsif element.match(/[0-9]{3}[A-Za-z]?/)
141     if last_element.match(/[A-Z]{2,4}/)
142       final_array << "#{last_element}_#{element}"
143     else
144       final_array << "#{current_dept}_#{element}"
145     end
146   end
147 end
```

Appendix B. Course Parsing Scripts

```
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
15    string == "and" or string == "And" or string == "AND"
16 end
```

Appendix B. Course Parsing Scripts

```
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     else
63       prefix_eqn.push an_element
64
65     end
66   end
67 end
68
69 unless operator_stack.empty?
70   prefix_eqn += operator_stack.reverse
71 end
72
73
74 prefix_eqn.reverse
```

Appendix B. Course Parsing Scripts

```
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
90           {|e| e.strip!})
91
92         lines_of_file << line.gsub(/\[(.+)\]/, prefix_array.inspect)
93       else
94         lines_of_file << line
95       end
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

```
1  courses = Course.create([
2
3  {
4    number: "ECE 101",
5    title: "Intro to Elect & Computer Eng",
6    credits: 1,
7    description: "Insight into electrical engineering is gained through videos, hands-on
8                  experiments, use of computer software to learn basic problem-solving skills and
9                  a team-oriented design project.",
10   prereq_string: "",
11   coreq_string: "",
12   co_or_prereq_string: "",
13   semester_offering: "",
14   prereq_array: [],
15   coreq_array: [],
16   pre_or_coreq_array: []
17 },
18 {
19   number: "ECE 131",
20   title: "Programming Fundamentals",
21   credits: 3,
22   description: "Fundamental programming concepts, including consideration of abstract
23                 machine models with emphasis on the memory hierarchy, basic programming constructs
24                 , functions, parameter passing, pointers and arrays, file I/O, bit-level
25                 operations and interfacing to external devices.",
26   prereq_string: "",
27   coreq_string: "",
28   co_or_prereq_string: "",
29   semester_offering: "",
30   prereq_array: [],
```


Appendix C. Database Seed Files

```
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
36         's laws. Resistive networks, node and loop analysis. Network theorems. First-
37         order and second-order circuits. Sinusoidal sources and complex representations:
38         impedance, phasors, complex power. Three-phase circuits.",
39     prereq_string: "ECE 131 and MATH 163.",
40     coreq_string: "",
41     co_or_prereq_string: "MATH 316 and PHYC 161.",
42     semester_offering: "",
43     prereq_array: ["*", "ECE 131", "MATH 163"],
44     coreq_array: [],
45     pre_or_coreq_array: ["*", "MATH 316", "PHYC 161"]
46 },
47
48 {
49     number: "ECE 206L",
50     title: "Instrumentation",
51     credits: 2,
52     description: "Introduction to laboratory practices and the use of test equipment.
53         Measurements on basic electrical components, dc and ac circuits using ohmmeters,
54         voltmeters, ammeters and oscilloscopes. Circuit simulation.",
55     prereq_string: "203L and ENGL 102.",
56     coreq_string: "",
57     co_or_prereq_string: "",
58     semester_offering: "",
59     prereq_array: ["*", "ECE 203L", "ENGL 102"],
60     coreq_array: [],
61     pre_or_coreq_array: []
62 },
63
64 {
65     number: "ECE 213",
66     title: "Circuit Analysis II",
67     credits: 3,
68     description: "General transient analysis of electrical circuits. Laplace transform
69         with applications to circuit analysis. State-space equations. Fourier series
70         analysis. The network function; convolution; frequency response.",
71     prereq_string: "203L and MATH 316.",
72     coreq_string: "MATH 314.",
73     co_or_prereq_string: "",
74     semester_offering: "",
75     prereq_array: ["*", "ECE 203L", "MATH 316"],
76     coreq_array: ["MATH 314"],
77     pre_or_coreq_array: []
78 },
79
80 {
81     number: "ECE 231",
82     title: "Intermediate Programming",
83     credits: 3,
```

Appendix C. Database Seed Files

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

Appendix C. Database Seed Files

```
124     prereq_array: ["ECE 213"],
125     coreq_array: [],
126     pre_or_coreq_array: []
127 },
128
129 {
130     number: "ECE 322L",
131     title: "Electronics II",
132     credits: 4,
133     description: "Analysis, design, and characterization of linear circuits including
                  operational amplifiers. Design of biasing and reference circuits, multistage
                  amplifiers, and feedback circuits.",
134     prereq_string: "321L.",
135     coreq_string: "",
136     co_or_prereq_string: "",
137     semester_offering: "",
138     prereq_array: ["ECE 321L"],
139     coreq_array: [],
140     pre_or_coreq_array: []
141 },
142
143 {
144     number: "ECE 330",
145     title: "Software Design",
146     credits: 3,
147     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.",
148     prereq_string: "231.",
149     coreq_string: "",
150     co_or_prereq_string: "",
151     semester_offering: "",
152     prereq_array: ["ECE 231"],
153     coreq_array: [],
154     pre_or_coreq_array: []
155 },
156
157 {
158     number: "ECE 331",
159     title: "Data Structures & Algorithms",
160     credits: 3,
161     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.",
162     prereq_string: "231 and MATH 327.",
163     coreq_string: "340.",
164     co_or_prereq_string: "",
165     semester_offering: "",
166     prereq_array: ["*", "ECE 231", "MATH 327"],
167     coreq_array: ["ECE 340"],
168     pre_or_coreq_array: []
169 },
170
171 {
172     number: "ECE 335",
173     title: "Integrated Software Systems",
174     credits: 3,
```

Appendix C. Database Seed Files

```
175     description: "Course considers design principles, implementation issues, and
176         performance evaluation of various software paradigms in an integrated computing
177         environment. Topics include performance measurement and evaluation, program
178         optimization for the underlying architecture, integration and security for large
179         -scale software systems.",
180     prereq_string: "",
181     coreq_string: "",
182     co_or_prereq_string: "",
183     semester_offering: "",
184     prereq_array: [],
185     coreq_array: [],
186     pre_or_coreq_array: []
187 },
188 {
189     number: "ECE 337",
190     title: "Computer Arch & Organization",
191     credits: 3,
192     description: "Survey of various levels of computer architecture and design;
193         microprogramming and processor architecture, assembly language programming,
194         operating system concepts and input/output via the operating system. Three
195         lectures, 1 hr. lab.",
196     prereq_string: "231 and 238L.",
197     coreq_string: "",
198     co_or_prereq_string: "",
199     semester_offering: "Spring",
200     prereq_array: ["*", "ECE 231", "ECE 238L"],
201     coreq_array: [],
202     pre_or_coreq_array: []
203 },
204 {
205     number: "ECE 338",
206     title: "Intermediate Logic Design",
207     credits: 3,
208     description: "Advanced combinational circuits; XOR and transmission gates; computer-
209         based optimization methods; RTL and HDL; introduction to computer aided design;
210         advanced sequential machines; asynchronous sequential machines; timing issues;
211         memory and memory interfacing; programmable logic devices; and VLSI concepts.",
212     prereq_string: "238L.",
213     coreq_string: "",
214     co_or_prereq_string: "",
215     semester_offering: "",
216     prereq_array: ["ECE 238L"],
217     coreq_array: [],
218     pre_or_coreq_array: []
219 },
220 {
221     number: "ECE 340",
222     title: "Probabilistic Methods in Eng",
223     credits: 3,
224     description: "Introduction to probability, random variables, random processes,
225         probability distribution/density functions, expectation correlation, power
226         spectrum, WSS processes, confidence internals, transmission through LIT systems,
227         applications of probability.",
228     prereq_string: "314 and MATH 314.",
229     coreq_string: "",
```

Appendix C. Database Seed Files

```
220     co_or_prereq_string: "",
221     semester_offering: "",
222     prereq_array: ["*", "ECE 314", "MATH 314"],
223     coreq_array: [],
224     pre_or_coreq_array: []
225 },
226
227 {
228     number: "ECE 341",
229     title: "Communication Systems",
230     credits: 3,
231     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.",
232     prereq_string: "314 and 340.",
233     coreq_string: "",
234     co_or_prereq_string: "",
235     semester_offering: "",
236     prereq_array: ["*", "ECE 314", "ECE 340"],
237     coreq_array: [],
238     pre_or_coreq_array: []
239 },
240
241 {
242     number: "ECE 344L",
243     title: "Microprocessors",
244     credits: 4,
245     description: "Computers and Microprocessors: architecture, assembly language
                programming, input/output and applications. Three lectures, 3 hrs. lab.",
246     prereq_string: "206L and 238L and 321L.",
247     coreq_string: "",
248     co_or_prereq_string: "",
249     semester_offering: "Fall, Spring",
250     prereq_array: ["*", "*", "ECE 206L", "ECE 238L", "ECE 321L"],
251     coreq_array: [],
252     pre_or_coreq_array: []
253 },
254
255 {
256     number: "ECE 345",
257     title: "Intro to Control Systems",
258     credits: 3,
259     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.",
260     prereq_string: "314.",
261     coreq_string: "",
262     co_or_prereq_string: "",
263     semester_offering: "",
264     prereq_array: ["ECE 314"],
265     coreq_array: [],
266     pre_or_coreq_array: []
267 },
268
269 {
270     number: "ECE 360",
271     title: "Electromagnetic Fields & Waves",
```

Appendix C. Database Seed Files

```
272     credits: 3,
273     description: "Maxwell s equations , plane wave propagation , waveguides and
                transmission lines , transient pulse propagation and elementary dipole antenna.",
274     prereq_string: "213 and PHYC 161 and MATH 264.",
275     coreq_string: "",
276     co_or_prereq_string: "",
277     semester_offering: "",
278     prereq_array: ["*", "*", "ECE 213", "PHYC 161", "MATH 264"],
279     coreq_array: [],
280     pre_or_coreq_array: []
281 },
282
283 {
284     number: "ECE 371",
285     title: "Materials & Devices",
286     credits: 4,
287     description: "Introduction to quantum mechanics , crystal structures , insulators ,
                metals , and semiconductor material properties , bipolar , field effect and light
                emitting devices.",
288     prereq_string: "PHYC 262.",
289     coreq_string: "",
290     co_or_prereq_string: "",
291     semester_offering: "",
292     prereq_array: ["PHYC 262"],
293     coreq_array: [],
294     pre_or_coreq_array: []
295 },
296
297 {
298     number: "ECE 381",
299     title: " Intro to Power Systems",
300     credits: 3,
301     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.",
302     prereq_string: "213.",
303     coreq_string: "",
304     co_or_prereq_string: "",
305     semester_offering: "",
306     prereq_array: ["ECE 213"],
307     coreq_array: [],
308     pre_or_coreq_array: []
309 },
310
311 {
312     number: "ECE 413",
313     title: "Intro to Ray Graphics",
314     credits: 3,
315     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.",
316     prereq_string: "331 or CS 361L.",
317     coreq_string: "",
318     co_or_prereq_string: "",
319     semester_offering: "",
```

Appendix C. Database Seed Files

```
320     prereq_array: ["+", "ECE 331", "CS 361L"],
321     coreq_array: [],
322     pre_or_coreq_array: []
323 },
324
325 {
326     number: "ECE 419",
327     title: "Senior Design I",
328     credits: 3,
329     description: "Design methodology and development of professional project-oriented
330                 skills including communication, team management, economics, and engineering
331                 ethics. Working in teams, a proposal for a large design is prepared in response
332                 to an industrial or in-house sponsor. Restriction: ECE major, senior standing.",
333     prereq_string: "",
334     coreq_string: "",
335     co_or_prereq_string: "",
336     semester_offering: "",
337     prereq_array: [],
338     coreq_array: [],
339     pre_or_coreq_array: []
340 },
341
342 {
343     number: "ECE 420",
344     title: "Senior Design II",
345     credits: 3,
346     description: "Continuation of 419. Students work in assigned teams to implement
347                 proposal developed in 419. Prototypes are built and tested to sponsor
348                 specifications, and oral and written reports made to the project sponsor.",
349     prereq_string: "419.",
350     coreq_string: "",
351     co_or_prereq_string: "",
352     semester_offering: "",
353     prereq_array: ["ECE 419"],
354     coreq_array: [],
355     pre_or_coreq_array: []
356 },
357
358 {
359     number: "ECE 421",
360     title: "Analog Electronics",
361     credits: 3,
362     description: "Design of advanced analog electronic circuits. BJT and MOSFET
363                 operational amplifiers, current mirrors and output stages. Frequency response and
364                 compensation. Noise. A/D and D/A converters.",
365     prereq_string: "322L.",
366     coreq_string: "",
367     co_or_prereq_string: "",
368     semester_offering: "",
369     prereq_array: ["ECE 322L"],
370     coreq_array: [],
371     pre_or_coreq_array: []
372 },
373
374 {
375     number: "ECE 424",
376     title: "Digital VLSI Design",
377     credits: 3,
```

Appendix C. Database Seed Files

```
371     description: "CMOS logic gates and circuits, transistor implementations, applications
372         to sequential circuits, VLSI data path and controller design, VLSI routing
373         issues and architectures, RTL and VLSI impacts and applications to microprocessor
374         design.",
375     prereq_string: "321L and 338.",
376     coreq_string: "",
377     co_or_prereq_string: "",
378     semester_offering: "",
379     prereq_array: ["*", "ECE 321L", "ECE 338"],
380     coreq_array: [],
381     pre_or_coreq_array: []
382 },
383 {
384     number: "ECE 432",
385     title: "Intro to Parallel Processing",
386     credits: 3,
387     description: "(Also offered as CS 442.) Machine taxonomy and introduction to parallel
388         programming. Performance issues, speed-up and efficiency. Interconnection
389         networks and embeddings. Parallel programming issues and models: control parallel
390         , data parallel and data flow. Programming assignments on massively parallel
391         machines.",
392     prereq_string: "(331 or CS 351L) and (337 or CS 341L).",
393     coreq_string: "",
394     co_or_prereq_string: "",
395     semester_offering: "",
396     prereq_array: ["*", "+", "ECE 331", "CS 351L", "+", "ECE 337", "CS 341L"],
397     coreq_array: [],
398     pre_or_coreq_array: []
399 },
400 {
401     number: "ECE 435",
402     title: "Software Engineering",
403     credits: 3,
404     description: "Management and technical issues including business conduct and ethics
405         related to the design of large engineering projects. Student teams will address
406         the design, specification, implementation, testing and documentation of a large
407         hardware/software project.",
408     prereq_string: "331 and 335.",
409     coreq_string: "",
410     co_or_prereq_string: "",
411     semester_offering: "",
412     prereq_array: ["*", "ECE 331", "ECE 335"],
413     coreq_array: [],
414     pre_or_coreq_array: []
415 },
416 {
417     number: "ECE 437",
418     title: "Computer Operating Systems",
419     credits: 3,
420     description: "(Also offered as CS 481.) Fundamental principles of modern operating
421         systems design, with emphasis on concurrency and resource management. Topics
422         include processes, interprocess communication, semaphores, monitors, message
423         passing, input/output device, deadlocks memory management, files system design.",
424     prereq_string: "(330 and 337) or CS 341L.",
425     coreq_string: "",
```


Appendix C. Database Seed Files

```
416     co_or_prereq_string: "",
417     semester_offering: "",
418     prereq_array: ["+", "*", "ECE 330", "ECE 337", "CS 341L"],
419     coreq_array: [],
420     pre_or_coreq_array: []
421 },
422
423 {
424     number: "ECE 438",
425     title: "Design of Computers",
426     credits: 3,
427     description: "Computer architecture; design and implementation at HDL level; ALU,
428                 exception handling and interrupts; addressing; memory; speed issues; pipelining;
429                 microprogramming; introduction to distributed and parallel processing; buses; bus
430                 protocols and bus masters. CAD project to include written and oral presentations
431                 .",
432     prereq_string: "337 and 338 and 344L.",
433     coreq_string: "",
434     co_or_prereq_string: "",
435     semester_offering: "",
436     prereq_array: ["*", "*", "ECE 337", "ECE 338", "ECE 344L"],
437     coreq_array: [],
438     pre_or_coreq_array: []
439 },
440
441 {
442     number: "ECE 439",
443     title: "Intr Digital Signal Processing",
444     credits: 3,
445     description: "Bilateral Z transforms, region of convergence, review of sampling
446                 theorem, aliasing, the discrete Fourier transform and properties, analysis/design
447                 of FIR/IIR filters, FFT algorithms spectral analysis using FFT.",
448     prereq_string: "314.",
449     coreq_string: "",
450     co_or_prereq_string: "",
451     semester_offering: "",
452     prereq_array: ["ECE 314"],
453     coreq_array: [],
454     pre_or_coreq_array: []
455 },
456
457 {
458     number: "ECE 440",
459     title: "Computer Networks",
460     credits: 3,
461     description: "(Also offered as CS 485.) Theoretical and practical study of computer
462                 networks, including network structures and architectures. Principles of digital
463                 communications systems. Network topologies, protocols and services. TCP/IP
464                 protocol suite. Point-to-point networks; broadcast networks; local area networks;
465                 routing, error and flow control techniques.",
466     prereq_string: "330 and 337.",
467     coreq_string: "340.",
468     co_or_prereq_string: "",
469     semester_offering: "",
470     prereq_array: ["*", "ECE 330", "ECE 337"],
471     coreq_array: ["ECE 340"],
472     pre_or_coreq_array: []
473 },
```

Appendix C. Database Seed Files

```
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
```

Appendix C. Database Seed Files

```
function design for non-linear systems, and laboratory design project.",
512 prereq_string: "345.",
513 coreq_string: "",
514 co_or_prereq_string: "",
515 semester_offering: "",
516 prereq_array: ["ECE 345"],
517 coreq_array: [],
518 pre_or_coreq_array: []
519 },
520
521 {
522 number: "ECE 448",
523 title: "Fuzzy Logic with Applications",
524 credits: 3,
525 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.",
526 prereq_string: "",
527 coreq_string: "",
528 co_or_prereq_string: "",
529 semester_offering: "",
530 prereq_array: [],
531 coreq_array: [],
532 pre_or_coreq_array: []
533 },
534
535 {
536 number: "ECE 456",
537 title: "Entrepreneurial Engineering",
538 credits: 3,
539 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.",
540 prereq_string: "",
541 coreq_string: "",
542 co_or_prereq_string: "",
543 semester_offering: "",
544 prereq_array: [],
545 coreq_array: [],
546 pre_or_coreq_array: []
547 },
548
549 {
550 number: "ECE 460",
551 title: "Microwave Engineering",
552 credits: 3,
553 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.",
554 prereq_string: "360.",
555 coreq_string: "",
556 co_or_prereq_string: "",
557 semester_offering: "",
558 prereq_array: ["ECE 360"],
```

Appendix C. Database Seed Files

```
559     coreq_array: [],
560     pre_or_coreq_array: []
561   },
562
563   {
564     number: "ECE 463",
565     title: "Advanced Optics I",
566     credits: 3,
567     description: "(Also offered as PHYC 463.) Electromagnetic theory of geometrical
                    optics, Gaussian ray tracing and matrix methods, finite ray tracing, aberrations,
                    interference and diffraction.",
568     prereq_string: "PHYC 302.",
569     coreq_string: "",
570     co_or_prereq_string: "",
571     semester_offering: "",
572     prereq_array: ["PHYC 302"],
573     coreq_array: [],
574     pre_or_coreq_array: []
575   },
576
577   {
578     number: "ECE 464",
579     title: "Laser Physics I",
580     credits: 3,
581     description: "(Also offered as PHYC 464.) Resonator optics. Rate equations;
                    spontaneous and stimulated emission; gas, semiconductor and solid state lasers,
                    pulsed and mode-locked laser techniques.",
582     prereq_string: "360 or PHYC 406.",
583     coreq_string: "",
584     co_or_prereq_string: "",
585     semester_offering: "",
586     prereq_array: ["+", "ECE 360", "PHYC 406"],
587     coreq_array: [],
588     pre_or_coreq_array: []
589   },
590
591   {
592     number: "ECE 469",
593     title: "Antennas for Wireless Comm",
594     credits: 3,
595     description: "Aspects of antenna theory and design; radiation from dipoles, loops,
                    apertures, microstrip antennas and antenna arrays.",
596     prereq_string: "360.",
597     coreq_string: "",
598     co_or_prereq_string: "",
599     semester_offering: "",
600     prereq_array: ["ECE 360"],
601     coreq_array: [],
602     pre_or_coreq_array: []
603   },
604
605   {
606     number: "ECE 471",
607     title: "Materials & Devices II",
608     credits: 3,
609     description: "An intermediate study of semiconductor materials, energy band structure
                    , p-n junctions, ideal and non-ideal effects in field effect and bipolar
                    transistors.",
```

Appendix C. Database Seed Files

```
610     prereq_string: "360 and 371.",
611     coreq_string: "",
612     co_or_prereq_string: "",
613     semester_offering: "",
614     prereq_array: ["*", "ECE 360", "ECE 371"],
615     coreq_array: [],
616     pre_or_coreq_array: []
617 },
618
619 {
620     number: "ECE 474L",
621     title: "Microelectronics Processing I",
622     credits: 3,
623     description: "(Also offered as NSMS 574L.) Materials science of semiconductors,
624                 microelectronics technologies, device/circuit fabrication, parasitics and
625                 packaging. Lab project features small group design/fabrication/testing of MOS
626                 circuits.",
627     prereq_string: "",
628     coreq_string: "",
629     co_or_prereq_string: "",
630     semester_offering: "",
631     prereq_array: [],
632     coreq_array: [],
633     pre_or_coreq_array: []
634 },
635
636 {
637     number: "ECE 475",
638     title: "Optoelectronics",
639     credits: 3,
640     description: "Basic electro-optics and opto-electronics, with engineering
641                 applications. Interaction of light with matter. Introduction to optics of
642                 dielectrics, metals and crystals. Introductory descriptions of electro-optic,
643                 acousto-optic and magneto-optic effects and related devices. Light sources,
644                 displays and detectors. Elementary theory and applications of lasers, optical
645                 waveguides and fibers.",
646     prereq_string: "371.",
647     coreq_string: "",
648     co_or_prereq_string: "",
649     semester_offering: "",
650     prereq_array: ["ECE 371"],
651     coreq_array: [],
652     pre_or_coreq_array: []
653 },
654
655 {
656     number: "ECE 482",
657     title: "Electric Drives & Transformers",
658     credits: 3,
659     description: "Electromagnetic theory and mechanical considerations are employed to
660                 develop models for and understanding of Transformers, Induction Machines and
661                 Synchronous Machines. Additionally, DC Machines are discussed.",
662     prereq_string: "203 and 213.",
663     coreq_string: "",
664     co_or_prereq_string: "",
665     semester_offering: "",
666     prereq_array: ["*", "ECE 203", "ECE 213"],
667     coreq_array: [],
```

Appendix C. Database Seed Files

```
658     pre_or_coreq_array: []
659   },
660
661   {
662     number: "ECE 483",
663     title: "Power Electronics",
664     credits: 3,
665     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. Prerequisite: 321L and 371 and 381.",
666     prereq_string: "",
667     coreq_string: "",
668     co_or_prereq_string: "",
669     semester_offering: "",
670     prereq_array: [],
671     coreq_array: [],
672     pre_or_coreq_array: []
673   },
674
675   {
676     number: "ECE 484",
677     title: "Photovoltaics",
678     credits: 3,
679     description: "Technical concepts of photovoltaics. Solar cell device level operation,
        packaging, manufacturing, designing photovoltaic system for stand-alone or grid-
        tied operation, some business-case analysis and some real-life scenarios of
        applicability of these solutions.",
680     prereq_string: "381 and MATH 121.",
681     coreq_string: "",
682     co_or_prereq_string: "",
683     semester_offering: "",
684     prereq_array: ["*", "ECE 381", "MATH 121"],
685     coreq_array: [],
686     pre_or_coreq_array: []
687   },
688
689   {
690     number: "ECE 488",
691     title: "Future Energy Systems",
692     credits: 3,
693     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."
694     ,
695     prereq_string: "381 and 482 and 483 and 484.",
696     coreq_string: "",
697     co_or_prereq_string: "",
698     semester_offering: "",
699     prereq_array: ["*", "*", "*", "ECE 381", "ECE 482", "ECE 483", "ECE 484"],
700     coreq_array: [],
701     pre_or_coreq_array: []
702   },
703
704   {
705     number: "ECE 490",
706     title: "Internship",
707     credits: 3,
```

Appendix C. Database Seed Files

```
707     description: "Professional practice under the guidance of a practicing engineer.
708         Assignments include design or analysis of systems or hardware, or computer
709         programming. A preliminary proposal and periodic reports are required. The
710         engineer evaluates student s work; a faculty monitor assigns grade. Restriction:
711         ECE major, junior standing. (12 hours/week) (24 hours/week in summer session).
712         Offered on a CR/NC basis only.",
713     prereq_string: "",
714     coreq_string: "",
715     co_or_prereq_string: "",
716     semester_offering: "",
717     prereq_array: [],
718     coreq_array: [],
719     pre_or_coreq_array: []
720 },
721 {
722     number: "ECE 491",
723     title: "Undergrad Problems",
724     credits: 6,
725     description: "Registration for more than 3 hours requires permission of department
726     chairperson.",
727     prereq_string: "",
728     coreq_string: "",
729     co_or_prereq_string: "",
730     semester_offering: "",
731     prereq_array: [],
732     coreq_array: [],
733     pre_or_coreq_array: []
734 },
735 {
736     number: "ECE 493",
737     title: "Honors Seminar",
738     credits: 3,
739     description: "A special seminar open only to honors students. Registration requires
740     permission of department chairperson.",
741     prereq_string: "",
742     coreq_string: "",
743     co_or_prereq_string: "",
744     semester_offering: "",
745     prereq_array: [],
746     coreq_array: [],
747     pre_or_coreq_array: []
748 },
749 {
750     number: "ECE 494",
751     title: "Honors Individual Study",
752     credits: 6,
753     description: "Open only to honors students. Registration requires permission of the
754     department chairperson and of the supervising professor.",
755     prereq_string: "",
756     coreq_string: "",
757     co_or_prereq_string: "",
758     semester_offering: "",
759     prereq_array: [],
760     coreq_array: [],
761     pre_or_coreq_array: []
```

Appendix C. Database Seed Files

```
757 },
758
759 {
760   number: "ECE 495",
761   title: "Special Topics",
762   credits: 4,
763   description: "Restriction: ECE major, senior standing.",
764   prereq_string: "",
765   coreq_string: "",
766   co_or_prereq_string: "",
767   semester_offering: "",
768   prereq_array: [],
769   coreq_array: [],
770   pre_or_coreq_array: []
771 },
772
773 {
774   number: "ECE 500",
775   title: "Theory of Linear Systems",
776   credits: 3,
777   description: "State space representation of dynamical systems. Analysis and design of
778               linear models in control systems and signal processing. Continuous, discrete and
779               sampled representations. This course is fundamental for students in the system
780               areas.",
781   prereq_string: "",
782   coreq_string: "",
783   co_or_prereq_string: "",
784   semester_offering: "",
785   prereq_array: [],
786   coreq_array: [],
787   pre_or_coreq_array: []
788 },
789
790 {
791   number: "ECE 506",
792   title: "Optimization Theory",
793   credits: 3,
794   description: "Introduction to the topic of optimization by the computer. Linear and
795               nonlinear programming. The simplex method, Karmakar method, gradient, conjugate
796               gradient and quasi-Newton methods, Fibonacci/Golden search, Quadratic and Cubic
797               fitting methods, Penalty and Barrier methods.",
798   prereq_string: "",
799   coreq_string: "",
800   co_or_prereq_string: "",
801   semester_offering: "",
802   prereq_array: [],
803   coreq_array: [],
804   pre_or_coreq_array: []
805 },
806
807 {
808   number: "ECE 510",
809   title: "Medical Imaging",
810   credits: 3,
811   description: "This course will introduce the student to medical imaging modalities (e
812               .g. MRI, Nuclear Imagine, Ultrasound) with an emphasis on a signals and systems
813               approach. Topics include hardware, signal formation, image reconstruction, and
814               application.",
```


Appendix C. Database Seed Files

```
806     prereq_string: "",
807     coreq_string: "",
808     co_or_prereq_string: "",
809     semester_offering: "",
810     prereq_array: [],
811     coreq_array: [],
812     pre_or_coreq_array: []
813 },
814
815 {
816     number: "ECE 511",
817     title: "fMRI Analysis Methods",
818     credits: 3,
819     description: "This course will be an introduction to signal and image processing
820                 methods for functional magnetic resonance imaging (fMRI) of the brain.",
821     prereq_string: "",
822     coreq_string: "",
823     co_or_prereq_string: "",
824     semester_offering: "",
825     prereq_array: [],
826     coreq_array: [],
827     pre_or_coreq_array: []
828 },
829 {
830     number: "ECE 512",
831     title: "Advanced Image Synthesis",
832     credits: 3,
833     description: "Course covers image synthesis techniques from perspective of high-end
834                 scanline rendering including physically-based rendering algorithms. Topics:
835                 radiometry, stachastic ray tracing, variance reduction, photon mapping,
836                 reflection models, participating media, advanced algorithms for light transport."
837     ,
838     prereq_string: "",
839     coreq_string: "",
840     co_or_prereq_string: "",
841     semester_offering: "",
842     prereq_array: [],
843     coreq_array: [],
844     pre_or_coreq_array: []
845 },
846 {
847     number: "ECE 513",
848     title: "Real-time Graphics",
849     credits: 3,
850     description: "Course covers advanced algorithms in real-time rendering and graphics
851                 hardware, bringing students up to speed with cutting edge real-time graphics.
852                 Topics: advanced GPU algorithms for graphics and non-graphics applications. Term
853                 project required.",
854     prereq_string: "",
855     coreq_string: "",
856     co_or_prereq_string: "",
857     semester_offering: "",
858     prereq_array: [],
859     coreq_array: [],
860     pre_or_coreq_array: []
861 },
```

Appendix C. Database Seed Files

```
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
```

Appendix C. Database Seed Files

```
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.",
904   prereq_string: "510.",
905   coreq_string: "",
906   co_or_prereq_string: "",
907   semester_offering: "Spring",
908   prereq_array: ["ECE 510"],
909   coreq_array: [],
910   pre_or_coreq_array: []
911 },
912
913 {
914   number: "ECE 519",
915   title: "MEMS Transducer Devices & Tech",
916   credits: 3,
917   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.",
918   prereq_string: "",
919   coreq_string: "",
920   co_or_prereq_string: "",
921   semester_offering: "",
922   prereq_array: [],
923   coreq_array: [],
924   pre_or_coreq_array: []
925 },
926
927 {
928   number: "ECE 520",
929   title: "VLSI Design",
930   credits: 3,
931   description: "Advanced topics include: 1C technologies, CAD tools, gate arrays,
standard cells and full custom designs. Design of memories, PLA, I/O and random
logic circuit. Design for testability.",
932   prereq_string: "",
933   coreq_string: "",
934   co_or_prereq_string: "",
935   semester_offering: "",
936   prereq_array: [],
937   coreq_array: [],
938   pre_or_coreq_array: []
939 },
940
941 {
942   number: "ECE 522",
943   title: " Hrdwr Sftwr Codesign FPGA's",
944   credits: 3,
945   description: "This course provides an introduction to the design of electronic
systems that incorporate both hardware and software components.",
946   prereq_string: "433.",
947   coreq_string: "",
948   co_or_prereq_string: "",
949   semester_offering: "",
950   prereq_array: ["ECE 433"],
951   coreq_array: [],
952   pre_or_coreq_array: []
```

Appendix C. Database Seed Files

```
953 },
954
955 {
956   number: "ECE 523",
957   title: "Analog Electronics",
958   credits: 3,
959   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.",
960   prereq_string: "",
961   coreq_string: "",
962   co_or_prereq_string: "",
963   semester_offering: "",
964   prereq_array: [],
965   coreq_array: [],
966   pre_or_coreq_array: []
967 },
968
969 {
970   number: "ECE 524",
971   title: "Collaborative Interdisc Teach",
972   credits: 3,
973   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.",
974   prereq_string: "",
975   coreq_string: "",
976   co_or_prereq_string: "",
977   semester_offering: "",
978   prereq_array: [],
979   coreq_array: [],
980   pre_or_coreq_array: []
981 },
982
983 {
984   number: "ECE 525",
985   title: "Microelectronics Test Eng",
986   credits: 3,
987   description: "This course provides an introduction to hardware security and trust
primitives and their application to secure and trustworthy hardware systems.",
988   prereq_string: "",
989   coreq_string: "",
990   co_or_prereq_string: "",
991   semester_offering: "",
992   prereq_array: [],
993   coreq_array: [],
994   pre_or_coreq_array: []
995 },
996
997 {
998   number: "ECE 528",
999   title: "Embedded Systems Architecture",
1000   credits: 3,
1001   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
```

Appendix C. Database Seed Files

```
        programmable logic systems.",
1002     prereq_string: "",
1003     coreq_string: "",
1004     co_or_prereq_string: "",
1005     semester_offering: "",
1006     prereq_array: [],
1007     coreq_array: [],
1008     pre_or_coreq_array: []
1009 },
1010
1011 {
1012     number: "ECE 533",
1013     title: "Digital Image Processing",
1014     credits: 3,
1015     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.",
1016     prereq_string: "",
1017     coreq_string: "",
1018     co_or_prereq_string: "",
1019     semester_offering: "",
1020     prereq_array: [],
1021     coreq_array: [],
1022     pre_or_coreq_array: []
1023 },
1024
1025 {
1026     number: "ECE 534",
1027     title: "Plasma Physics I",
1028     credits: 3,
1029     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.",
1030     prereq_string: "",
1031     coreq_string: "",
1032     co_or_prereq_string: "",
1033     semester_offering: "",
1034     prereq_array: [],
1035     coreq_array: [],
1036     pre_or_coreq_array: []
1037 },
1038
1039 {
1040     number: "ECE 536",
1041     title: "Computer System Software",
1042     credits: 3,
1043     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.",
1044     prereq_string: "",
1045     coreq_string: "",
1046     co_or_prereq_string: "",
1047     semester_offering: "",
```

Appendix C. Database Seed Files

```
1048     prereq_array: [],
1049     coreq_array: [],
1050     pre_or_coreq_array: []
1051 },
1052
1053 {
1054     number: "ECE 537",
1055     title: "Foundations of Computing",
1056     credits: 3,
1057     description: "Computational aspects of engineering problems. Topics include machine
1058                 models and computability, classification and performance analysis of algorithms,
1059                 advanced data structures, approximation algorithms, introduction to complexity
1060                 theory and complexity classes.",
1061     prereq_string: "",
1062     coreq_string: "",
1063     co_or_prereq_string: "",
1064     semester_offering: "",
1065     prereq_array: [],
1066     coreq_array: [],
1067     pre_or_coreq_array: []
1068 },
1069
1070 {
1071     number: "ECE 538",
1072     title: "Advanced Computer Architecture",
1073     credits: 3,
1074     description: "Course provides an in-depth analysis of computer architecture
1075                 techniques. Topics include high speed computing techniques, memory systems,
1076                 pipelining, vector machines, parallel processing, multiprocessor systems, high-
1077                 level language machines and data flow computers.",
1078     prereq_string: "",
1079     coreq_string: "",
1080     co_or_prereq_string: "",
1081     semester_offering: "",
1082     prereq_array: [],
1083     coreq_array: [],
1084     pre_or_coreq_array: []
1085 },
1086
1087 {
1088     number: "ECE 539",
1089     title: "Digital Signal Processing I",
1090     credits: 3,
1091     description: "Hilbert spaces, orthogonal basis, generalized sampling theorem,
1092                 multirate systems, filterbanks, quantization, structures for LTI systems, finite
1093                 word-length effects, linear prediction, min/max phase systems, multiresolution
1094                 signal analysis.",
1095     prereq_string: "",
1096     coreq_string: "",
1097     co_or_prereq_string: "",
1098     semester_offering: "",
1099     prereq_array: [],
1100     coreq_array: [],
1101     pre_or_coreq_array: []
1102 },
1103
1104 {
1105     number: "ECE 540",
```

Appendix C. Database Seed Files

```
1097     title: "Advanced Networking",
1098     credits: 3,
1099     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.",
1100     prereq_string: "",
1101     coreq_string: "",
1102     co_or_prereq_string: "",
1103     semester_offering: "",
1104     prereq_array: [],
1105     coreq_array: [],
1106     pre_or_coreq_array: []
1107 },
1108
1109 {
1110     number: "ECE 541",
1111     title: "Probab and Stochastic Process",
1112     credits: 3,
1113     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.",
1114     prereq_string: "",
1115     coreq_string: "",
1116     co_or_prereq_string: "",
1117     semester_offering: "",
1118     prereq_array: [],
1119     coreq_array: [],
1120     pre_or_coreq_array: []
1121 },
1122
1123 {
1124     number: "ECE 542",
1125     title: "Digital Communication Theory",
1126     credits: 3,
1127     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.",
1128     prereq_string: "541.",
1129     coreq_string: "",
1130     co_or_prereq_string: "",
1131     semester_offering: "",
1132     prereq_array: ["ECE 541"],
1133     coreq_array: [],
1134     pre_or_coreq_array: []
1135 },
1136
1137 {
1138     number: "ECE 546",
1139     title: "Multivariable Control Theory",
1140     credits: 3,
1141     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
```

Appendix C. Database Seed Files

```
        approach to feedback system design. Optimal linear-quadratic-Gaussian (LQG)
        control. Multivariable Nyquist stability criteria.",
1142     prereq_string: "500.",
1143     coreq_string: "",
1144     co_or_prereq_string: "",
1145     semester_offering: "",
1146     prereq_array: ["ECE 500"],
1147     coreq_array: [],
1148     pre_or_coreq_array: []
1149 },
1150
1151 {
1152     number: "ECE 547",
1153     title: "Neural Networks",
1154     credits: 3,
1155     description: "(Also offered as CS 547.) A study of biological and artificial neuron
        models, basic neural architectures and parallel and distributed processing.",
1156     prereq_string: "",
1157     coreq_string: "",
1158     co_or_prereq_string: "",
1159     semester_offering: "",
1160     prereq_array: [],
1161     coreq_array: [],
1162     pre_or_coreq_array: []
1163 },
1164
1165 {
1166     number: "ECE 548",
1167     title: "Fuzzy Logic with Applications",
1168     credits: 3,
1169     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.",
1170     prereq_string: "",
1171     coreq_string: "",
1172     co_or_prereq_string: "",
1173     semester_offering: "",
1174     prereq_array: [],
1175     coreq_array: [],
1176     pre_or_coreq_array: []
1177 },
1178
1179 {
1180     number: "ECE 549",
1181     title: "Inform Thry&Coding",
1182     credits: 3,
1183     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.",
1184     prereq_string: "340 or equivalent.",
1185     coreq_string: "",
1186     co_or_prereq_string: "",
1187     semester_offering: "",
1188     prereq_array: ["ECE 340"],
1189     coreq_array: [],
1190     pre_or_coreq_array: []
```


Appendix C. Database Seed Files

```
1191 },
1192
1193 {
1194   number: "ECE 550",
1195   title: "Soc & Eth Iss Nanotechnology",
1196   credits: 3,
1197   description: "(Also offered as CHNE, NSMS 550.) In this course, students will examine
      issues arising from this emerging technology, including those of privacy,
      health and safety, the environment, public perception and human enhancement.",
1198   prereq_string: "",
1199   coreq_string: "",
1200   co_or_prereq_string: "",
1201   semester_offering: "",
1202   prereq_array: [],
1203   coreq_array: [],
1204   pre_or_coreq_array: []
1205 },
1206
1207 {
1208   number: "ECE 551",
1209   title: "Problems",
1210   credits: 6,
1211   description: "",
1212   prereq_string: "",
1213   coreq_string: "",
1214   co_or_prereq_string: "",
1215   semester_offering: "",
1216   prereq_array: [],
1217   coreq_array: [],
1218   pre_or_coreq_array: []
1219 },
1220
1221 {
1222   number: "ECE 553L",
1223   title: "Exp Techniques Plasma Physics",
1224   credits: 3,
1225   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.",
1226   prereq_string: "534.",
1227   coreq_string: "",
1228   co_or_prereq_string: "",
1229   semester_offering: "",
1230   prereq_array: ["ECE 534"],
1231   coreq_array: [],
1232   pre_or_coreq_array: []
1233 },
1234
1235 {
1236   number: "ECE 554",
1237   title: "Advanced Optics II",
1238   credits: 3,
1239   description: "(Also offered as PHYC 554.) Diffractions theory, coherence theory,
      coherent objects, and incoherent imaging, and polarization.",
1240   prereq_string: "",
1241   coreq_string: "",
1242   co_or_prereq_string: "",
```

Appendix C. Database Seed Files

```
1243     semester_offering: "",
1244     prereq_array: [],
1245     coreq_array: [],
1246     pre_or_coreq_array: []
1247 },
1248
1249 {
1250     number: "ECE 555",
1251     title: "Foundations of Engineering EM",
1252     credits: 3,
1253     description: "Mathematical foundations for engineering electromagnetics: linear
1254                 analysis and method of moments, complex analysis and Kramers-Kronig relations,
1255                 Green's functions, spectral representation method and electromagnetic sources.",
1256     prereq_string: "",
1257     coreq_string: "",
1258     co_or_prereq_string: "",
1259     semester_offering: "",
1260     prereq_array: [],
1261     coreq_array: [],
1262     pre_or_coreq_array: []
1263 },
1264
1265 {
1266     number: "ECE 556",
1267     title: "Entrepreneurial Engineering",
1268     credits: 3,
1269     description: "(Also offered as ME 556.) Review and application of necessary elements
1270                 for successfully launching technical businesses; focuses upon technology,
1271                 manufacturing, management, marketing, legal and financial aspects. Students work
1272                 in groups developing elements of new businesses and producing business plans.",
1273     prereq_string: "",
1274     coreq_string: "",
1275     co_or_prereq_string: "",
1276     semester_offering: "",
1277     prereq_array: [],
1278     coreq_array: [],
1279     pre_or_coreq_array: []
1280 },
1281
1282 {
1283     number: "ECE 557",
1284     title: "Pulsed Pwr & Char Partic Accel",
1285     credits: 3,
1286     description: "Principles of pulsed power circuits, components, systems and their
1287                 relationship to charged particle acceleration and transport. Energy storage,
1288                 voltage multiplication, pulse shaping, insulation and breakdown and switching.
1289                 Single particle dynamics and accelerator configurations.",
1290     prereq_string: "",
1291     coreq_string: "",
1292     co_or_prereq_string: "",
1293     semester_offering: "",
1294     prereq_array: [],
1295     coreq_array: [],
1296     pre_or_coreq_array: []
1297 },
1298
1299 {
1300     number: "ECE 558",
```

Appendix C. Database Seed Files

```
1293     title: "Charged Partical Beams",
1294     credits: 3,
1295     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.",
1296     prereq_string: "557 and CHNE 545.",
1297     coreq_string: "",
1298     co_or_prereq_string: "",
1299     semester_offering: "",
1300     prereq_array: ["*", "ECE 557", "CHNE 545"],
1301     coreq_array: [],
1302     pre_or_coreq_array: []
1303 },
1304
1305 {
1306     number: "ECE 559",
1307     title: "Intern: Optical Science & Eng",
1308     credits: 3,
1309     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.",
1310     prereq_string: "",
1311     coreq_string: "",
1312     co_or_prereq_string: "",
1313     semester_offering: "",
1314     prereq_array: [],
1315     coreq_array: [],
1316     pre_or_coreq_array: []
1317 },
1318
1319 {
1320     number: "ECE 560",
1321     title: "Microwave Engineering",
1322     credits: 3,
1323     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.",
1324     prereq_string: "",
1325     coreq_string: "",
1326     co_or_prereq_string: "",
1327     semester_offering: "",
1328     prereq_array: [],
1329     coreq_array: [],
1330     pre_or_coreq_array: []
1331 },
1332
1333 {
1334     number: "ECE 561",
1335     title: "Electrodynamics",
1336     credits: 3,
1337     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.",
1338     prereq_string: "555.",
```

Appendix C. Database Seed Files

```
1339     coreq-string: "",
1340     co_or_prereq-string: "",
1341     semester_offering: "",
1342     prereq_array: ["ECE 555"],
1343     coreq_array: [],
1344     pre_or_coreq_array: []
1345 },
1346
1347 {
1348     number: "ECE 562",
1349     title: "Electronics RF Design",
1350     credits: 3,
1351     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.",
1352     prereq-string: "",
1353     coreq-string: "",
1354     co_or_prereq-string: "",
1355     semester_offering: "",
1356     prereq_array: [],
1357     coreq_array: [],
1358     pre_or_coreq_array: []
1359 },
1360
1361 {
1362     number: "ECE 563",
1363     title: "Comp Methods Electromagnetics",
1364     credits: 3,
1365     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.",
1366     prereq-string: "561.",
1367     coreq-string: "",
1368     co_or_prereq-string: "",
1369     semester_offering: "",
1370     prereq_array: ["ECE 561"],
1371     coreq_array: [],
1372     pre_or_coreq_array: []
1373 },
1374
1375 {
1376     number: "ECE 564",
1377     title: "Guided Wave Optics",
1378     credits: 3,
1379     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.",
1380     prereq-string: "",
1381     coreq-string: "",
1382     co_or_prereq-string: "",
1383     semester_offering: "",
1384     prereq_array: [],
1385     coreq_array: [],
1386     pre_or_coreq_array: []
1387 },
```

Appendix C. Database Seed Files

```
1388
1389 {
1390     number: "ECE 565",
1391     title: "Optical Comm Components Subsys",
1392     credits: 3,
1393     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.",
1394     prereq_string: "",
1395     coreq_string: "",
1396     co_or_prereq_string: "",
1397     semester_offering: "",
1398     prereq_array: [],
1399     coreq_array: [],
1400     pre_or_coreq_array: []
1401 },
1402
1403 {
1404     number: "ECE 566",
1405     title: "Advanced Optical Networks",
1406     credits: 3,
1407     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.",
1408     prereq_string: "565.",
1409     coreq_string: "",
1410     co_or_prereq_string: "",
1411     semester_offering: "",
1412     prereq_array: ["ECE 565"],
1413     coreq_array: [],
1414     pre_or_coreq_array: []
1415 },
1416
1417 {
1418     number: "ECE 569",
1419     title: "Antennas for Wireless Comm",
1420     credits: 3,
1421     description: "Aspects of antenna theory and design; radiation from dipoles, loops,
        apertures, microstrip antennas and antenna arrays.",
1422     prereq_string: "",
1423     coreq_string: "",
1424     co_or_prereq_string: "",
1425     semester_offering: "",
1426     prereq_array: [],
1427     coreq_array: [],
1428     pre_or_coreq_array: []
1429 },
1430
1431 {
1432     number: "ECE 570",
```

Appendix C. Database Seed Files

```
1433     title: "Semicon Materials & Devices",
1434     credits: 3,
1435     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).",
1436     prereq_string: "471 or 572.",
1437     coreq_string: "",
1438     co_or_prereq_string: "",
1439     semester_offering: "",
1440     prereq_array: ["+", "ECE 471", "ECE 572"],
1441     coreq_array: [],
1442     pre_or_coreq_array: []
1443 },
1444
1445 {
1446     number: "ECE 572",
1447     title: "Physics of Semiconductors",
1448     credits: 3,
1449     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.",
1450     prereq_string: "471.",
1451     coreq_string: "",
1452     co_or_prereq_string: "",
1453     semester_offering: "",
1454     prereq_array: ["ECE 471"],
1455     coreq_array: [],
1456     pre_or_coreq_array: []
1457 },
1458
1459 {
1460     number: "ECE 574L",
1461     title: "Microelectronics Processing I",
1462     credits: 3,
1463     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.",
1464     prereq_string: "",
1465     coreq_string: "",
1466     co_or_prereq_string: "",
1467     semester_offering: "",
1468     prereq_array: [],
1469     coreq_array: [],
1470     pre_or_coreq_array: []
1471 },
1472
1473 {
1474     number: "ECE 576",
1475     title: "Modern VLSI Devices",
1476     credits: 3,
1477     description: "Review of the evolution of VLSI technology and basic device physics.
                  Detailed analysis of MOSFET devices, CMOS device design including device scaling
                  concepts.",
```

Appendix C. Database Seed Files

```
1478     prereq_string: "471 or 572.",
1479     coreq_string: "",
1480     co_or_prereq_string: "",
1481     semester_offering: "",
1482     prereq_array: ["+", "ECE 471", "ECE 572"],
1483     coreq_array: [],
1484     pre_or_coreq_array: []
1485 },
1486
1487 {
1488     number: "ECE 577",
1489     title: "Fundmts of Semic Lasers & LEDs",
1490     credits: 3,
1491     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.",
1492     prereq_string: "572.",
1493     coreq_string: "",
1494     co_or_prereq_string: "",
1495     semester_offering: "",
1496     prereq_array: ["ECE 572"],
1497     coreq_array: [],
1498     pre_or_coreq_array: []
1499 },
1500
1501 {
1502     number: "ECE 578",
1503     title: "Advanced Semiconductor Lasers",
1504     credits: 3,
1505     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.",
1506     prereq_string: "577.",
1507     coreq_string: "",
1508     co_or_prereq_string: "",
1509     semester_offering: "",
1510     prereq_array: ["ECE 577"],
1511     coreq_array: [],
1512     pre_or_coreq_array: []
1513 },
1514
1515 {
1516     number: "ECE 580",
1517     title: "Advanced Plasma Physics",
1518     credits: 3,
1519     description: "(Also offered as PHYC 580, CHNE 580.)",
1520     prereq_string: "534 or PHYC 534.",
1521     coreq_string: "",
1522     co_or_prereq_string: "",
1523     semester_offering: "",
1524     prereq_array: ["+", "ECE 534", "PHYC 534"],
1525     coreq_array: [],
1526     pre_or_coreq_array: []
1527 },
1528
```

Appendix C. Database Seed Files

```
1529 {
1530   number: "ECE 581",
1531   title: "Coll Nanocrystal Biomed Appl",
1532   credits: 3,
1533   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.",
1534   prereq_string: "",
1535   coreq_string: "",
1536   co_or_prereq_string: "",
1537   semester_offering: "",
1538   prereq_array: [],
1539   coreq_array: [],
1540   pre_or_coreq_array: []
1541 },
1542
1543 {
1544   number: "ECE 582",
1545   title: "Electric Drives & Transformers",
1546   credits: 3,
1547   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.",
1548   prereq_string: "",
1549   coreq_string: "",
1550   co_or_prereq_string: "",
1551   semester_offering: "",
1552   prereq_array: [],
1553   coreq_array: [],
1554   pre_or_coreq_array: []
1555 },
1556
1557 {
1558   number: "ECE 583",
1559   title: "Power Electronics",
1560   credits: 3,
1561   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.",
1562   prereq_string: "",
1563   coreq_string: "",
1564   co_or_prereq_string: "",
1565   semester_offering: "",
1566   prereq_array: [],
1567   coreq_array: [],
1568   pre_or_coreq_array: []
1569 },
1570
1571 {
1572   number: "ECE 584",
1573   title: "Photovoltaics",
1574   credits: 3,
1575   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
```


Appendix C. Database Seed Files

```
        applicability of these solutions.",
1576     prereq_string: "",
1577     coreq_string: "",
1578     co_or_prereq_string: "",
1579     semester_offering: "",
1580     prereq_array: [],
1581     coreq_array: [],
1582     pre_or_coreq_array: []
1583 },
1584
1585 {
1586     number: "ECE 585",
1587     title: "Modern Manufacturing Methods",
1588     credits: 3,
1589     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.",
1590     prereq_string: "",
1591     coreq_string: "",
1592     co_or_prereq_string: "",
1593     semester_offering: "",
1594     prereq_array: [],
1595     coreq_array: [],
1596     pre_or_coreq_array: []
1597 },
1598
1599 {
1600     number: "ECE 586",
1601     title: "Design for Manufacturability",
1602     credits: 3,
1603     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.",
1604     prereq_string: "",
1605     coreq_string: "",
1606     co_or_prereq_string: "",
1607     semester_offering: "",
1608     prereq_array: [],
1609     coreq_array: [],
1610     pre_or_coreq_array: []
1611 },
1612
1613 {
1614     number: "ECE 588",
1615     title: "Future Energy Systems",
1616     credits: 3,
1617     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."
1618     ,
1619     prereq_string: "",
1620     coreq_string: "",
1621     co_or_prereq_string: "",
1622     semester_offering: "",
1623     prereq_array: [],
1624     coreq_array: [],
1625     pre_or_coreq_array: []
```

Appendix C. Database Seed Files

```
1625 },
1626
1627 {
1628   number: "ECE 590",
1629   title: "Graduate Seminar",
1630   credits: 1,
1631   description: "Offered on a CR/NC basis only.",
1632   prereq_string: "",
1633   coreq_string: "",
1634   co_or_prereq_string: "",
1635   semester_offering: "",
1636   prereq_array: [],
1637   coreq_array: [],
1638   pre_or_coreq_array: []
1639 },
1640
1641 {
1642   number: "ECE 591",
1643   title: "INCBN Seminar",
1644   credits: 1,
1645   description: "Graduate seminar on Integrating Nanotechnology with Cell Biology and
1646               Neuroscience. Grades based on active participation, including oral presentation.",
1647   prereq_string: "",
1648   coreq_string: "",
1649   co_or_prereq_string: "",
1650   semester_offering: "",
1651   prereq_array: [],
1652   coreq_array: [],
1653   pre_or_coreq_array: []
1654 },
1655
1656 {
1657   number: "ECE 594",
1658   title: "Complex Systems Theory",
1659   credits: 3,
1660   description: "Advanced topics in complex systems including but not limited to
1661               biological systems social and technological networks, and complex dynamics.",
1662   prereq_string: "graduate standing.",
1663   coreq_string: "",
1664   co_or_prereq_string: "",
1665   semester_offering: "",
1666   prereq_array: [],
1667   coreq_array: [],
1668   pre_or_coreq_array: []
1669 },
1670
1671 {
1672   number: "ECE 595",
1673   title: "Special Topics",
1674   credits: 4,
1675   description: "",
1676   prereq_string: "",
1677   coreq_string: "",
1678   co_or_prereq_string: "",
1679   semester_offering: "",
1680   prereq_array: [],
1681   coreq_array: []
1682 }
```

Appendix C. Database Seed Files

```
1680     pre_or_coreq_array: []
1681   },
1682
1683   {
1684     number: "ECE 599",
1685     title: "Masters Thesis",
1686     credits: 6,
1687     description: "Offered on a CR/NC basis only.",
1688     prereq_string: "",
1689     coreq_string: "",
1690     co_or_prereq_string: "",
1691     semester_offering: "",
1692     prereq_array: [],
1693     coreq_array: [],
1694     pre_or_coreq_array: []
1695   },
1696
1697   {
1698     number: "ECE 620",
1699     title: "T: Interdis Bio and Biomed Sci",
1700     credits: 3,
1701     description: "(Also offered as ANTH 620, BIOL 520, CS 520, STAT 520). Varying
1702                 interdisciplinary topics taught by collaborative scientists from UNM, SFI, and
1703                 LANL.",
1704     prereq_string: "",
1705     coreq_string: "",
1706     co_or_prereq_string: "",
1707     semester_offering: "",
1708     prereq_array: [],
1709     coreq_array: [],
1710     pre_or_coreq_array: []
1711   },
1712
1713   {
1714     number: "ECE 633",
1715     title: "T: Image Processing",
1716     credits: 3,
1717     description: "Advanced topics including but not limited to computational,
1718                 mathematical, multi-scale, and spatial statistical methods for multi-dimensional
1719                 signal processing, multi-spectral imagery, image and video processing.",
1720     prereq_string: "",
1721     coreq_string: "",
1722     co_or_prereq_string: "",
1723     semester_offering: "",
1724     prereq_array: [],
1725     coreq_array: [],
1726     pre_or_coreq_array: []
1727   },
1728
1729   {
1730     number: "ECE 637",
1731     title: "Topics-Algorithms",
1732     credits: 3,
1733     description: "Advanced topics including parallel and high-performance computing,
1734                 multimedia, virtual reality, real-time systems and robotics, encryption and
1735                 security, information technology, applied algorithmics and computational science
1736                 algorithms and applications.",
1737     prereq_string: "537.",
```

Appendix C. Database Seed Files

```
1731     coreq-string: "",
1732     co_or_prereq-string: "",
1733     semester_offering: "",
1734     prereq_array: ["ECE 537"],
1735     coreq_array: [],
1736     pre_or_coreq_array: []
1737 },
1738
1739 {
1740     number: "ECE 638",
1741     title: "Topics-Architecture & Systems",
1742     credits: 3,
1743     description: "Advanced topics including advanced computer architecture, networks,
1744                 distributed computing, large-scale resource management, high-performance
1745                 computing and grid-based computing.",
1746     prereq-string: "538.",
1747     coreq-string: "",
1748     co_or_prereq-string: "",
1749     semester_offering: "",
1750     prereq_array: ["ECE 538"],
1751     coreq_array: [],
1752     pre_or_coreq_array: []
1753 },
1754
1755 {
1756     number: "ECE 642",
1757     title: "Detection & Estimation Theory",
1758     credits: 3,
1759     description: "Hypothesis testing; Karhunen-Loeve representation; optimal detection of
1760                 discrete- and continuous-time signals; ML, MMSE, and MAP estimation; sufficient
1761                 statistics, estimation error bounds; Wiener and Kalman-Bucy filtering; detection/
1762                 receivers for multiuser and multipath fading channels.",
1763     prereq-string: "541.",
1764     coreq-string: "",
1765     co_or_prereq-string: "",
1766     semester_offering: "",
1767     prereq_array: ["ECE 541"],
1768     coreq_array: [],
1769     pre_or_coreq_array: []
1770 },
1771
1772 {
1773     number: "ECE 651",
1774     title: "Problems",
1775     credits: 6,
1776     description: "",
1777     prereq-string: "",
1778     coreq-string: "",
1779     co_or_prereq-string: "",
1780     semester_offering: "",
1781     prereq_array: [],
1782     coreq_array: [],
1783     pre_or_coreq_array: []
1784 },
1785
1786 {
1787     number: "ECE 661",
1788     title: "Topics-Electromagnetics",
```

Appendix C. Database Seed Files

```
1784     credits: 3,
1785     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.",
1786     prereq_string: "561.",
1787     coreq_string: "",
1788     co_or_prereq_string: "",
1789     semester_offering: "",
1790     prereq_array: ["ECE 561"],
1791     coreq_array: [],
1792     pre_or_coreq_array: []
1793 },
1794
1795 {
1796     number: "ECE 699",
1797     title: "Dissertation",
1798     credits: 12,
1799     description: "Offered on a CR/NC basis only.",
1800     prereq_string: "",
1801     coreq_string: "",
1802     co_or_prereq_string: "",
1803     semester_offering: "",
1804     prereq_array: [],
1805     coreq_array: [],
1806     pre_or_coreq_array: []
1807 }
1808
1809 ])
1810
1811 Dept.where(acronym: 'ECE').first.courses = Array.new(courses)
```