

Lobo Scheduler: A UNM Open Data Mobile-First Web App From a Big Data Perspective

Lucas Nunno
University of New Mexico
Computer Science Department
Albuquerque, New Mexico, United States
lnunno@cs.unm.edu

Abstract—Planning our schedules for months at a time can be an immensely stressful and difficult task. In the context of college, this may be one of the most overlooked tasks that has a massive impact on the future of students in both the short-term in terms of immediate coursework and long-term in terms of career path and required skills for the work force. It is often difficult for students to plan out their lives months in advance and universities also need to have a grasp on what exactly their course catalog looks like for a particular semester. In this paper we describe Lobo Scheduler, a University of New Mexico (UNM) Open Data web application that aims to make schedule planning easier. We describe how data visualization and big data techniques allow us to make an application that is both easy to use and scalable. We will also show several visualizations that help put the schedule in context and help understand the data as a whole. Lobo Scheduler is to be deployed to Amazon Web Services (AWS) and delivered to the UNM IT department’s 2014 Mobile Application contest as a result of the work described within.

Keywords—scheduling; mobile-first; web application; UNM;

I. INTRODUCTION

A. Motivation

College students face a difficult problem at the beginning of every semester, they have to decide from a catalog of potentially thousands of courses the small subset of 3-6 courses that they will take for the next 16 weeks. We often overlook how difficult of a problem it really is to decide and plan our lives for these large chunks at a time and at the same time have a picture in our heads of the trajectory of our academic and professional careers which are both effected immensely by the small steps we take every semester.

This is why an effective schedule visualization tool can have a big impact for students that are trying to plan out their lives ahead of time. If the tools are poor, it makes discovering courses and finding the correct ones all the more difficult.

That is why we will show in this paper that having the right tools isn’t only beneficial to students, it directly aids the business goals of universities as well by cutting down costs on advisements, and metrics that negatively impact universities such as graduation rates, dropout rates, and more. Our hypothesis is that if students are able to have an

understanding of the classes they are going to take before they take them, most will make better and more informed decisions that benefit everyone.

A schedule visualization app is important to UNM because it:

- 1) Saves students and the university **time and money**.
- 2) Replaces legacy schedule web pages.
- 3) Provides a **data-centric framework** and an opportunity to expand functionality in other areas.
- 4) Improves UNM’s reputation as a forward-thinking institution and helps **attract new students**.
- 5) Breaks the schedule into more easily digestible pieces for administrators looking to **improve the school’s curriculum**.

A big data issue that we are running into as a modern society is that there are a number of public institutions that are funded by tax payer dollars, but we don’t really have the knowledge of where the money is going, how these institutions operate, or what kind of impact our funds are having on the community.

While this project does not solve these issues, it is the beginning of a larger effort to visualize and bring this data to the masses so we can have more complete knowledge of how the system is working.

As software engineers, we have a moral responsibility to empower people with the data that they rightfully own. Interacting with UNM’s schedule data is a small piece of a larger zeitgeist of empowering the masses through open data.

The tools that we will be describing have big data in mind although they are used on relatively small datasets. The technologies used have been proven to be web scalable for large companies with massive datasets.

B. Goals

For Lobo Scheduler we have a number of goals that will help move forward the schedule visualization capabilities at the University of New Mexico including:

- 1) Visualize the schedule.
- 2) Find classes easily.

- 3) Analysis of schedule data.
- 4) Provide a mobile-first cross-platform way for viewing and interacting with the schedule.

II. RELATED WORK

The related work for this project includes many web applications that we have looked at in order to emulate what a mobile friendly site looks like that can handle a significant amount of users. When choosing the appropriate backend to use we looked into NoSQL databases since they are very scalable and have a large number of companies using them in production such as Craigslist and Reddit [8]. It is becoming more common in modern software development to build in scalability from the start and companies that are massively scalable are becoming more valuable. The Facebook acquisition of WhatsApp last year shows how important scalability is for large tech companies [3]. Companies that have not built scalability in from the start are having issues as they rapidly expand [2].

Big data has also been used substantially for higher education analytics in other universities [5].

We are also dealing with the issue of native application versus web application, there has been research in this area [1] that also touches on the design considerations when developing a web application as opposed to a native app. After reading this research, web applications appeared to have more advantages since they are platform agnostic and can be visited on many other devices so long as they have a modern web browser. We are also less concerned about UI speed as you will see, since we rely on a strong back end for the performance requirements.

III. METHODS

The UNM schedule data was pulled and imported into MongoDB using a number of Python scripts. First, the data is retrieved from UNM IT's Open Data repository in XML format. Next, we create JSON serializable Python objects using custom JSON encoders. This way we can import these created Python objects into MongoDB using pymongo. We have leveraged the web scalability of MongoDB to make a web application that is massively scalable to large amounts of data.

The frontend of the application is an HTML5 application that uses the Metro UI CSS framework. Jinja2 is used to generate HTML markup on the server side while JavaScript and jQuery is used on the client side to manipulate the DOM in real time while a user is interacting with the pages. The design of the application is shown in figure 1.

Data analytics were performed with MongoDB's map reduce operations. We have used map reduce to group classes by subject and start times. This could also be used much like SQL's group by operations. We also used MongoDB's unwind functionality to query the nested structures of courses, such as meeting times and sections.

The UNM schedule data was pulled and imported into MongoDB using a number of Python scripts. First, the data is retrieved from UNM IT's Open Data repository in XML format. Next, we create JSON serializable Python objects using custom JSON encoders. This way we can import these created Python objects into MongoDB using pymongo. We have leveraged the web scalability of MongoDB to make a web application that is massively scalable to large amounts of data.

The UNM datasets that are used include:

- Schedule XML data
- ABQ building JSON data
- Perks JSON data

The ABQ building data and perks data were easy to import into MongoDB since they are JSON format, the schedule data was more difficult to import and required binding to the serializable Python objects as described above.

The frontend of the application is an HTML5 application that uses the Metro UI CSS framework. Jinja2 is used to generate HTML markup on the server side while JavaScript and jQuery is used on the client side to manipulate the DOM in real time while a user is interacting with the pages. The design of the application is shown in figure 1.

Data analytics were performed with MongoDB's map reduce operations. We have used map reduce to group classes by subject and start times. This could also be used much like SQL's group by operations. We also used MongoDB's unwind functionality to query the nested structures of courses, such as meeting times and sections.

The web server we are using is on Amazon Web Services (AWS). The deployment process was done by pulling the repository to the virtual machine hosted on AWS. We have created Python scripts that populate the MongoDB instance for the first time, all that is required is the UNM Open Data. We run the database Python script and then our MongoDB is populated and ready to go. After this, we just need to run the CherryPy web server using the server Python script. This binds the Python functions that are written to URLs such that when a page is loaded the Python function is called. Typically, when a URL is called we render a web page template with the Jinja2 templating system.

IV. RESULTS

We have created a UI that displays the available subjects offered for the next semester as shown in figure 2. The sidebar is used to navigate the web application. A user can move to the colleges and places page, including the list of courses that have been saved previously.

This user interface design was chosen because it follows design conventions that can be seen in many other native and mobile applications

The UI for a single course is shown in figure 3. This shows information on the number of credit hours a course is, the fees that are expected upon registration, and the delivery

Design overview of Lobo Scheduler

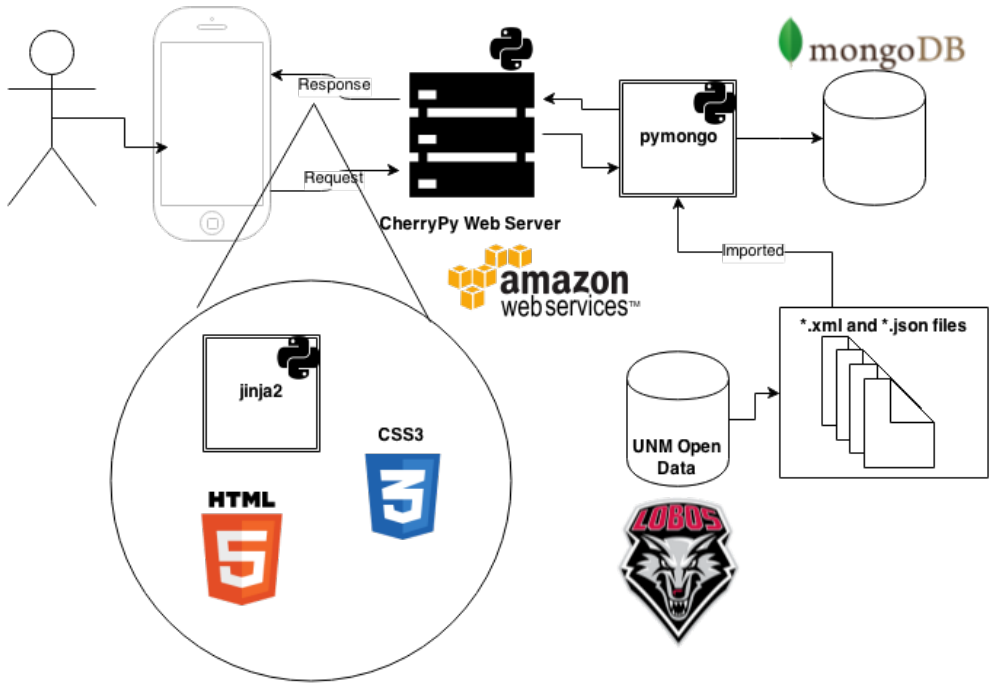


Figure 1: Design overview of the Lobo Scheduler web application and the technologies used. The backend is Python and MongoDB hosted on an Amazon Web Services VM. The frontend is a variety of HTML5 and CSS3 frameworks paired with modern JavaScript technologies.

type of a course. The amount of students enrolled in the course is shown by a progress bar. The color indicates how full a class is. The color key is as follows:

- **Green:** < 50% full
- **Yellow:** between 50% and 75% full
- **Red:** > 75% full

There is also a table of detailed meeting times which includes lab meeting times. The location is a link that navigates to a page that shows a map view of the classes location. This is draggable and zoomable like a typical Google Maps view.

Some of the data analytics that we have run include an analysis on class starting times for the Spring 2015 UNM semester. Figure 4 compares subjects to one another based on how many classes each has and the starting time for all sections. Here we can get an idea of the makeup of UNM as a whole and analyze approximately how large each subject is and when campus is typically busy based off of the number of classes that start during certain times of day.

Since the UNM schedule data is offered in XML, we tried using this format and the Python ElementTree XML API to perform traversals and search on the data, but it

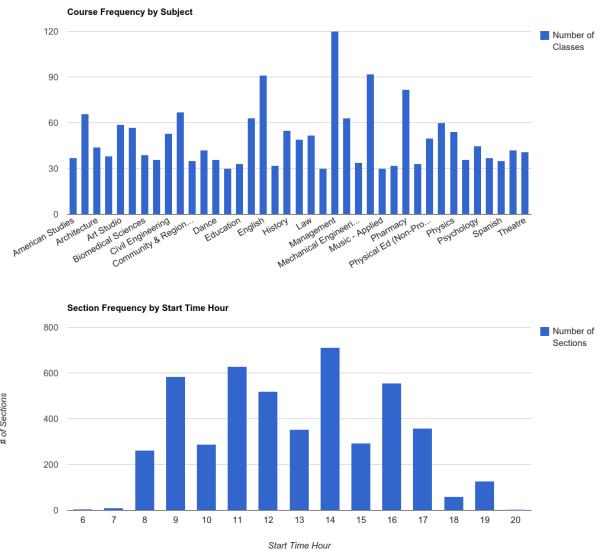


Figure 4: (Top): Visualization of number of courses by subject. (Bottom): Graphic of the starting hour time frequency of sections.

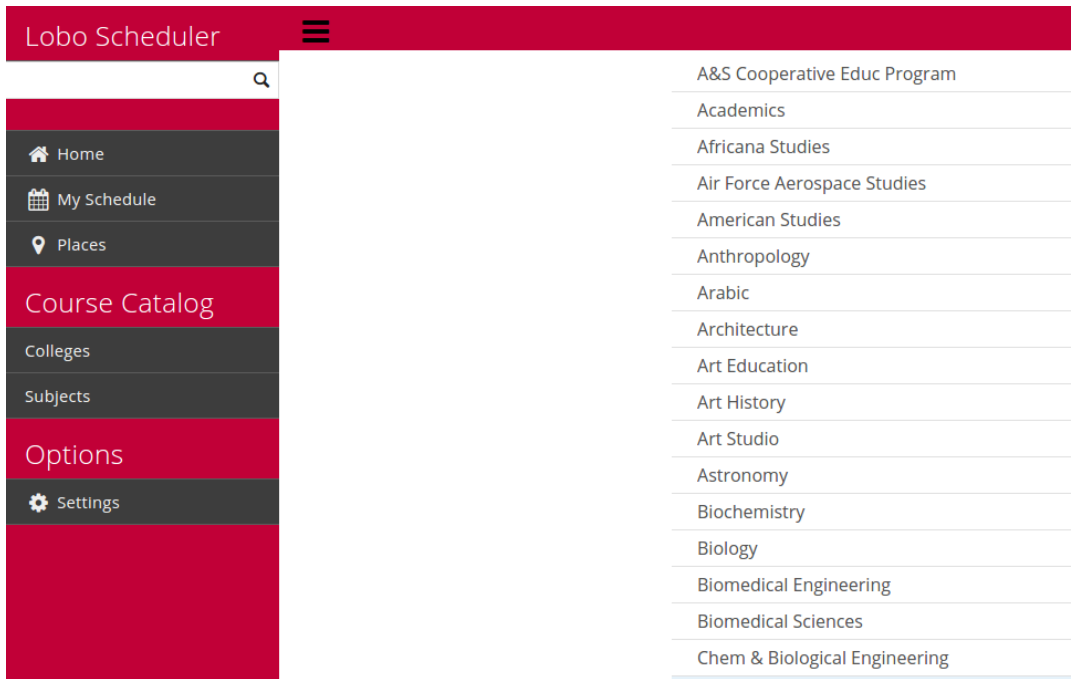


Figure 2: Lobo Scheduler UI from the splash page. By default, we show all the available subjects available at UNM. Once one is selected, only classes offered in that subject are shown. **(Left):** The side bar UI. This shows different menu options to the user. This is only shown once the menu bottom on the top navigation bar is selected and can swiped and clicked away.



CS 529: Introduction Machine Learning

Course description

Introduction to principles and practice of systems that improve performance through experience. Topics include statistical learning framework, supervised and unsupervised learning, Bayesian analysis, time series analysis, reinforcement learning, performance evaluation and empirical methodology; design tradeoffs. Prerequisite: 362 or 530 or 561.

Sections

Section 001

3 Credit Hours **\$45 Fee**

Enrollment

62/55 enrolled

22/28 on wait list

Meeting times

Days	Time	Date Range
T R	09:30-10:45	2015-01-12 to 2015-05-09

Instructors

Trilce Estrada-Piedra
 trilce@unm.edu
 Primary instructor

Figure 3: The UI for a single course. All the important information about a class is shown here, including meeting times and locations, course description, and instructor information.

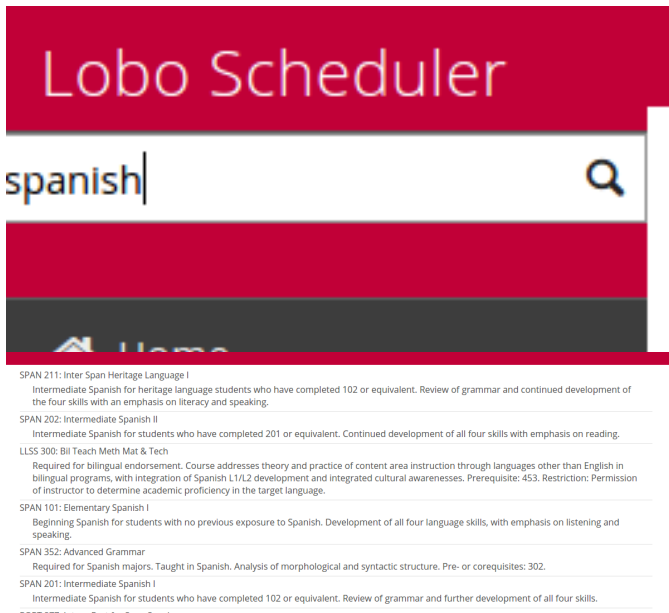


Figure 5: (Top): Entering text in the search field. (Bottom): Search results for the Spring 2015 semester.

proved to be much too slow to do search in a reasonable amount of time and only basic XPath queries were allowed. MongoDB, however, provided the ability to do SQL-like queries using a NoSQL JSON document based database. The speed of MongoDB for search was much greater than using the ElementTree API, this may be do to the indexing provided by MongoDB, so it will not linearly search through nodes like in XML. An example search query is shown in figure 5.

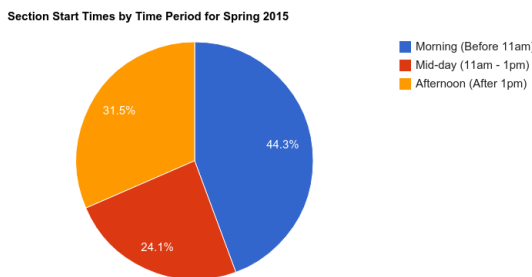


Figure 6: Summary of class start times for the Spring 2015 semester at UNM.

Figure 6 gives a summary of figure 4 and groups starting times based off of time of day. This can be used by schedule administrators when planning future class schedules, although it does not reflect student preferences; which probably vary wildly depending on background and availability.

Interestingly, we can see that most classes start in the morning and a large amount of courses occur midday as well relative to the number of hours per time period.

V. CONCLUSION

Having a data centric approach to application development modularizes the model from the view components of applications. Using a MongoDB NoSQL database with knowledge of UNM's schedule and other information makes it easy to write adaptable, responsive applications.

We have made some progress in visualizing the UNM schedule data and provided a mobile application to navigate and interact with the schedule. The application that has been made has the potential to be more useful than the current schedule navigation techniques through LoboWeb and the schedule.unm.edu interface since there exists a full search of course titles and descriptions and also the mobile-first approach is more in line with students' needs. Technology and the way that individuals interact with the web is moving towards a more mobile platform which will make businesses and institutions change their approaches to the way they present their content in order to be cognizant of this fact.

VI. FUTURE WORK

Although we have shown an effective schedule visualization tool, there are areas for improvement. Incorporating more of UNM's other datasets into the app such as location and further interactive components are areas where the app could be expanded in the future. Machine learning could also be incorporated to learn about students using the app.

It would also be interesting to do more data visualization work in this area, particularly with heatmaps using campus locations and the knowledge about when classes occur as provided by the schedule data.

ACKNOWLEDGMENT

The author would like to thank Professor Trilce Estrada at the University of New Mexico for her guidance on the project and the open source Python and MongoDB community, without whom this project would have had a drastically different fate.

REFERENCES

- [1] Consortium of Mobile Application Developers. Mobile web apps vs. mobile native apps: How to make the right choice. *Lionbridge Whitepapers Publishing*, 2012.
- [2] J. Edberg. Scaling reddit from 1 million to 1 billion: Pitfalls and lessons. *InfoQ*, 2013.
- [3] S. Ember. Facebook's \$16 billion deal for whatsapp. *The New York Times*, 02 2014.
- [4] MongoDB. Serving the digitally-oriented consumer. *MongoDB White Paper*, 2014.

- [5] A. Picciano. The evolution of big data and learning analytics in american higher education. *Journal of Asynchronous Learning Networks*, 16(3):9–20, 2012.
- [6] The CherryPy Team. CherryPy: A minimalist python web framework, 2001-2014.
- [7] UNM IT Department. Unm open data repository. <http://opendata.unm.edu/>, 2014.
- [8] J. Zawodny. Mongodb live at craigslist. <http://blog.mongodb.org/post/5545198613/mongodb-live-at-craigslist>, 2011.