COS 333 Spring 2015 Report: Noms

Annie Chu, Evelyn Ding, Nathan Lam, Clement Lee, Zi Xiang (Sean) Pan

Princeton University

May 12, 2015

1   Introduction

1.1   Initial Ideas

We started with two brainstorming sessions where we fired off and accumulated a list of possible project ideas. We then narrowed down the list based on feasibility and the interest of the group. Eventually we decided on doing a timeline and calendar integration app, something we could each see ourselves using and benefiting from. However, when we actually started planning the development of the application, we quickly realized that the complexity of the timeline design would mean that it would not necessarily be a better alternative to traditional calendars. Furthermore, we saw that the main component and challenge of the application was user interface design. Since we all agreed that wanted to put more of our efforts into learning new technologies, figuring out algorithmic challenges, and creating something that would be universally useful, we decided to scrap the idea and go back to the drawing board. Thinking again about the problems

we face everyday, we decided on creating a single-decision restaurant finder. We chose to create an iOS application over Android because none of us had previous experience with iOS development and we were all eager to learn. Armed with four Macbooks, one iPhone, and a Lenovo PC, we were now ready to go.

1.2   Motivation

Too often have we deliberated for too long on a simple question: Where shall we go for dinner? The advent of the Internet turned word-of-mouth recommendations into detailed lists and ratings and reviews, but that has not changed the core of the problem: we simply cannot make up our minds given the choices. This decision apathy is what Noms is trying to solve, by giving users a consolidated page of information such that they can focus on the deciding and not on the deliberating.

2   Design

We worked concurrently on three separate parts of the application: data collection, back-end algorithms, and front-end (iOS) development. Because of the mutual independence of the three parts, individual members of our team can work on separate parts independently. We also found that division into these three categories was appropriate as we were able to seamlessly link together the three parts once they were completed, and the bugs in any one component did not severely hinder the progress of another. We also chose to use the Parse framework, which helped facilitate this division of backend from frontend, and allowed us to focus our efforts on developing the actual application and not be bogged down with initial details such as setting up

servers and databases.

2.1   Data Collection

The database of restaurant data is a vital component of our application. For our application to work, we would need a variety of information about and a way to identify nearby restaurants. We first turned to the Yelp API and quickly realized its capabilities were limiting and the information retrievable from API calls was limited to restaurant name and address and was missing a lot of important information about the restaurant. We also realized that using Yelp API calls was slow, and decided it would be best to curate our own database of restaurants with the information we needed. With a bit more searching we discovered the Yelp academic dataset, which contains information for 250 closest businesses for 30 universities, one of which was Princeton. Although this dataset did not have much information in terms of restaurant properties, it did give us the yelp webpage URLs for a set of businesses. After filtering out the non-restaurant businesses (salons, shopping malls etc.), we used the URLs to scrape the yelp website and extract the relevant information we needed, such as pictures, phone numbers, costs. We then loaded this information into our Parse database. This scraping was done via Beautiful Soup, a python library for pulling data out of HTML and XML files from webpages.

Through this process, we learned that any application that needs an existing database of information is an extra challenge. It is not trivial to curate a sufficiently complete database, and thus the availability, sourcing, and maintenance of a database should be a primary concern for any web app. We learned creative ways to obtain data, such as scraping and parsing Yelp webpages directly rather than relying on API calls.

2.2   Backend

The crux of our backend was an algorithm for restaurant matching given a set of preferences that can be called from the iOS application and ran on the Parse server. When designing the algorithm, we considered a couple of approaches such as direct filtering and machine learning but decided they were too complex and unpredictable for our needs. Ultimately, we settled on using a cosine similarity approach, which requires minimal user information (no need to keep track of previous user input, only current preferences) and seemed to produce promising results. Cosine similarity scoring is also resistant to bad outputs because of its consistency and simplicity, which meant that there was little room for error. The ease of implementation convinced us too, as we learned to avoid overly clever or complicated things from previous COS 333 assignments.

Due to the subjective nature of the results, it was difficult to do automated testing and determine whether the results were accurate. Ultimately, we evaluated our algorithm by running a variety of queries at five different locations and taking the results and comparing it against all the nearby restaurants in our database at that location. This allowed us to fine tune a weight matrix so that we can adjust the relative importance of the elements in the user preferences object.

Working on the backend, we recognized the importance of keeping our algorithms simple and straightforward. A simple algorithm allowed us to quickly get a working version and also ensure that our code ran fast enough to keep up with the needs of the application. We also saw the importance of first having a good working foundation to build upon. We thought it was a good choice to first have a working method that returned a list of the top matching restaurants, because it allowed us to build upon it later (such as with adding weighting capabilities) while

always having a working version to fall back on. This meant that we can consistently test the backend against any inputs from the front-end and isolate any bugs immediately because of our solid knowledge of the algorithms and their correct return signatures.

2.3   Frontend

We began the frontend application development with creating mockups of the major views that we wanted and then creating the actual views in iOS. Learning how to use the iOS Storyboard and relating storyboard elements to code had a steep learning curve at the beginning, but became smoother as we got more used to the framework. We ended up redoing the layout for a number of the views to make the UI more intuitive and usable. We found that while this seemed like a necessary process, it could have easier if we had asked for more feedback on and spent more time perfecting our original mockups. However, it worked well to plan which views to have before even working on the storyboard because it meant that even though the interface was changed repeatedly, much of the code we wrote could be recycled with the changing interface. Having a set of mockups early in the design process also allowed us to fit in successive views in the storyboard while always keeping in view the core user interface and experience.

3   Testing

3.1   Alpha

We felt that our alpha test was a bit behind and not as complete as it could have been when we showed it to our TA Kelvin. At that point in time we had linked up the frontend with the

backend and the skeletal framework works, but we lacked polishing of the user interface. Had it been more complete, we potentially could have received more detailed feedback in terns of use experience.

3.2 Beta

We released a beta test at the beginning of reading period. Because of the restrictions of iOS development, we were only able to show the app to our testers using our own phone. While this restricted the amount of testing we could do, we found that we still received valuable feedback. We asked our testers to focus on functionality and ease of use, as during our beta the full aesthetic appeal of the user interface was not yet finished. We received many comments requesting a simplification of the options menu. We originally had 12 additional options listed, many of which our testers identified as not important to their restaurant choice. We iterated on this feedback and ended up keeping only 6 of the original 12. Testers also replied that our information page was awkwardly placed because it took too many click to navigate in between the main view and restaurant information page, in response we shifted the restaurant information page button onto the main view and added a separate button for committing to a restaurant. In so doing the use does not need to navigate more than one click away from the main view in any situation.

4   Milestones

Due to changing our idea two weeks into the second half of the semester, we only had 4½ weeks to complete our project before demo day. This was definitely a time crunch and limited our ability to reach our initial milestones on time. We definitely felt that we could have done a better

job with many of the earlier milestones, such as having a functional front end a week before alpha tests and having a more complete alpha test. However, we believe we caught up at the end and were able to release a Beta test and receive feedback on the application from our fellow students. Having different group members work on different parts of our app helped greatly in meeting the milestones on time, as we can always shift manpower from one part of the project to another to keep progressing on all fronts.

5   Lessons Learned

Although we believe this project to ultimately be a success, we definitely had a good share of problems and mistakes throughout. We decided to change our project about 2 weeks into the working period due to a few implementation limits and a general lack of interest in the previous idea. We do not regret this change but definitely think it could have been avoided had we spent more time deciding out idea and fleshing out the details before settling on a project. Changing our idea meant that we had significantly less time to work on our project. Even though we tried to put in extra work, we definitely would have been able to do even more had we been able to start on time. Thus, we would advise groups next year to not take brainstorming lightly and carefully think about the project idea before deciding.

Choosing to split up the work based on different components of the application worked our well in terms of efficiency, but less well in terms of allowing all the members of the team to be knowledgeable about all parts of the application. Next time, we would definitely try to facilitate more communication so that everyone is aware of all components of the application and can give input on all parts.

Lastly, we learned about the challenges of different screen sizes in mobile development. We did not pay as much attention as we should have to the differences between different versions of the iPhone. We did much of our initial development on an iPhone 6 environment, assuming that the iPhone 6 would be the most common version of the phone around. We later realized that we only had an iPhone 5S and that the versions do not scale elegantly to each other. Thus we had to completely modify almost all of our views to fit the iPhone 5S screen size. Looking back, we should have created our application in a way that either fits the screen size we will be testing on, or automatically scale to multiple sizes.

6   Future Work

If we had more time, there are a variety of features that we believe would have enhanced our application. Enabling Facebook login to the application would allow us to add number of features relying on friends and collaboration. One key feature is the ability to create group preference profiles that is shared and can be edited among groups of friends. Another similar feature is the ability to take into account multiple preference profiles (especially those of different people) and find restaurants that optimize for the set of preference profiles. Additionally, our application is not made to scale to different screen sizes. While it looks great on iPhone 5 and 5S screens, there are a few components of the user interface that appearing in unintended places on the iPhone 6. With more time, we would definitely make it so that all common iPhones would be able to use our application to the full extent. Furthermore, we would like to extend this application to Android and the web as well, since having only an iOS application largely limits our user base.

7 Acknowledgments

The project was an extremely valuable learning opportunity for Team Noms. Although we would have liked to accomplish a lot more with the application, we felt that we did create a substantial product that is worthwhile to keep working on in the future. We were able to get a glimpse of what it's like to work on a project in industry and felt that we learned valuable skills pertinent to our careers all around. Although we faced many challenges, hurdles, and mishaps, they were all valuable learning opportunities both technically and otherwise. Team Noms would like to thank Professor Brian Kernighan, Chris Moretti, and our TA Kelvin for providing the wonderful opportunity to work on a complete software project through the COS 333 course and for all the mentorship and guidance throughout.