The Great War of 2017

The Tale of LottoDeal

"Victorious warriors win first and then go to war, while defeated warriors go to war first and then seek to win"

- Sun Tzu

By

Prateek Swain, Antony Toron, Dominic Whyte, Lucas Irvine, Steven Takeshita

The Saga

The Army Assembles

Prateek looked across the room towards Lucas and knew he wanted to partner with this trusted code-warrior. A few quick iMessages later, they had a third: Steven. Prateek knew Dominic from freshman orientation, and asked him to join their battalion. With a larger group, they would be able to tackle a more challenging problem. Dominic had decided to partner with Antony, but they had been searching for other members. It was thus decided to merge and a team of five was ready for the battle of COS333.

The Grand Strategy

Fresh off an inaugural team lunch, we embarked upon the arduous task of deciding on an idea. A Frist classroom would be invaded on several afternoons for this endeavor. Days turned into weeks as we quashed ideas because gathering consensus from a team of five proved extremely difficult. The Frist classroom turned into a scene from the UN Security Council as heavy decisions were made and diplomacy was employed to receive a majority vote. But a deadlock could not be broken. One such frustrated evening, as Dominic, Prateek, and Antony sat dejected on a USG sofa, inspiration struck. E-commerce with lotteries soared from the ashes of the previously rejected ideas. And thus, LottoDeal was born. Everyone quickly came on board and an attack strategy was chalked out.

The team unanimously accepted that the best route to pursue this would be through a webapp. Dominic had built a few iOS apps before, Prateek had some experience making websites, and Antony had worked with NodeJS and AngularJS before. But besides Antony, none of us even knew what the MEAN stack was (MEAN stack, for others like us, refers to MongoDB, Express, AngularJS and Nodejs) In the spirit of the class, we decided to employ MEAN as our weapon of choice, with the goal to master this new craft. The integration between the technologies of MEAN stack would make development faster and the asynchronous nature of the stack allows for scalability and quick processing. Or so Antony told us.

The Battle Begins (Early Days)

The weekend of HackPrinceton marked the inception of LottoDeal. We worked on developing the MongoDB database that weekend by outlining its structure and crucial server API calls. However, after 30 minutes of trying to set up the environment for Mongo, it was concluded that this wasn't going to be easy. What was supposed to be a trivial process of downloading software and developing the database became our first battle. After three long hours, Lucas emphatically announced that there was going to be a steep learning curve to finish this project on time.

We sat down and began researching the MEAN stack before entering the battlefield. Steven emerged triumphant, and earned the nickname Mongod, when he discovered his new weapon of choice: mongoose, an object modelling intermediary between node.js and mongo. Due to the ease with which the database could be wiped and fields added, we were off to a great start in the first week with a working database. We set a solid foundation for the upcoming battles.

We Fight for Victory (Core Feature Development)

As the days carried on, Lewis Library's fourth floor corner study room became our 'War Room'. We would camp there for several hours at a time: brainstorming, debating, scribbling on the whiteboards, and then furiously typing on our keyboards to churn out code. We started building the homepage to display all items, which would each pop-up in a modal. But talking to Nick made us realize that users would like the ability to have multiple tabs open so that they could quickly compare items. Equipped with a new battle to be conquered, we eventually figured out how to open new pages by passing in the item ID into a new template item page, which Angular would populate given the ID. We used the divide and conquer strategy for the rest of the pages: Prateek on front-end, Antony and Dominic on backend, and Lucas and Steven filling in wherever help was needed. We soon got accustomed to the drill. Set up a basic HTML template, add CSS, infuse AngularJS, and hook it up to the backend through node and express. Once this was done, we would engage in the back and forth dance with debugging.

On another such meeting, Nick questioned the viability of our product if we lacked a reviewing mechanism for buyers. However, adding reviews on a product page was not enough. We soon realized we needed to show them somewhere apart from the product page, and so it was added to the profiles page. But how was anyone going to know when they bid on an item, or if a lottery had happened? Thus a notifications feature was added. Our Friday meetings held us accountable and made sure we were delivering everything we had initially promised. Thanks to the invaluable feedback of Nick, we added features that we had not anticipated.

Proud of our progress so far, during one such meeting, we decided to create something novel to further enhance our project. Brainstorming possible ideas, our secret weapon was born: A suggested items algorithm, similar to other great platforms like Amazon, to improve the user-experience. We created our own algorithm and data structure that would that would take into account users and items and produce suggested items custom-tailored to each individual. With graphs in the back of our mind from COS 340, we modeled the network of items and users as a connected graph. Using all of the data points available from LottoDeal, we developed our own algorithm for creating useful suggested items for users.

Never Gonna Give You Up (The Setbacks)

LottoDeal finally looked robust. We had added the core functionality of uploading images, viewing and selling items, and ultimately auctioning off the item to a winner. However, with new features came new testing. As further testing of our code with real users began, bugs began to emerge. For example, as multiple items with large pictures were added, our load time slowed down exponentially. What once took only half a second to load began to take 15 or more. However, by adding a compression algorithm to the backend all posted images could be significantly compressed, and we could store both a full sized and a compressed version of the image. With this additional library added to our arsenal, compressed images loaded significantly faster on our home page, with full-size images on the item pages. Here we learned an important tradeoff - that of quality against speed. By prioritizing speed on the homepage, and quality on the item page, we marched forward with our victory behind us.

One battle we had to retreat from was deploying LottoDeal a week before the beta test. We had hoped to gain a larger base of users to fully test our product, but were presumptuous considering all of the features that were yet to be completed and all of the bugs left to be resolved. As a result, LottoDeal was not deployed until the beta test. On the brighter side, by waiting to deploy, a more stable product was released.

It was then decided to enable HTTPS instead of the less secure HTTP. We researched how to create SSL security certificates using RSA encryption. After creating the correct certificates, our website became HTTPS and was labelled secure. However, due to the enhanced security and verification, Lottodeal's loading time increased exponentially. With a heavy heart, and Professor Kernighan's approval, we decided to trade the security for speed.

Now experienced at the art of war and negotiating setbacks, we attacked our website from a new angle. It came to our knowledge that the Facebook IDs were being stored in the local browser storage. Because of this, a user could 'inspect element' and edit the JavaScript on the front-end, allowing them to change their ID to another user's and therefore manipulate the database. After some intense googling, our ally Facebook came to help; Facebook returned access tokens, which were sent securely over the web. Using this knowledge, we upgraded all of our functions to utilize this powerful weapon (access tokens). In conjunction with this change, we modularized our code so that our Facebook login code was not copied across JavaScript files.

Unbeknownst to us, these new changes rendered the entire platform useless. Frustrated and borderline distraught, we spent seven hours going through each of our JS,HTML, and even CSS file trying to understand what was wrong, albeit with no luck. To add fuel to the fire, the next day was supposed to be the much anticipated beta-test. By 3 am, the army's morale was at its lowest. Just as the sun began to rise, a ray of hope presented itself: we noticed that the

asynchronous nature of our functions and the way angular controllers worked caused function calls across different files to return unexpected values. After making the appropriate changes, we deployed the website, logged in, and used the site. We uploaded a bike, bid a dollar and waited in nervous anticipation. It finally worked! Steven won the bike, and everyone hysterically laughed with delight in the Wilson Common Room. That night we had come close to giving up, but rejoiced at the fact that at Lottodeal lived to fight another day.

The Final Frontier (Week before Demo)

In the final days of the battle, all the functionality of the project had been completed and minor changes needed to be done. Stripe had to be switched into live mode, CSS needed some final reinforcements for mobile, and thorough inspections of the barracks of our code had to be done to see if any security or server issues remained. Skilled at our craft now, these tasks were completed in lightning speed. It was time for the final frontier: the demo.

Prateek, a veteran of presentations, created the slides. Dominic, the master of media, along with Prateek, created a video for the demo to help elaborate and illustrate all the features LottoDeal had to cover. And just like that, the team started rehearsing. But there was still something missing: the oomph factor. After an hour of intense discussion, it was decided to bravely open LottoDeal to the world and have a live lottery at the end. Although opening the platform to a room full of strangers would be risky, we knew that a live demo was the secret weapon that would complete our presentation. Rehearsals went late into the night. Come dawn, we were destined to ride.

The Showdown (Demo Day)

It was the moment we had all been waiting for. Armoured with our suit jackets and color coordinated outfits, we reconvened at Friend Center and inspected our crown jewel: The chocolate espresso cupcake that had been bought for the demo. After a final rehearsal, we chanted our warcry and walked boldly towards the battlefield (stage). The light shone upon us. Moments later, we emerged triumphant as the victors of the Great War of 2017.

~The End~

What Worked Well

We realized the importance of having modular code to allow for easy changes in one javaScript file that multiple html files linked to. Our Facebook login code originally was located in every HTML's controller. However, we were able to condense all of the code into a separate module and allow all the HTML code to access it. This cut down the codebase significantly and allowed us to change every HTML's Facebook code all at once.

We also learned the importance of establishing stylistic standards with regards to variable naming and function comments. With so many people working on the code base, we needed a standard such that the next set of us could continue right where the previous group had left off. Therefore, we came up with an internal naming schema for variable names and littered the code with comments.

One useful debugging technique during the development of the web application, especially when we were boundary testing, was printing out descriptive error messages. Whenever the server crashed, we were able to address the error immediately by knowing exactly where it crashed and usually what was wrong (ie. user is null in getPublicAccount).

We wrote a testing script to test the major functionality of the backend. This proved to be extremely useful for stress-testing and checking if any new code broke our existing program. It had the functionality for adding items, bidding on them, and deleting them.

Advice For Next Year's Class

During the course of this project, we learned to work in a group and program, but there are some bits of information that we think would be valuable to the next set of COS 333 students that come through the class.

1. GitHub is often times your best friend, easily allowing you to share code amongst five people at a time. Learn how to properly use git and to never edit the same files. (This also means don't make a shell script that automatically adds, commits, and pushes all at once--no matter how enticing it is to save about 30 seconds at each commit).
2. Make sure to pick a group that you won't mind spending 30 hours (possibly more) a week together.
3. Learn the full stack because it becomes easier to debug when you understand the entire project from bottom up.

The Next Battle

Although we set a demanding goal for ourselves, we are very pleased with LottoDeal. One of the reasons we chose to work on a non-Princeton specific app was because we all hoped that one day we could launch it to a wider audience. In order to do so, we need to implement several more features, but could not complete them because of our time constraints.

One important feature is dynamic pricing for bids. Right now, a user can only bid $1, $2, or $5 on the website. This strategy is not scalable for very expensive items, which we believe LottoDeal sells more easily than traditional e-commerce platforms. After we acquire data on which price-points work best for certain selling-prices, we will develop an algorithm that allows users to bid in amounts proportional to the selling price that optimizes the speed with which an item is sold.

We also want to ensure that our content is appropriate for all audiences. This means that we must moderate the content on our site, including the images and descriptions that are posted. We want to do so using image and text recognition software, so that we can guarantee that the content on LottoDeal is appropriate and the descriptions of items are accurate.

Reflection

In spite of the immense amount of time we spent on this project, working on LottoDeal was an invaluable experience that has prepared us for any future endeavour. Although we could have settled for an easier project or stuck to our initial idea of the MVP, we became passionate about LottoDeal and were able to immerse ourselves into not just learning new programming languages but step into the shoes of hypothetical users to make product decisions, and learn how to work in a large group setting. This included learning to meet deadlines, adapt our code based on unforeseen obstacles, and to divide tasks amongst five members to maximize our efficiency as a group.

Acknowledgements

We want to give thanks to everyone who helped us throughout our time developing this project. Thank you to Nick for ensuring that we stayed on track and offering invaluable insight on design choices while we developed LottoDeal. Thank you to Professor Kernighan for his exciting lectures and for offering us important lessons from his own career. Thank you to Christopher Moretti for always providing technical help on Piazza and in person. Lastly, thank you to anyone who gave advice and feedback to us on LottoDeal. Without any of you, LottoDeal would not exist.