

# menyou - Reflection

Harihar Subramanyam, Danielle Man, Ryan Chipman, Tawanda Zimuto

## Evaluation

We are very happy with how menyou turned out. The app is polished and the codebase is clean. We were able to implement the features we wanted, and we **did not have to scramble** to get things done as the deadline approached.

The **planning and design** process proved **tremendously useful** as we approached implementation time. For instance, after we discussed the recommendation algorithm in our pitch, design doc, in-class design review, and meetings, implementing the algorithm in code was almost trivial. Similarly, the Question feature, which seemed complex and time-consuming at the start of the project was made simple and very useful after we determined how it would fit in our data model (instead of storing question templates and writing a question generation algorithm, we could reduce it to simply storing specific keywords).

Another **useful** practice which eased app development was our **frequent meetings with purpose**. We met after every class for about an hour in addition to our meeting with the TAs on Thursday. For each meeting, we had a very clear agenda and discussed our points efficiently without getting distracted or going off-topic. Furthermore, we had a very active mailing list where team members could work together to resolve technical issues and propose design changes.

One teamwork strategy which may have been helpful, but which we didn't do, would be to **use a project management tool**, such as Asana. All of our communication was through email, phone call, and in-person meetings and we stored all our important documents and task assignments in Google Drive. Sometimes this led to us having to scan through emails or look through documents to find a useful piece of information. Using a project management tool, or at least documenting everything in a single location, may have made it easier to store and track down information when we needed it.

Overall, we had a lot of fun working on menyou, we saw how design up front can save a lot of time later on, and we built a cool app that we plan to use ourselves. It was a good project!

## Lessons Learned

As we built menyou, there were a number of useful lessons we learned along the way.

### **If you use a JavaScript framework, use it from the beginning**

We did our MVP without a JavaScript framework (ex. Angular, Backbone) because we felt that the MVP was simple enough that we didn't need a framework. We reasoned that, if the

app grew in complexity later on, we'd use a framework. However, after the MVP, we discovered that it would be very tough to retrofit our app with the JavaScript framework, so that wasn't an option anymore

### **Agree on the API before any frontend/backend code is written**

We worked on the frontend and backend in parallel, but before we did that we took time to clearly specify exactly what our API would look like - the endpoints, headers, arguments, responses, etc. Thanks to this, we were able to work on our respective backend/frontend code completely independently of each other, which significantly speeded up development time.

### **Concepts and purposes are particularly useful for the MVP**

The first two weeks of the project were devoted to the pitch and design doc. We had considered starting the code during this time as well, so that we could get a jumpstart on development. However, after further discussion (and extra work in other classes), we decided to focus on the design and do the coding after the design. This proved to be very useful when it came time to start implementing our MVP. Many of the implementation details we had in mind before we started designing turned out not be necessary at all, and our design helped us dramatically simplify the code needed for features. It saved us a great deal of time. While it is natural to want to dive in and code, being forced to spend time designing instead was much better.

### **Test-driven development?**

We decided not to implement any tests for the MVP, because it was not required. We reasoned that this would give us more time to implement the features we cared about. However, much of the time we would have saved was put into tracking down bugs. Perhaps if we had done testing from the beginning, or used test-driven development, this would have been easier. We are not convinced that TDD was the answer, but it's worth doing in a future project and comparing it to this one.

## **Peer Review**

### **Harihar**

It was great working with Danielle, Ryan, and Tawanda. They were a fantastic team!

#### Danielle

One of the key selling points of menyou is its stunning frontend, and we have Danielle to thank for that. She wrote a great deal of the frontend code, and did most of the styling. As a result, the app looks quite polished and has great usability. I want to highlight this and recommend that, even if she works on backend/infrastructure on some future project, she ensure that either she or a teammate maintain high standards for usability for the frontend. This made a world of difference for menyou.

### Ryan

Ryan was the first to commit code in our project - he laid out the directory structure, set up build automation with Gulp, hosting with DigitalOcean, logging, debugging, and more. This proved to be an incredibly wise move, because we could see the structure of the project and we knew exactly where to put our code. Tasks like compiling templates and deploying were all automated, so we could focus entirely on the code itself. I want to emphasize how useful this was, and recommend that in the first coding-related meeting he has in a future project, he should layout the directory structure and introduce tools like Gulp/DigitalOcean, etc to his team. If his teammates spend a little time learning these tools from the beginning, it will save them a lot of time later on.

### Tawanda

Tawanda worked with me on the backend, and I was very impressed with the code he wrote, and especially the tests that he wrote. I had remarked that it would be challenging to test parts of the API thoroughly, but Tawanda was able to build good API tests (ex. mock database, making API calls in code, doing a “fuzzy” matching results when precise results could not be known). This was remarkably useful in helping us track down bugs (I recall finding a subtle null reference bug thanks to his tests). Not only will I use his tests as reference when I want to do node API testing in the future, I also recommend that Tawanda keep these tests and show his teammates in future projects how to write such API tests.

## **Danielle**

### Harihar

Harihar did a fantastic job working on and leading this project. From the start, he made sure that we knew what the deadlines were and completed everything on time. He did a great job formulating and giving the presentations that we made to the class (and leading the design at the beginning). Additionally, he was very organized and made sure that we maximized the productivity of our meetings with the TAs by coming to every one with a clear agenda. He also made sure that all of the documents that we had to submit (including this one) were formatted nicely and extremely thorough and well thought out. All of these attributes were reflected in the code that he wrote as well. The recommendation system that Harihar wrote and the way that he overcame challenges which we did not foresee while dealing with the Locu API were both phenomenal. The questions feature that he implemented was also incredibly impressive and adds a lot of depth to the app. All of the code that Harihar wrote was very clean and well documented, and for me personally, it is a standard to which I will try to write my code in the future.

### Ryan

Ryan did an incredibly impressive job setting up the app at the beginning. I had no idea that such things as Gulp existed, and I'll definitely be using them in the future. He actually introduced me to a lot of new things that I didn't know existed before, such as Stylus, as well, and I'm grateful for that. He was very adamant that we write code cleanly and keep it in an organized file structure, and I'm thankful for that as well because even when I was trying

to find parts of the project that I had never seen before (server things), I had no issue finding what I was looking for quickly. I interfaced with Ryan the most because we were both working on the client, and I learned so much from him by just asking questions when I got stuck. He did a great job connecting the middle of the app and packaging all data that needed to be sent to the client into this `Menyou.state` variable, which was incredibly useful and made the implementation much cleaner. He also introduced me to the idea of tokens, which I hadn't thought of using in authentication before (though I knew they existed), and made the super cool loading animation!

### Tawanda

Tawanda was the only team member I think to have touched all parts of the code, from testing the server to making the map actually respond to user requests. I was so impressed with everything that he did! I'll definitely be using the tests that he wrote for the API as an example and standard to which I will try to write my tests in the future. I was also super impressed by the way he made the google map change locations based on the user repositioning the map or making a geolocation request. He incorporated that new feature into the UI very well and it adds a lot of depth to the app. He was a reliable teammate and wrote code that works extremely well, and he did it very quickly and cleanly. My only comment would be that he should voice his opinion more in the design process, because I would have liked to hear that and I was certainly very impressed with everything else that he did.

### **Ryan**

### Harihar

Since Harihar's work was mostly backend-focused, the phase of the project where we ended up working closely was the design phase. Harihar's thorough, verbose documentation and meticulous attention to detail throughout the design process and throughout the entire project were incredibly helpful, and on multiple occasions saved me a trip into the codebase when I needed a behavior/API clarification. Harihar was also a fantastic sounding board for design ideas. It was very helpful to have someone who would always ask "why?" and "what are the advantages of that solution compared to this?" whenever we discussed design or implementation decisions, and I would encourage him to continue to do so, especially as he leaves the classroom environment for industry work, as it makes him a very valuable teammate.

### Danielle

Since Danielle and I both focused on the front-end code, I ended up working more closely with her than with any other member of the `menyou` team. She was a hard-working and flexible partner, and was always happy to work around last-minute changes and fill in with the odd job here and there that would otherwise have fallen through the cracks. That, coupled with her great work on the front end, made this project a great one to work on and made her a great teammate to work with.

### Tawanda

Though I had less of an opportunity to work directly with Tawanda than with some of the other team members, I nonetheless benefited from and gained a great appreciation for his invaluable role on the team. Though I had almost no interaction with the backend code beyond a few scaffolding commits early on, everything Tawanda produced was readable, well-documented, and exhaustively tested, and the times that I did need to dig into his code were incredibly easy and pleasant experiences! Tawanda's contributions were incredibly well executed, and my only other feedback would have been that he could have been a bit more assertive in the design decision-making, since he certainly has the technical chops to back it up.

### **Tawanda**

### Harihar

Harihar insisted on keeping the code base clean from the start. As a result we didn't have trouble refactoring the code in the later stages of the project. He made sure that he tests and comments every piece of code that he worked on. He did a very good job acting as the representative for our team when we met with the TAs and presented in the class.

### Danielle

Danielle's work on the user interface was excellent. She carefully chose the colors, the font types etc. for the UI. The user interface is very responsive which is a very remarkable feature for our application. The client code she worked on alongside Ryan was very clean. I didn't have trouble adding new features to the UI.

### Ryan

Ryan laid a viable security framework for the application. Having worked on only cookies before, I found inspiration from his token based authentication and authorization. The client code that he worked on with Danielle was clean and extensible. He wrote several utility tasks which increased the overall productivity of the team. His work on deploying the MVP and the final version was very useful.