

ByteBuddies: Final Report

TIKKI CUI, ILYA MRUZ, NATHAN ROBERTS, and SACHIT TRIPATHI

Going to university is often overwhelming, especially when pursuing a difficult major such as Computer Science (CS). Being assigned a mentor upon arrival can help ease the transition and help students succeed as CS majors. We created a bot that makes the process easier for both mentors and mentees alike. The bot lets the mentors and mentees to fill out their personal profiles, and then finds best matches for everyone. This makes the matching process easy and streamlined. Although the bot is currently designed with CS students as the primary focus target, it can easily be applied to many other fields, both academic and industrial.

ACM Reference Format:

Tikki Cui, Ilya Mruz, Nathan Roberts, and Sachit Tripathi. 2023. ByteBuddies: Final Report. 1, 1 (December 2023), 4 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

The realm of the computer science industry is ever-changing and constantly evolving. Students constantly need to learn complex algorithms, difficult coding languages, and new technologies. The demand for proficiency and mastery in the field is higher than ever, with job markets seeking individuals skilled in the latest advancements. In this difficult environment, students often lack a source of guidance and mentorship.

ByteBuddies is designed to tackle this very problem. It will act as a way for students to connect to possible mentors. These mentors possess the expertise and skill to guide students through the labyrinth of academia and into the competitive job market. Providing this resource for CS students will aid in their knowledge, skills, and confidence. This product will help foster a community in the computer science field that will continue helping each new generation of CS students.

Both mentee and mentor will benefit from this. The mentee has someone who can guide them through their homework, course planning, and overall academic journey. The mentee also has someone who can identify their weak areas and provide suggestions on how to improve that area. The mentor also greatly benefits from mentoring someone. By teaching someone what they already know, the mentor is continuing to strengthen their knowledge. Mentoring someone will help the mentor to gain a truly thorough grasp on their knowledge. After all, teaching others is the best way to learn.

2 MOTIVATING EXAMPLE

Suppose a new student comes to Virginia Tech, planning to major in Computer Science. However, although they are very interested in the field of computer science, they have yet to learn any relevant skills. Right off the bat, there's a ton of material that needs to be learned. From languages such as Java and C, to tools like Git, and abstract concepts such as theory and computer architecture, it can be very overwhelming for this student.

By having a mentor, the student can be guided towards resources to learn new skills and hone existing ones. The mentor can also help identify where the student is struggling and how to best improve in that area. However,

Authors' address: Tikki Cui, ttcui@vt.edu; Ilya Mruz, ilyamruz@vt.edu; Nathan Roberts, nathan03@vt.edu; Sachit Tripathi, sachitkt@vt.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, or post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

XXXX-XXXX/2023/12-ART \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

having a mentor is only half of the battle. It's also important to have a *good* mentor. With ByteBuddies, these students can find the best-fit mentor for them. They can then connect with this mentor and form a strong and continuing relationship throughout their academic lives.

3 RELATED WORK

3.1 CEED

Virginia Tech's Department of Engineering already has a similar program called CEED Peer Mentoring, where it matches up older students with new students to help them adapt to VT and grow in their skills. [1] However, the program's criteria for matching is not very detailed. While there are some criteria, it's fairly generic so students aren't always paired with the most suitable mentor. Furthermore, the purpose of this program to target underrepresented demographics.

ByteBuddies has a slightly different purpose. Instead of applying to all engineering students, it focuses specifically on CS students. Instead of focusing on the underrepresented demographics, it's designed for all CS students. ByteBuddies uses detailed criteria and profile matching algorithms to ensure that mentees can find the best possible mentor. ByteBuddies also allows the mentee complete freedom on who their mentor is. While ByteBuddies may offer recommendations based on the matching algorithm, the final choice is entirely in the mentee's hands.

3.2 Chronus

Chronus is a mentoring service tailored towards software engineers. [2] It allows users to create profiles highlighting their skills and experience. Both mentors and mentees can fill out similar profiles. Afterwards, Chronus indicates how well two profiles match with a "percent match" indicator. This aids mentors in finding suitable mentees to enhance their experience. This also helps mentees to find a good mentor that will fulfill their specific needs.

This service is very similar to ByteBuddies. Both uses profile matching to help find the best pairings between mentees and mentor. Both are also targeted towards software engineers. However, ByteBuddies narrows its scope down to students, whereas Chronus is more for industry people. As a result, ByteBuddies has some more specific additional criteria. In addition to considering skills, experience, and technical goals, ByteBuddies also considers a student's age and academic goals. At the end of the day, both mentee and mentor are students, so ByteBuddies needs to ensure it does not hinder either student's educational journey.

4 IMPLEMENTATION

4.1 Design Decisions

For the overall project, we chose to use the Model-View-Controller (MVC) model. This model suited our project since it needs to support a lot of interaction between users and behind-the-scenes database operations. As we designed the frontend, we made sure to follow good HCI practices and follow the 3 Golden Rules for user interface design.

For the website frontend, we decided to use ReactJS with TypeScript as the main framework, with some CSS to make everything look good. For the backend, we used Python's Django library since it supports the MVC model very well. The database is hosted on MongoDB since our data is closer to the non-relational data scheme. The code repository is managed with GitHub and we use Trello as our Kanban task-tracking board.

4.2 Process

We initially started with the Agile Scrum methodology. We felt like it was a very flexible and incremental approach that allowed us to efficiently break down our project into manageable sprints. This process worked out well for

the first few milestones. Before we started a milestone, we would hold a Sprint planning to discuss the assignment specifications and divide up the work as we saw fit. While we didn't hold daily stand-ups, we held them a few times a week to check in on the progress. After a milestone was completed and before we started the next one, we would do a Sprint review. This allowed us to reflect on the process, go over feedback for any missed points, and ensure everyone was on the same page.

However, towards the later end of the semester, it became more difficult to hold these meetings each time. Due to increase workload for other classes and personal matters, our schedules simply couldn't align often to hold these meetings. Therefore, we opted to switch to Kanban. We felt like it was suitable for our project development without requiring regular meetings. We chose to use Trello for our task management board and it worked out very well until the end.

4.3 Testing Approach

For black box testing, we came up with several different use cases and recorded the input with the expected output. We would then compare the expected with actual output and note any discrepancies. For white box testing, we followed the strategy laid out in class. [3]

We started off with unit tests for all the different classes and components that we had written. These tests are written atomically and thoroughly test every possible outcome in the code. This ensures that each component of our code works as intended. Next, we ensure that these components can work together properly. We do this with integration testing and ensure that the system is still running properly. Afterward, we would start using actual users for input. We do validation testing by having users try out the system and mess around with it. This helps us uncover any unusual scenarios that we might not have thought of and fix them.

Finally, we wrap up with system testing. This is similar to validation testing, except it's the developers and designated testers that are testing out the system. We want to make sure that everything is working not just on a high level, but also behind the scenes. With these testing strategies, we can have a comprehensive and thorough test run-through on our finished product.

5 DEPLOYMENT PLAN

5.1 Deploying

We would probably start off with deploying using either AWS or GitHub pages. Both sites allow quick and easy hosting options for a simple startup website. Although AWS costs money, it is a lot more powerful and has more hosting options for things like databases. On the other hand, GitHub Pages is free, but also very basic. We would probably start off with GitHub pages as an early prototype. Once we are able to generate a solid user base and start gaining some revenue, we can migrate to AWS to scale up our capabilities.

5.2 Maintaining

To maintain this project, we would have some sort of issue tracking system that users can contribute to. If we chose to have this project open source, we could use GitHub's built-in issue tracking system. If not, we can use other methods, such as discussing forums or bug-reporting forms so users can report bugs and suggest feedback. This would help us to stay up-to-date with our users and what they want.

Within the team, we can track these issues with Kanban, like with the Trello board. While we are working with user input, we would also be finding problems within the code. We would use the floss refactoring process to ensure that the code stays maintainable throughout its entire life cycle. This also helps minimize risk of introducing a huge change that potentially breaks the code.

6 CONCLUSION

6.1 Limitations

One limitation of ByteBuddies is that it's not fully implemented. Although we started to implement it, the end of semester became too busy for us. We ended up having to shelve the project in order to prioritize our other classes and projects.

Another limitation is that ByteBuddies is very small and not powerful enough to handle the whole Virginia Tech population of CS students. We would need to scale up both the hosting service and database in order to be able to fully serve every Virginia Tech CS student.

6.2 Future Work

There are plenty of possible future improvements to ByteBuddies. One is to broaden the scope beyond just CS students. ByteBuddies has the potential to apply in both directions. It can go up to the industry and research fields, while also coming down to help lower education students. ByteBuddies also doesn't have to be just CS, it can be for any field of any study.

Another improvement is with the algorithm that matches profiles. Rather than making matches purely on calculated numbers, we can use machine learning to make matches for more subjective criteria, such as preferred style of learning. This is something that math can't easily handle. However, this will require a lot of data design and collection.

6.3 Final Thoughts

Overall, ByteBuddies was very successful in terms of a design project. Even though it wasn't fully implemented, we learned a lot about the overall journey of good software design principles. From analyzing requirements to designing the project to implementing and hosting the project, we acquired a lot of knowledge and skills. Not only were we able to learn new concepts and practices, we were able to sharpen our existing ones. Everyone on the team enjoyed this project and improved a lot as a programmer.

REFERENCES

- [1] "CEED Undergraduate Programs." *Engineering | Virginia Tech*, 25 Oct. 2023, eng.vt.edu/ceed/ceed-undergraduate-programs.html.
- [2] "About Chronus Mentorship Software." *Chronus*, 4 Dec. 2023, chronus.com/about-us.
- [3] Brown, Chris. "Testing." *CS3704: Intermediate Software Design*. CS3704: Intermediate Software Design, 13 Nov. 2023.