**Overview**

This document will present my reflective piece of everything that happened this trimester. What I have learned (WHAT), How I used what I learned throughout this trimester (SO WHAT). How will I use what I learned moving forward (NOW WHAT)

**What I have learned (WHAT)**

Starting this module, I got very excited to learn that it would focus on critical aspects of what I was about to do in my full-time software engineering job: security and Microservices. At the start, things were going very well for me. I had just started my full-time teaching job at my local university. I have been pursuing the job for the past five years while working there as a part-time lecturer. I have also just gotten promoted to CTO at my software engineering job. I was responsible for setting up the backend architecture for our upcoming MMORPG (Massive multiplayer online role-playing game). I knew I would go towards the microservices architecture route to scale our servers to millions of players, and I was excited to know that I would be learning it in this module. It was effectively hitting two birds with one stone.

After finding out this would be a group project, my excitement lessened, thinking that the entire group workload would fall on my shoulders like every group I have ever been a part of. However, I was pleasantly surprised with my group. Every single person pulled their own weight. This was the only group I have ever been a part of where I did not have to do all the work. I enjoyed working with every single member of the team.

Since I had the most experience between everyone, I was voted to be the team lead. I started by setting up Trello, Slack, WhatsApp, Google Calendar, Git and GitHub for myself and for everyone on the team. We got all that set up in the first week, which was awesome because we could now focus on our project. Every Sunday, we would meet at 11 PM UK time to work on the project, and throughout the 12 weeks, we all did our part. Everyone showed up to our weekly meetings except for maybe once or twice.

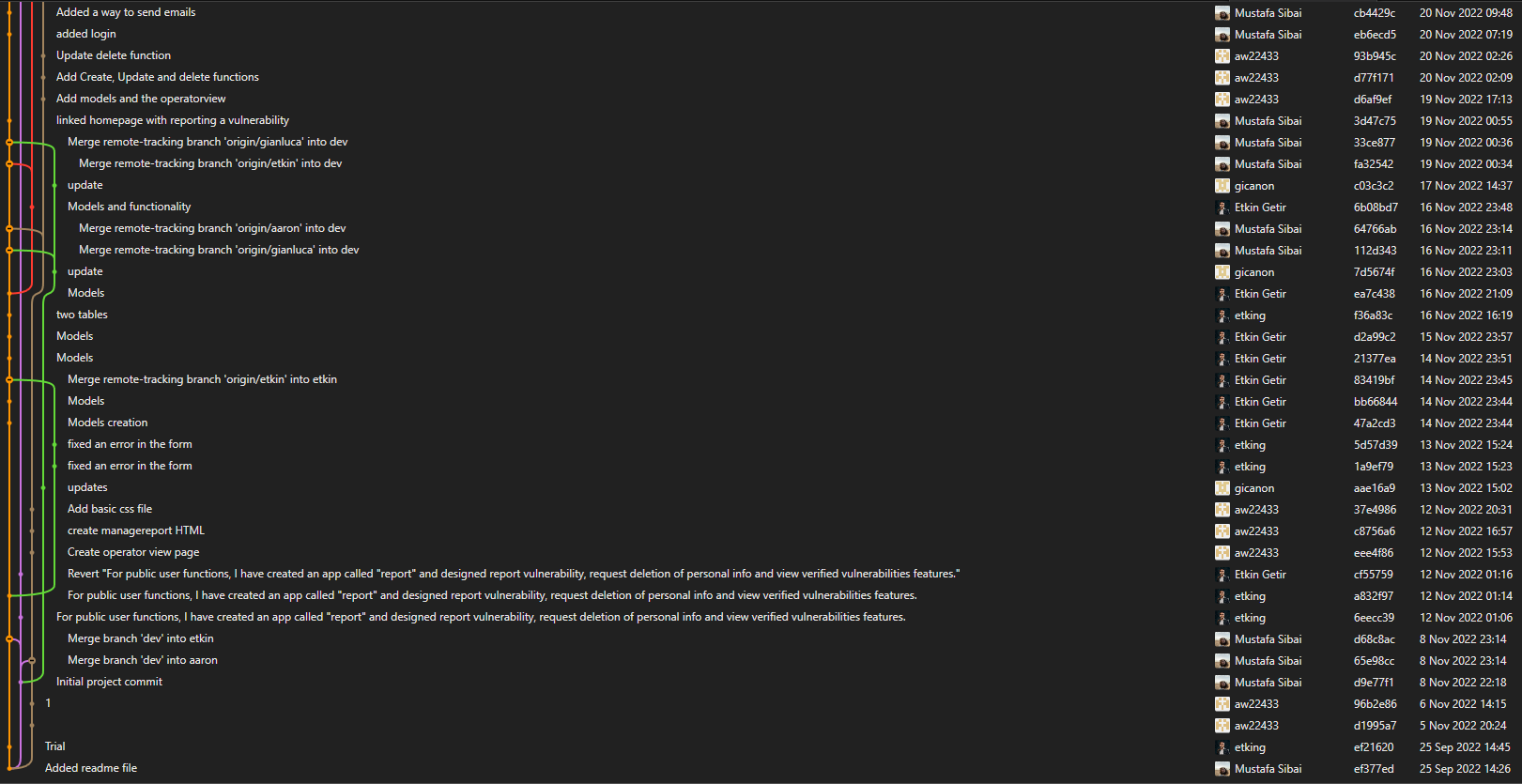
Graphical user interface, text, application

Description automatically generated

Our first submission was fairly easy. We all voted to use Django as our technology, and things were going excellent. We did our UML diagrams, and we all contributed to the software architecture. Everyone selected a UML diagram to work on. Sadly at this point, one of our team members, John, dropped out, but despite that, things were going well.

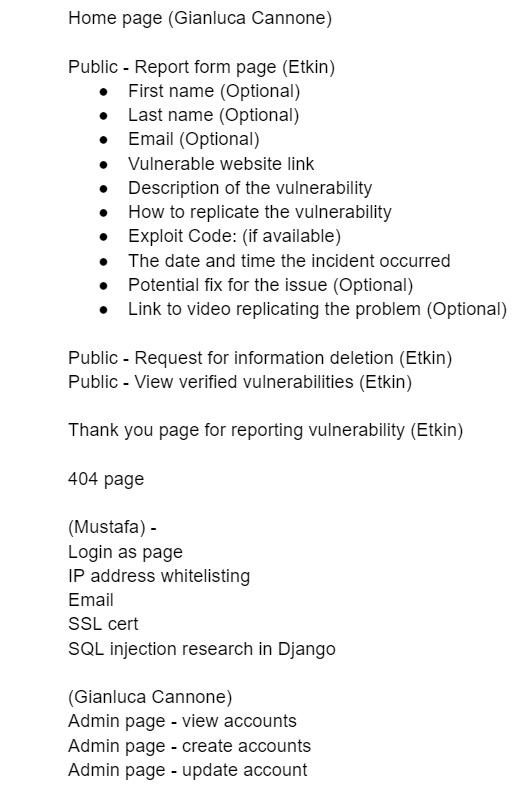
After submitting our first project, we split our tasks equally. I picked sending emails, IP whitelisting, authentication, SSL certification, and SQL injection. Gianluca worked on the Admin Page and Homepage. Arron worked on the operator page. Etkin worked on the public report submission page. Ola helped around in the project and was responsible for developing the design and ensuring the website layout and theme fit together plus he did the project test cases. We all met every week to share our progress and learn from each other.

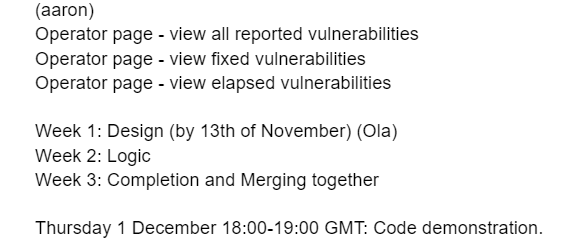
The following five weeks were both fantastic and terrible for me. The good part is I got to work on Django again. A framework I previously used in my last module. The bad part was that my father passed away, which was a rough time for me. However, I made sure not to leave my team behind and contribute to the project as much as possible, which I managed to do successfully. Sadly while I was contributing to the project every week, I did not contribute to my weekly activities relating to my e-portfolio. As a result, I had to finish eight weeks worth of activities in week 12 which was not a pleasant experience.



[The above picture shows how everyone on the team was contributing to Git]

As part of my work, I had to learn about SSL certificates, setting up an HTTPS server, and how to deploy a Django website with Nginx and Gunicorn, all technologies I had never used before. As a team, we expected that there would be changes from our planned UML diagram and that things might be more challenging than we initially anticipated. While our planned UML diagram slightly changed, I was surprised by how easy the development went. It turns out that Django is a well-developed framework with functions protecting from SQL injection. Django also had different types of encryption algorithms for authentication.





[The above image shows which team member took which task]

Finally, I learned a lot about OWASP and security risks in the first six units in this module. I had never heard of OWASP up until that point. I also got a good refresher on buffer overflows and SQL injection attacks and how to prevent them. While I already knew these topics, I never really thought about them much when writing code. However, after researching them more, I started to look out for them whenever I wrote code in C++ and TypeScript.

**How I used what I learned throughout this trimester (SO WHAT)**

This trimester I created a Django-based admin page for my full-time job and finished creating the microservices architecture for my company's MMORPG. In addition, I learnt Kubernetes and have applied everything I learnt regarding security to my code in my full-time job. Furthermore, I started to look more into which encryption algorithm is best for hashing passwords and encrypting data. While I have always had security in mind regarding my data, security was not a top priority when I wrote code. This module has completely changed that. Now, whenever I write code, I always think, is this code secure? How can an attacker use my code against my software to steal users' data?

**How will I use what I learned moving forward (NOW WHAT)**

Moving forward, I need to watch out for buffer overflow and integer overflow. Furthermore, I plan to take code security much more seriously in my day-to-day programming. This module has opened my eyes towards what a simple mistake can cause in terms of security.

Overall this module has been great. I learned a lot about security and microservices and expanded my knowledge in Django. I really enjoyed my seminars, and Dr Cathryn was fantastic throughout the module.

The only downside was I did not progress with my master's degree thesis preparation. As I mentioned in my previous reflection, I am planning on building an ARM-based operating system from scratch. I did quite a lot of work on that last trimester, but sadly this trimester, I did not progress at all.

References:

What are microservices? (no date) microservices.io. Available at: https://microservices.io/ (Accessed: December 13, 2022).

Django (no date) Django Project. Available at: https://www.djangoproject.com/ (Accessed: December 13, 2022).

Advanced load balancer, web server, & reverse proxy (2022) NGINX. Available at: https://www.nginx.com/ (Accessed: December 13, 2022).

Gunicorn 'green unicorn' is a python WSGI HTTP server for UNIX. it's a pre-fork worker model. the gunicorn server is broadly compatible with various web frameworks, simply implemented, light on server resources, and fairly speedy. (no date) Gunicorn. Available at: https://gunicorn.org/ (Accessed: December 13, 2022).

Production-grade container orchestration (no date) Kubernetes. Available at: https://kubernetes.io/ (Accessed: December 13, 2022).