IT UNIVERSITY OF COPENHAGEN

The Pentuple
MiniTwit

Adam Hadou Temsamani (ahad)

Christian Bank Lauridsen ()

Jacob Grum (jacg)

Rasmus Herskind ()

Thor Liam Møller Clausen (tcla)

Deadline for submission: May 30, 2025

Word count: XXX

(excluding front page, table of contents, references and appendices)

Contents

1 System's Perspective

1.1 Modules

insert module view

1.1.1 Api load balancer

Api uses docker swarm with one leader giving tasks to the workers. The leader is the load balancer and the workers are the servers.

.Net minimal api because its nice.

1.1.2 Client

Blazor because it is nice and webassembly etc.

1.1.3 Seq monitor

Seq[1] because it is specialized for .NET

1.1.4 Database

Postgres because we like it

1.2 Sequence diagram

Diagram of user posting msg on client or simulator making api call. Calls api, uses logging, database, logging again. Reponse back to client/simulator.

2 Process' Perspective

2.1 GitHub actions

2.2 Terraform

2.3 Logging and monitoring

2.4 Security assessment

2.5 Scaling

Client is scalable since it is client side rendered. Api is scalable as it is stateless and uses docker swarm.

2.6 AI-assistants

Gpt was of great help in translating python to csharp. Generating files. Not so great at ...?

3 Reflection Perspective

3.1 Biggest issues

How we solved them

3.2 Lesssons learned

with regards to evolution and refactoring operation, and maintenance

3.3 The DevOps difference

Which parts were devopsy and different than other projects

References

[1] Datalust. Seq. `https://datalust.co/seq`. [Online; accessed 06-May-2025].

Appendix

A   Project repository

https://github.com/Grumlebob/The-Pentuple-MiniTwit
   (The README.md contains links to our client, api, and monitoring).

B   An Appendix Chapter (Optional)

...