# Adam Blackwater

me@adamblackwater.com • www.linkedin.com/in/adam-blackwater • 07722-432-866
www.adamblackwater.com • www.github.com/adam-blackwater

## Personal Statement

Software engineer with professional experience in back-end software development, and a Computer Science background. Currently looking for a new role after previous employer ran out of money.

Worked across teams to deliver public facing code and internal tools. Effective team member, thoroughly enjoy the collaborative aspect of software development.

Product minded and focused on creating value for both customers and business.

## Skills

| | | | | | | |
|---|---|---|---|---|---|---|
| SQL | REST | AWS/GCP | cURL | Python | Redis | Micro services |
| Git | TDD | CI/CD | Linux | aiohttp | Postgres | API design & documentation |
| SSH | Java | Kubernetes | HTTP | PyTest | Docker | Pair programming |

## Work Experience

**WayHome** | Back-End Software Engineer

Developed containerised micro-services orchestrated using Kubernetes. Wrote services for consumption by the data team to aid in creating data models.

Integrated third-party RESTful API's and wrote a service to automate a business process, freeing up valuable customer facing employee time.

## Education

**Computer Science; First-class honours** | The University Greenwich - 2019

Won lead student representative role in first year. Chaired meetings, and took sole responsibility of producing a document for review by senior staff and external observers.

## Projects

**A Chess Portable-Game-Notation (PGN) Analysis Tool** | *Rust*

▷ Command line program that iterates over PGN files in a directory.
▷ Parses PGN files and returns win/loss/draws against opening moves played.
▷ Used TDD throughout development.
▷ Learnt Rust while developing this tool.

**Open Risk** | *Python - aiohttp - MySql*

▷ Collaborating with others developers remotely
▷ Writing asynchronous Python code
▷ Graph data structure for data modelling

**Optimisation and Scheduling Software** | *thesis*

▷ TTD based project using OOP principals. Used the MVC design pattern. Made use of the Flask framework for front-end, and Python for the back-end.
▷ Applied a genetic combinatorial optimisation algorithm to optimise a scheduling problem.
▷ System read/writes MySQL database. Queries database, result given to algorithm to automate schedule creation.