

Wyatt Rasmussen

✉ me@wyattr.dev 🌐 wyattr.dev
📄 /wyatt-ras 🔄 /why-002

Education

University of Minnesota, Twin Cities (B.S. in Computer Science, 3.7 GPA)

September 2022 - December 2024

Programming Languages and Technologies

Programming Languages: Rust, Java, C#, C, C++, SQL, Ocaml, Python, JavaScript, HTML/CSS

Technologies: Git, .Net Core, Spring Boot, Gradle, Docker, AWS, Datadog, RabbitMQ, Postgres, SQL Server, Terraform, MongoDB

Experience

Software Developer Intern, IDEaS Revenue Solutions

May 2024 - August 2024

- Lowered S3 costs by 75% for inbound data processing by identifying inefficiencies in the data pipeline
- Investigated and resolved a bug causing 4 critical systems to indefinitely suspend data processing without warning
- Reduced inbound data compute usage by developing an autoscaling system using AWS CloudWatch and RabbitMQ metrics
- Improved observability for a major client integration by instrumenting 80+ steps in the incoming data pipeline using Datadog

Software Developer Intern, Force America

June 2023 - March 2024

- Saved 2 minutes per invoice by developing an accounts payable management application using Javascript and C#
- Automated internal invoices and auto-filled data for 90% of external invoices by leveraging PDF scraping and ERP integration
- Ensured that the application met stakeholder needs by leading meetings to determine ongoing system requirements
- Streamlined audit invoice retrieval by developing a custom dashboard with extended search functionality

Student Office Assistant, University of Minnesota

August 2022 - June 2023

- Increased overall student office assistant effectiveness by building several internal office tools
- Relieved burden for full-time staff by assisting the department with allowed accounting tasks
- Added to the overall visitor experience by addressing a wide range of needs of faculty and students alike

Projects

Seafoam, Distributed K/V Store (github)

Rust

- Handled 95,000+ requests each second per node by developing a K/V store in Rust
- Increased read throughput by an additional 16x by implementing a read-pipelining system
- Ensured data consistency by implementing the Raft consensus algorithm to handle leader election and log replication

Drone Delivery Simulation, CSCI 3081W Program Design and Development (github)

C++, Docker

- Designed and implemented a drone delivery simulation in C and TypeScript in a team of 4
- Cut 75% of unnecessary drone travel for most workloads by replacing the scheduler with an auctioneer algorithm
- Minimized potential conflicts with other extensions by designing the new scheduler to use the mediator design pattern

Stochastic Modeling for Card Game Probabilities (github)

Rust

- Minimized additional work for the user by creating a simple logic language for defining successes and following the current open-source standard for deck definition
- Allowed the simulation to be comparable to mathematical methods by simulating one million simulations in under 0.2 seconds

Url Shortener API (github)

C#, Docker

- Cut 25% of the wait time on short link generation by implementing pre-generation of short links
- Reduced redirect latency 40% by rewriting the .NET Core HTTP cache policy to store 302 redirects

TA Connections, 1st @ TechStars Startup Weekend 2022 (slide deck)

- Enabled TAs to engage with students outside of traditional office hours using a new, innovative video format
- Led a collaborative effort to interview test groups and conduct market research in search of a viable solution
- Delivered a concise 5-minute pitch of the MVP to fellow competitors, staff members, and judges