

Johnathan J. Flaggs

LinkedIn: www.linkedin.com/in/johnathanflaggs

Website: <https://www.johnathanflaggs.com/>

Email: johnathanflaggs@gmail.com

Phone: [+1.949.414.9545](tel:+19494149545)

Software Architect with a focus on backend system development for fintech and robotics applications. Successful record of leading projects from proof of concept into production. Professional references available upon request.

Education

University of California, Riverside & Davis

Sep. 2010-Jun. 2014

- UCR: BSME with concentration in Control Theory under Department of Mechanical Engineering (BCOE)
- UCD: Control Theory Concentration under Department of Mechanical and Aerospace Engineering

Technical Toolset

Development Languages & Environments:

- | | | | | |
|-----------------------|--------------------|---------------|------------------|-----------------|
| ▪ Visual Studio | ▪ Codeblocks | ▪ SVN/GIT | ▪ Batch Script | ▪ Node.js |
| ▪ Atmel Studio | ▪ C, C++ | ▪ C# .NET | ▪ MATLAB | ▪ HTML, CSS, JS |
| ▪ RSLogix/FactoryTalk | ▪ Beckhoff TwinCAT | ▪ MagneMotion | ▪ Fanuc Robotics | ▪ Cognex Vision |

Professional Experience

Software Engineering Manager & Architect at CarbonCapture

May. 2023-Present

Driving system software architecture and development for modular Direct Air Capture (DAC) reactors. Ensuring that our team builds a software core that is sufficiently robust, scalable, and maintainable to support current and future product deployments. Leading hiring efforts and building infrastructure and culture to support a quickly growing technical team.

- **Identify** development resource needs and create new technical positions to aid in development efforts.
- **Coordinate** hiring efforts, budget, and team structure with executives and HR. Create and improve hiring processes
- **Synchronize** a team of mechanical, software, process, and data science engineers on a weekly basis.
- **Mentor** engineers on approaching complex problems and upkeeping best coding standards
- **Enforced** OOP and heavy emphasis on robustness, scalability, maintainability, and patterns in C# and C++
- **Anticipate** and mitigate the impact of future design changes on the software layers
- **Drive** design of macro and micro software architectures which define the core product
- **Manage** and review source control on a per-commit basis to avoid common development pitfalls
- **Drive** strategic decisions regarding budget, mechanical design, safety, maintenance, and scalability
- **Interface** with external teams for outsourcing code and supplying measurement/control devices
- **Hosted** a bi-weekly coding program wherein Jr. and Sr. developers can learn and share their contributions
- **Establish** clear and achievable performance expectations and conduct regular performance evaluations

Professional Experience

Vice President, Engineering at Lighthouse Digital

Jun. 2020-Jan. 2023

Leading software architecture and development for a proprietary loan liquidations platform. The platform allows global users to participate in liquidation markets and hosts a proprietary Smart Liquidation Algorithm to provide liquidity into the markets.

- **Designed** and trained a predictive model to compete in a time-only based order book
- **Lead** the design of macro and micro software architectures which define the core product
- **Advised** CEO on market trends, competition, and macro-economic factors that impact company direction
- **Consulted** for Anchor (TFL) on improving their liquidation order book
- **Contributed** strategic decisions on product features and business operations that determine company direction
- **Managed** and mentored a team of Sr. and Jr. level developers on approaching complex problems and coding standards
- **Anticipated** and mitigated the impact of 3rd party and vendor API changes on the software layers
- **Hosted** quarterly meetings for investors and strategic partners on the status and development path of our products
- **Advised** and worked with the CEO to implement market strategies, raise funds, and allocate resources
- **Enforced** OOP and heavy emphasis on robustness, scalability, maintainability, and design patterns
- **Anticipate** and mitigate the impact of future design changes on the software layers
- **Manage** and review version control on a per-commit basis
- **Created** jobs, conducted technical screening, and helped develop an onboarding process for new-hire's
- **Allocate** and manage budgets for hiring, infrastructure, investors, and reserves

Director of Engineering at Satsy Algorithmic Trading

Mar. 2019-Jun. 2021

Lead development of an algorithmic trading platform and backend core which integrates multiple brokerage API's for live, simulated, and historical trading/analysis. The platform is multi-threaded and supports many unique features that are not available via popular retail trading platforms.

- **Developed** a custom charting solution for rendering up to 250k data points in view and > 2M points in memory
- **Lead** a non-technical team of 5 including CPAs and traders to translate their requirements into working features
- **Developed** an easy-to-use strategy/indicator language that is unlimited in extensibility
- **Multi-Threaded** C# application allows real time data streams and an intuitive trader-first UI
- **Created** a thread-safe real-time library for advanced mathematics using time series data
- **Low-Latency** execution C# library benchmarked against C/C++ for performance testing
- **Modular** core architecture allows any UI technology to consume the Satsy backend service
- **Supports** trading with brokerages including TD Ameritrade, Binance, Gain Capital, and Simulated data feeds

Lead Robotics Software Engineer at Seagate Technology

Mar. 2017-Mar. 2019

Developing software for cutting edge processes in digital storage technology. My contributions include:

- **Windows C# .NET** proprietary machine and vision process controls
- **Support** core vision libraries in C/C++
- **Cognex VisionPro** API integration
- **Machine-Vision Calibration** to establish precision robot coordinates
- **Motion Control** Kinematics, pick-n-place, multi-axis coordination, Quantum HSM framework
- **User Interface** Allowing users to fluidly interact with the multi-threaded application
- **Version Control** Using SVN, TFS and Agile/Scrum using Jira
- **OOP** Heavy emphasis on encapsulation, inheritance, polymorphism, and robust design patterns