

Madison Zaphiris

✉ mzaphiris@alumni.unc.edu

☎ 336-542-7197

🌐 [linkedin.com/in/mzaphiris](https://www.linkedin.com/in/mzaphiris)

🌐 <https://bitsandbytes.cc>

Skills

🔗 Programming	Java, Python, PHP, Node.js, C	🗄️ Databases	Cassandra, Oracle, Postgres, MySQL
⚙️ Tools/Frameworks	Spring, Kafka, Consul, Splunk, Git	🌐 Web Dev	HTML, CSS, JavaScript, Typescript, Angular
✅ Testing	JUnit, Mockito, Wiremock	👥 Soft Skills	Root Cause Analysis, Mentorship, Teamwork

Education

- Aug 2018 – Dec 2020 🎓 **B.A. Computer Science, University of North Carolina at Chapel Hill**
Coursework: *Digital Logic, Databases and File Systems, Computer Organization, Data Structures, Discrete Math, Models of Language and Computation, Algorithms and Analysis, Foundations of Programming, Text Mining, Modern Web Programming*
- Aug 2016 – Aug 2018 🎓 **A.S. College Transfer Program, Forsyth Technical Community College**
Awards: *Recipient of Nancy Hawley Women in Engineering Scholarship, 2017-2018.*

Experience

- Aug 2022 – Present 🍏 **Backend Software Engineer, Apple Pay**
Apple Pay Third Party Installments and Rewards
- Balancing needs of multiple integrating banks, drafted **OpenAPI specifications** to enable third-party installment and rewards offers through Apple Pay. Built application from the ground-up using **Spring Webflux** and **Java**.
 - Created extensible framework matching users to eligible installments and rewards offers based on anonymized card data. Integrated with existing card data **REST APIs**.
 - Orchestrated cross-functional effort across four teams to architect new **server-driven UI system**, reducing time-to-market for new installments/rewards partners from months to weeks.
 - Engineered corresponding template engine and lightweight **JSON-based templating language** from the ground-up in **Java**. Ensured support for **caching** and **localization**.
- Internal Configuration Management Tool**
- Took ownership of existing internal configuration management framework as it rapidly scaled from a proof-of-concept to **critical infrastructure used by over 10 applications within Apple Pay**.
 - Maintained and improved **Java** libraries used by applications integrating with the tool. Wrote new **Spring Boot Auto-configuration library**, vastly simplifying the integration process.
 - Rapidly triaged Po production outages related to increased load on the system's backing **Cassandra** datastore. **Participated in root cause analysis** meetings with leadership and spearheaded effort to develop more efficient data models, as well as load testing protocols.
 - Wrote **JUnit** test suite for the tool, increasing test coverage of the code base from 0 to 70 percent.
 - Collaborated with senior manager on adjacent team to define requirements and architecture for the tool's new UI dashboard. **Mentored interns and new hires** on the project.

Fidelity

- Jun 2022 – Jul 2022 🏢 **Software Engineer**
- Worked alongside software architect and business stakeholders to develop **Java/Spring Boot** and **Angular** application which tracked expenses associated with investment funds.
 - Integrated application with internal SSO library.
 - Significantly reduced application latency by optimizing existing **SQL** queries and leveraging asynchronous requests.
- Jan 2021 – Jun 2022 🏢 **Associate Software Engineer**
- Developed internal record-keeping application to manage millions of dollars in custom retirement funds.
 - Implemented reactive front-end for the application using **Angular**, integrating with existing **REST API** endpoints.
 - Integrated with existing **Oracle** datastore to serve data through **Spring Boot REST APIs**.

Surescripts

- Jun 2020 – Jan 2021 🏢 **Software Engineer Intern**
- Using **Angular**, **D3.js**, and **Java/Spring Boot**, built internal data visualization tool which rendered a graph of organizational dependencies.
 - Troubleshooted and resolved bugs on existing **Kotlin** application used to generate mock customer data for testing.
 - Upgraded authentication/SSO systems for several existing applications.
 - Improved reliability of application by writing new unit/integration tests using **JUnit** and **Selenium**.