Alan Zeng
510-930-8801 zienzeng0510@gmail.com Alan Zeng San Jose, CA

EDUCATION

University of California, Berkeley | Berkeley, CA, USA
- **Degree:** B.A. in Computer Science and Applied Mathematics, Minor in Data Science
- **Period:** Aug. 2021 – Aug. 2022, Aug. 2023 – Aug. 2025
- **Relevant Coursework**: Artificial Intelligence, Optimization Algorithms, Internet Architecture and Protocols, Cyber Security

De Anza College | Cupertino, CA, USA
- **Period:** Aug. 2018 – May 2021

SKILLS

Technologies & Tools
- **Languages:** Java, Python, Go
- **Backend:** Spring Boot, MySQL, Redis, RESTful APIs
- **Tools & Libraries:** Git, LangChain, AWS (EC2)

Artificial Intelligence
- **Adversarial Search:** Minimax, Alpha-Beta Pruning
- **Reinforcement Learning:** Markov Decision Processes (MDPs), Q-Learning
- **Probabilistic Inference:** Bayesian Networks, Particle Filtering

WORK EXPERIENCE

Backend Development Intern | JD.com | Beijing, China **January.2024 - May.2024**
- **Core System Development:** Architected and developed the core backend service for a package management system using **Spring Boot and JPA**. Implemented a full suite of **RESTful APIs on Java** to provide distinct package tracking views for over 2,000 buyers and delivery personnel.
- **Performance Optimization:** Implemented the **Cache-Aside pattern** with Redis, applying a strategic policy to cache **popular items** and **partition data by city**. This approach offloaded over 50% of read queries from the database, significantly boosting system responsiveness.
- **Database Design:** Modeled one-to-many relationships in **MySQL** (e.g., Customer-to-Packages) and implemented **role-based field visibility** to ensure data privacy between different user types.
- **Quality Assurance:** Developed a testing suite using **JUnit** and **Postman** that achieved **over 80% test coverage** for all primary RESTful API endpoints, ensuring the reliability and correctness of JSON responses.

PROJECT EXPERIENCE

Secure Distributed File System **January.2025 - June.2025**
- **Technologies**: Go, Public Key Cryptography, HMAC, Digital Signatures, Integration Testing
- **Zero-Trust System Architecture**: **Personally designed and engineered** an end-to-end encrypted file system based on a zero-trust principle, ensuring data security on top of **completely untrusted storage**. All **cryptographic operations (PKE, DSS, HMACs)** were performed exclusively on the client-side, creating a secure environment that fundamentally guaranteed data confidentiality, authenticity, and integrity.
- **Advanced Access Control & Robustness**: Implemented a sophisticated access control system featuring secure user invitations and access revocation. Upon a user's removal, the system immediately nullified their access by securely **re-keying** the shared file's cryptographic key. To ensure robustness, authored a comprehensive test suite that simulated security threats like **data tampering** and **replay attacks**.

AI Agents for Pacman **Augest.2024-December.2024**
- **Technologies:** Python
- **AI Decision Model Implementation:** Focused on implementing the backend AI decision-making logic within the provided Pacman graphical framework. Applied **Markov Decision Processes (MDPs)** to model the uncertain environment and integrated a range of algorithms, including adversarial search, reinforcement learning, and probabilistic inference.
- **Key Algorithm Design & Tuning:** Core contributions included: designing and tuning the **Reward Function** for the **Q-Learning** algorithm to guide the agent toward optimal policies; optimizing Minimax search with **Alpha-Beta Pruning**; and building a **Particle Filter** for probabilistic target tracking.

Version Control System(Gitlet) **June.2024 - July.2024**
- **Technologies:** Java, SHA-1 Hashing, Object Serialization, JUnit
- **Core System Design:** Engineered a Git-like version control system in Java. Designed a custom persistence model where **`Commit`** objects track file versions via **SHA-1 hashes**, leveraging **Java Serialization** to save all metadata and version history to disk.
- **Key Feature - Merge Algorithm:** Developed a robust merge algorithm capable of reconciling divergent development branches and automatically flagging merge conflicts. The system's reliability was validated by its adoption by over 50 peers for project versioning.