Robert Jiang

(503) 810-9393 | rcjng@outlook.com | linkedin/rcjng | github/rcjng | rcjng.github.io

EDUCATION

University of California San Diego — Expected March 2023
Bachelor of Science, Computer Science | *Regents Scholar, Provost's Honors* — *3.85 out of 4.00*

SKILLS

Languages: C/C++, Java, Python, Go, SQL, HTML/CSS, JavaScript, Haskell
Libraries, APIs & Frameworks: JUnit, Robolectric, Espresso, NumPy, Google Nearby Messages, Room, Discord
Environments: Android, Arduino
Tools: Git, Jira, Postman, Confluence, Gerrit, Jenkins, Coverity
Processes: REST, Design Patterns, Object-Oriented Design, Embedded Systems, Functional Programming

EXPERIENCE

Software Engineer Intern — June 2022 – Present
Werfen — *San Diego, CA*

- Completed over 10,000 code contributions and 25+ tickets for the agile development of a Class II whole blood hemostasis testing medical device via the use of embedded programming, object-oriented programming, and design patterns in *C*, *C++*, and *Python*
- Built several production features to manage device operation and improve device stability, such as a system verification test suite for all device subsystems, telemetry and debugging interfaces, I2C and SPI sensor sampling, and sensor analog-to-digital conversions
- Developed multiple internal tools including an EEPROM calibration assistance tool, an assay summary PDF report generator, and an assay algorithm factor calculator to improve day-to-day efficiency and future team productivity
- Ensured V&V by creating unit tests for features, performing end-to-end integration testing for other changes via *Zephyr Scale*, conducting manual code reviews via *Gerrit*, and proactively updating specification documentation and records via *Confluence*

PROJECTS

Feather | *Java, Android, SQL, Google Nearby Messages API, Room API, Git, ZenHub*

- Developed a multi-threaded *Android* social networking app in *Java* that fosters friendships by recommending close-proximity students to one another via *Bluetooth* based upon academic criteria
- Iteratively implemented features such as Bluetooth messaging via *Google Nearby Messages API*, data storage and persistence via *Room API*, sorting/filtering/matching algorithms, and UI event handling, all using object-oriented programming and design patterns
- Implemented a variety of automated CI/CD tests via *GitHub Actions* including unit tests, integration tests, end-to-end tests, UI tests, smoke tests, load tests, recovery tests, acceptance tests, static tests, and regression tests via *JUnit*, *Robolectric*, and *Espresso*

Dynamic Display Settings Switcher (DDSS) | *Python, pywin32, psutil, wmi, threading, pystray*

- Created a multi-threaded system tray desktop application in *Python* for Windows laptops that automates display settings switching (screen resolution, refresh rate, and brightness) when connecting or disconnecting from AC power
- Provided substantial average battery life improvements of 1-2 hours with negligible to no performance decrease depending on workload when on battery and no performance decrease when on AC power
- Implemented user quality of life features such as a single-click manual resolution and refresh rate switch feature via *pystray* and a JSON configuration file for persistently storing and changing user-defined display setting profiles when on battery and AC power

ServeStore | *Go, gRPC*

- Developed a distributed file hosting service in *Go* with cloud storage that allows the uploading, downloading, and synchronization of files of all types and sizes from and to your desktop devices using *gRPC*
- Employs multiple distributed storage servers for file block data and file metadata and utilizes a consistent hashing algorithm to map requests to the correct server allowing for efficient and robust operation and scaling

Tracktivity Discord Bot | *Go, DiscordGo, Discord API*

- Developed a telemetry Discord bot in *Go* that continually tracks user and server activity and reports a specific user's info, user, server, and game activity, and online status (*Tracktivity* profile) upon another user's request through Discord chat message commands
- Implemented dozens of command and event handlers using *DiscordGo* to parse and handle user commands, locate and gather activity statistics, compile and generate Tracktivity profile reports, and queue and send Tracktivity profile reports via Discord chat messages

TinyTag | *C, TinyZero, Arduino, Bluetooth Low Energy (BLE), I2C Communication*

- Developed an ultra-low power, motion-based, *Bluetooth Low Energy* tracking device in *C* and *Arduino* using the *TinyCircuits TinyZero* processor board
- Designed a lost detection algorithm that analyzes recent Bosch BMA253 acceleration data across I2C communication and reports lost duration to a smartphone via *BLE* messages
- Optimized dynamic power dissipation to 6-10mW and 1-2mA through the deployment of clock gating, CPU and peripheral sleeping, clock source switching and disabling, clock frequency lowering, and simplified logic and system design