

Anthony Jiang

s.jiang@berkeley.edu | github.com/ejiang

(754) 265-1861 | ejiang.github.io | linkedin.com/in/tjiang2/

Education

Degree

UC Berkeley, BA Computer Science Dec 2017

Courses

Algorithms, Machine Learning, AI, Probability, Compilers, Security, Linear Algebra I & II, Abstract Algebra, Real Analysis

Experience

Software Engineering Intern @ LogDNA (Feb 2018-June 2018)

Revamped account sign up flow for web app, created and deployed two report generation tools for marketing department using Stripe invoice data and usage deviation, wrote pages for static website generator

CS61AS Reader (2015)

Worked with students for introductory CS course, office hours, and classroom instruction

Web Developer (2015)

Redesigned website for Berkeley's Institute of East Asian Studies

Projects

Visit Tracker (June 2018-October 2018)

Solo project. Published a native Android app on the Google Play Store. Created novel concept for a "(store) visit tracking app," designed novel UI, created SQL database model, and wrote entire codebase. Written in Kotlin. Leverages Foursquare API as a search interface. Uses MVVM architecture. Makes heavy use of Retrofit2, RxJava2, Android Architecture Components (Room, LiveData).

Includes animations for toolbar icon, RecyclerView items, and a custom nested scrolling layout for loading more items.

Annotated YouTube demo available here: <https://www.youtube.com/watch?v=RTK2eBhN5tg>

Play Store link: <https://play.google.com/store/apps/details?id=io.github.ejiang.visittracker>

Regex Engine (private repo)

Written in Java. Takes in a basic regex, creates automata, and determines if string matches

Skills

Languages

Java, Kotlin; JavaScript/ES6, Python; C, Scheme

Frameworks & Libraries

Android, Retrofit, RxJava, MVVM, Dependency Injection, Data Modeling, OkHTTP; Git, Gradle, Google Play, Maven, Ant, SQLite / NumPy, Scikit-Learn / IntelliJ IDEA, Android Studio, VS Code / Bulma, Express, Node, jQuery, MongoDB, Grunt, React, Docker, NPM/Yarn, Webpack, Babel, AssembleJS, Handlebars, AWS S3, Sass/CSS

Unit testing