

# Avik Hasija

## University of Waterloo 1B Systems Design Engineering



a3hasija@uwaterloo.ca



(647) 772-0459



avikhasija.com



AvikHasija



/in/avikhasija

## Skills

- Android
- Java
- C++
- Python
- JavaScript
- Node.js
- MongoDB
- SQLite
- Git

## Awards

- 2nd Place - Ontario Engineering Programming Competition, 2017
- 1st Place - Waterloo Engineering Programming Competition, 2016

## Experience

### Native App Developer *CBC/Radio-Canada*

Winter '17

- Worked on CBC News and CBC Sports Android apps
- Improved UX design, developed UI features
- Implemented **client-side integration** of internal CBC APIs to increase user functionality (ie. searching for news)
- **MAUs rose 16% to 500k+** as a result of aesthetic and functional improvements
- Performed bug fixes for faulty **HTTP tracking calls** for use with Adobe Omniture Analytics
- Worked on early developer preview of **Android Instant Apps** to create instant app split of CBC News
- Worked with Media Sessions and Browser Services to create **Android Auto** POC; integrated this feature into CBC News (currently pending release)

### Lead Android Developer *Mindbend Studio*

June '15 - June '16

- Worked on front-end design, user flow, and planned algorithmic operation of infor[me], an app to improve announcement systems in large organizations
- Built scalable app with **organizational hierarchy** and layered post approval system
- Created subscription model, wrote **backend functions** to serve relevant data
- App was released in closed beta to **200+ users**

## Projects

### Political News Bot

- Facebook Messenger chatbot which performs **sentiment analysis** on articles to determine the source's political stance
- User's rating of served news is used to determine political affiliation, and subsequent articles are pulled from sources aligned with user beliefs
- News source and user **models are continually adapted** with use
- Built in Node.js and MongoDB

### PWR

- Power grid management simulator which visualizes path between power plants and homes, built with JavaScript and Electron
- Computes distance and financial metrics
- Implemented **Dijkstra's pathfinding algorithm** to reduce path inefficiencies

### Learning Lock

- Uses **Android services** to create a lock screen which can differentiate between owner and intruder (both entering correct pattern)
- Integrated lock pattern, pulled data from pattern nodes used to train and improve neural network
- Selected as DevPost weekly staff pick in February 2016