

Hiring software engineer as easy as calling a taxi.

# Nickson M.

## Senior Python/Golang Engineer

### SUMMARY

- 6 years of experience in IT
- 5 years of experience with Golang
- 6 years of experience with Python
- Fluent English
- Available ASAP

### SKILLS

<b>Main Technical Skills</b>	Python (6 yr.), Go (5 yr.)
<b>Programming Languages</b>	C, C++, Go (5 yr.), Java, JavaScript, Python (6 yr.)
<b>AI &amp; Machine Learning</b>	CNN, LSTM
<b>Python Libraries and Tools</b>	Beautiful Soup, PyTesseract, Scrapy
<b>UI Frameworks, Libraries, and Browsers</b>	CSS, HTML
<b>Python Frameworks</b>	Flask
<b>Go Libraries and Tools</b>	GORM, Libp2p
<b>JavaScript Frameworks</b>	Node.js
<b>Data Analysis and Visualization Technologies</b>	Google Spreadsheets
<b>Databases &amp; Management Systems / ORM</b>	Airtable, Firestore DB, GORM, KSQL, MongoDB, PostgreSQL, Redis
<b>Cloud Platforms, Services &amp; Computing</b>	AWS, Azure, GCP
<b>Amazon Web Services</b>	AWS Aurora, AWS Mechanical Turk (MTurk), AWS SES (Amazon Simple Email Service), AWS SQS
<b>Azure Cloud Services</b>	Azure

<b>Google Cloud Platform</b>	Google Docs
<b>Platforms</b>	AWS Mechanical Turk (MTurk), sendgrid
<b>Deployment, CI/CD &amp; Administration</b>	CI/CD, Kubernetes, Kubernetes Cluster
<b>Virtualization, Containers and Orchestration</b>	Docker
<b>Codecs &amp; Media Containers</b>	Ffmpeg
<b>Version Control</b>	GitHub, GitLab
<b>Mail / Network Protocols / Data transfer</b>	GRPC
<b>Message/Queue/Task Brokers</b>	Kafka
<b>SDK / API and Integrations</b>	OpenAIs GPT3 APIs, Zoho API
<b>QA, Test Automation, Security</b>	Selenium Webdriver
<b>Methodologies, Paradigms and Patterns</b>	TDD
<b>Other Technical Skills</b>	Complex System Architectures, DLL, Elastic, Excel automation, fcm, Firebase Authentication, Google Cloud Run, Google Drive, Image processing, ML libraries, Ozzo, Protobuf, Rev.ai

## EXPERIENCE

### Backend Engineer Cactusapp.io (US Remote)

March 2021 – Present

#### Responsibilities:

Led the design and implementation of a distributed backend to automate and speed up UX/ discovery research by automating the processing of thousands of hours of raw interviews across multiple discovery studies

- Automated interview processing by Implementing backend microservices in Python and Golang to integrate with Google Drive, Airtable, Google Docs, Google Sheets, Amazon Mechanical Turk, and APIs to support the processing of user interviews.



- Implemented specialized microservices to perform more esoteric automations such as transcription of recordings (using Rev.ai), de-identification in video and audio (using ffmpeg, pytesseract, etc), automated text nuggetization and categorization (using OpenAIs GPT3 APIs, scikit learn ML libraries, etc), and the generation of highlight videos reels and reports. Leveraged knowledge in; Machine Learning, Python, Flask, Golang, gRPC, MongoDB, GitHub Actions for (CI/CD), Google Cloud Run for deployment, Test Driven Development (TDD), Docker, Image processing in python

### **Lead Software Engineer ConceptSauce.io (UK Remote)**

March 2019 – March 2021

#### **Responsibilities:**

Remotely lead software development at this 3D market research firm. Oversaw the design, development, and deployment of a highly available and distributed Golang and Python backend consisting of dozens of microservices all communicating through Kafka and deployed on digital oceans managed Kubernetes service

- Designed and built a custom data extraction and analysis application that significantly increased company revenue by automatically providing broad insights into clients' raw research data without manually building business intelligence models.
- Led the development of a dynamic and configurable react-based survey application, the admin panel for programming it, and the python backend powering it.
- Overseeing and managing react frontend engineers, Godot frontend engineers, a QA and client management team, specialist DevOps contractors as well as leading the hiring of all this talent.  
Leveraged knowledge in; Distributed systems, Python, Flask, Golang, Kafka, KSQL, MongoDB, Concourse and GitLab for Continuous Integration / Continuous Development (CI/CD), Test Driven Development (TDD), Docker, Kubernetes, Azure Managed Kubernetes Service, Google Cloud Platform (GCP), Digital Ocean Managed Kubernetes, Amazon Web Services (AWS)

### **Software Engineer, Software Company**

June 2018 – Present

#### **Responsibilities:**

Completed 18 short and long-term projects during the occupation. I have selected and highlighted details on projects for your convenience below.



## PROJECTS

### Web Crawler Enhancement

**Description:** Project for a US-based publishing consulting firm. The task was to build fast crawlers and scrapers for a popular app marketplace with over 2.5 million apps

- Implemented a separate crawler and scrapper architecture to independently retrieve app IDs, app metadata and review data to allow both vertical and horizontal scaling
- Scrapper designed to be resilient to page data changes by working based on logical information flow as opposed to simple HTML or CSS-based selectors.

**Main tools and skills:** Python threading and multi-processing tools, Gitlab for CI/CD, Complex System Architectures, BeautifulSoup, Selenium

### Improved Authentication System on a real estate app

**Description:**The project was for a UK startup with a project called HomeKnock, a new platform for finding houses for rent. My task was to improve API endpoints by giving them the ability to dynamically infer the requesting users' user ID based on their logged-in JWT authentication token. The previous iterations of the API had this ID data inferred by the frontend which was a bit of a security risk. The project was successfully completed.

**Main tools and skills:** Secure App development, PostgreSQL

### Automating the management of tenancy for a house rental agency

**Description:** A Swiss firm that helps clients get rent reductions.

- The system I built has helped reduce human involvement in the process by 80%.
- Handles everything from initial contact made by tenants, triaging them, assigning them to human handlers, scheduling emails and calls these handlers make to the customers, tracking the results of these calls, preparing documents, and saving them into Google drive, reminding agents to mail these docs and so on.

**Main tools and skills:** Python threading and multi-processing tools, Gitlab for CI/CD, Complex System Architectures

### Automating lead generation for an accounting-as-a-service firm

- In summary, thrice a week the script would go out, search for newly posted accounting jobs on 5 job search platforms, scrape these and filter them based on a complex ruleset. The result of this process is an email with information on which companies to reach out to.
- This system has improved client acquisition for this company.



**Main tools and skills:** Web scrapping using Scrapy, Selenium and BeautifulSoup, Zoho API, SendGrid, Excel automation

### **Transportation Application (Golang Backend)**

**Description:** In this project, I was a backend engineer in a team consisting of 4 other backend developers (one being the lead engineer), 2 QA Engineers, 2 Flutter Engineers, 1 Designer, 1 DevOps Engineer, 1 Project Manager, and 1 business analyst. I was contracted onto this project by a consultancy firm based in Singapore on behalf of an informatics company belonging to one of the largest Motor Vehicle Manufacturers in the world. Our task was to work on a transportation app that, in summary, allowed users to input their location and intended destination and it suggests a route that can make use of multiple modes of transportation they can use to get to their destination.

#### **Frontend**

The system we have built consists of a cross-platform flutter app that communicates to our backend. In this write-up, as a senior backend engineer on the project, I will mostly write about our backend and my contribution to it.

#### **Backend**

The backend consists of 12 core microservices (9 of which are built) deployed on AWS Elastic Kubernetes Service. Of the currently built services, we have an API gateway, a notification service, reporting, user management, reservation management, location, trip, routing, and planning services. Of the 12 core services, 10 are written in Golang. The routing service is written in JavaScript and the planning service is built around an existing open-source trip planning system written in Java. The core services are dependent on 7 major external systems; google maps, a parking system, and 5 other internal systems and services belonging to the Vehicle Manufacturer that I cannot describe in detail. But generally, these five provided identity and access management capabilities, payments, and vehicle data. Infrastructure-wise, the services utilize Redis for caching, AWS Aurora (PostgreSQL) as the main persistence layer, AWS SQS for task queue management, AWS SES for reporting, and Firebase Cloud Messaging for sending device notifications. Of the 9 currently built core microservices, I was designated to fully implement 4, the notification service, reporting, trip, and location services. I also contributed to the implementation of 3 others: the API gateway, reservations, and user management. All these 7 services made use of the 2 core



external systems and 5 internal manufacturer systems.

Some of the major Go packages we made use of:

- Mux router for handling routing in the API gateway
- Viper for config handling
- grpc and protobuf to help with service-to-service communication
- Ozzo-validation for validating request parameters
- Gorm as our ORM for interacting with PostgreSQL
- The firebase SDK for interacting with FCM
- The AWS SDK for interacting with SES and SQS
- Testify and gomock to help in testing

This list is by no means exhaustive.

Of the 7 services I fully or partially implemented, here is an overview of what I built for each

- Trip service: Support for trip searches, search history, bookmarking trips, and managing these bookmarks. This service also communicated with the routing and trip planning services as well as google maps and a host of the manufacturer's internal systems

- Notification service: This service is integrated (mainly) with SQS and FCM to allow sending notifications to customer devices. It exposed gRPC endpoints for mainly the reservation service that needed to send out periodic notifications.
- Reporting: This was used to generate reports for the business intelligence folks in the vehicle manufacturer tasked with the business end of this product
- Location: This service heavily relied on the manufacturer's existing systems to handle all the real-time location data needs of the application
- API Gateway: Here I mostly integrated endpoints with the corresponding services I built to handle their functional requirements.
- Reservations: For this service, I handled the parking slot-related reservations. This service also handled reservations for multiple other modes of public transportation that needed reservations but my involvement for those was only in building report collection RPC endpoints for use by my reporting service.
- User management: Like the reservations service above, here I only built report collection capabilities

All in all, the backend currently has 44 API endpoints specified. Of these, 30 have currently been built as per the products' v1 specification. Of the 30, I fully or partially implemented 16 endpoints.



## EDUCATION

### Kampala, Uganda Makerere University

August 2014 – June 2018

- Academic Program: Bachelor of Science in Computer Engineering
- Relevant Coursework: Software Engineering; Algorithm analysis and design; Computer Systems and Network Security; Artificial Intelligence; Software Architecture
- Undergraduate Dissertation: Using Artificial Intelligence for classification of lesions in breast ultrasound images. I also reviewed potential deployment scenarios by building a React web application with a Python microservice serving the constructed AI models

