Maksym Syniuhin

+48577962124 | maksimsinugin1@gmail.com | linkedin.com/in/msyniuhin | github.com/MaksimSinyu

Technical Skills

Languages: Java, Python, C/C++, SQL (PostgreSQL, MySQL), JavaScript, React, HTML/CSS, Go
Other Skills: Git, Docker, Redis, Kafka, RabbitMQ, Kubernetes, Celery, Algorithms, Data Structures, Leetcode (300+ problems solved)

Experience

Autopilot System for Remote Controlled Boat March 2022 – April 2022

Solo Development *Polish, Warszawa*

- Developed an autopilot system for a remote-controlled boat using an STM32 microcontroller and C++ programming language as the core processing unit to interface with GPS, remote control receiver, and motor controllers.
- Integrated a FlySky GPS module to obtain real-time location data via UART and implemented a NMEA data parser in C++ to extract latitude, longitude, and other navigation parameters.
- Established communication with the FlySky i6 remote control transmitter through the fs-ia6b receiver to receive manual control inputs and transmit system status updates.
- Implemented motor control algorithms in C++ to regulate the boat's speed and direction by generating PWM signals to electronic speed controllers (ESCs) connected to the propulsion system.
- Developed an autopilot navigation algorithm using C++ to calculate heading and distance to user-defined GPS waypoints based on current location coordinates.
- Implemented autonomous course correction logic in C++ to continuously adjust the boat's trajectory towards the target waypoint for precise GPS navigation.
- Designed a user interface on the FlySky i6 transmitter with three main modes: manual control, waypoint recording, and engaging the autopilot system.
- Conducted extensive testing, integrating all hardware components and validating the wireless remote control link, GPS tracking, and autonomous navigation capabilities.
- Utilized an embedded C++ development environment with ARM toolchain for programming the STM32 microcontroller, leveraging object-oriented principles for efficient real-time processing and peripheral control.

Taxi Service Nov. 2023 – Dec. 2023

Java, Spring API, PostgreSQL, Kafka, Redis, FCM, Elastic Stack, Docker, Kubernetes *Solo Development*

- Developed a highly scalable and distributed taxi service application using a microservices architecture, leveraging Java and the Spring Boot framework.
- Implemented a service-oriented architecture with multiple microservices, including User Service, Driver Service, Ride Service, Location Service, and Notification Service, communicating through an API Gateway.
- Utilized Spring Data JPA and PostgreSQL for efficient data management and storage of user information, ride details, driver details, and geospatial data.
- Incorporated Apache Kafka for asynchronous event-driven communication, enabling real-time processing of new ride requests, status updates, and notifications.
- Implemented caching mechanisms using Redis to improve performance by caching frequently accessed data, such as driver information and pricing rates.
- Integrated Redis for storing and processing geospatial data, enabling efficient ride matching and distance calculations.
- Ensured secure user authentication and authorization using Spring Security.
- Implemented push notification capabilities using Firebase Cloud Messaging (FCM) to keep customers and drivers informed about ride updates.
- Implemented monitoring and logging using Elastic Stack (Elasticsearch, Logstash, Kibana) for comprehensive application monitoring and log analysis.
- Utilized Docker and Kubernetes for containerization and orchestration, enabling efficient deployment and scalability of the microservices-based application.

Pastebin Service July 2024 – August 2024

Java, Spring Boot, Redis, Minio, PostgreSQL, Nginx, Docker, Kubernetes *Solo Development*

- Designed and implemented a Pastebin service architecture to store and retrieve text data efficiently.

- Used Nginx as a load balancer to distribute incoming traffic across multiple instances of the API service.
- Developed the API service using Spring Boot to handle requests, generate unique hashes for pastes, and store metadata.
- Integrated Minio as an alternative to Amazon S3 for object storage, ensuring high availability and scalability.
- Utilized Redis for caching metadata and block data to enhance performance and reduce latency.
- Implemented PostgreSQL as the primary database for storing user data and paste metadata.
- Ensured high availability and scalability of the system using Docker containers orchestrated by Kubernetes.
- Conducted extensive testing and validation to ensure the reliability and robustness of the architecture.
- Documented the architecture and created comprehensive guides for deployment and maintenance.

Education

im. Marcina Kasprzaka Technical School Warszawa, PL

Programmer, Further mathematics *Sept. 2023*

Frameworks

Frameworks: React, Node.js, Flask, JUnit, FastAPI, Django, Spring Boot