# Dmitry K.

SENIOR SOFTWARE ENGINEER

## Education

Master Degree

Information Systems and Technologies,

Software engineer

## Language proficiency

English — B2

## Domains

AI

FinTech

Information Services

Technology

Sales & Marketing

EdTech

Communication Services

## Software engineer with 7+ years of commercial experience.

Software Engineer with 7+ years of experience in backend and full-stack development. Hands-on expertise in building AI-powered systems using Python, FastAPI, OpenAI APIs, LlamaIndex, and pgvector, with a strong focus on LLM-based document processing, RAG pipelines, and multi-agent orchestration. Skilled in integrating NLP models (e.g., Whisper, DeepFace) for multimodal use cases such as real-time video analysis and smart assistants. Experienced with PostgreSQL, scalable architectures (MVC, Clean Architecture), and mentoring small agile teams. A pragmatic problem-solver with a passion for translating business requirements into GenAI-powered solutions.

### Backend technologies

Python, Django / Django REST, FastAPI, Flask, Python Celery, SQLAlchemy, BeautifulSoup, Plotly, WebSockets, AioHTTP, OpenAI API, LlamaIndex, Apache Tika, Whisper, DeepFace, MCP, Multi-agent systems

### Frontend technologies

JavaScript, TypeScript, React.js, Next.js, Redux, MobX, AntD, MateriaUI, SCSS/CSS, HTML, AG Grid

### Databases and other technologies

PostgreSQL, pgvector, MySQL, Redis, MongoDB, ElasticSearch, LogStash, Kibana

### Architecture principles, patterns

MVC, Clean Architecture, SOLID, DRY

### Message Brokers

RabbitMQ, Kafka, SNS

### Communication APIs

RedSMS API, Twilio API

### DevOps

Docker, Kubernetes, GitLab CI/CD, AWS Cloud, Azure Cloud, AWS Serverless, AWS Lambda

### Version Control Systems

Git, GitLab, Bitbucket, GitHub

# Projects

### NDA

A multi-agent system integrated with Jira, Confluence, GitLab, and corporate chat platforms such as Slack, Google Chat, or Rocket.Chat. The system monitors the status of projects in real time, enables contextual search (based on Retrieval-Augmented Generation, RAG) directly from the project chat, and allows users to define custom rules that are automatically validated by the system. When conditions are not met, issues are escalated to management.

### Environment

Python, FastAPI, Celery, OpenAI, SmolAgents, MCP, Multi-agent system, LlamaIndex, Apache Tika, PostgreSQL, pgvector, Pydantic, HashiCorp Vault, k3s

### Project Roles

Senior Software Engineer

### Period

11.2024 - Till now

### Responsibilities & Achievements

- Developed backend features for multi-agent task validation, contextual search, and chat interaction;
- Integrated OpenAI APIs and SmolAgents into multi-agent workflows;
- Implemented API integrations with Jira, GitLab, and Confluence for real-time project sync;
- Built RAG components using LlamaIndex and pgvector for semantic search;
- Used Apache Tika for document parsing and indexing in the retrieval pipeline;
- Collaborated with architect on database modeling and migration scripts;
- Participated in code reviews to maintain code quality and consistency;
- Contributed to internal documentation and onboarding materials;
- Participated in team planning sessions and technical discussions.

## NDA

App for a network of public schools and kindergartens. Allows you to analyse video recordings of lessons according to specified parameters (beginning and end of the lesson, emotional background, use of profanity, etc.) and provides a report on how the lesson went.

### Environment

Python, FastAPI, Celery, LLM, OpenAI Whisper, DeepFace, vLLM, RTSP

### Project Roles

Senior Software Engineer

### Period

11.2023 - Till now

### Responsibilities & achievements

- Defined the database structure to ensure efficient data storage, integrity, and scalability;
- Implemented Celery tasks and backend logic for asynchronous processing of video data;
- Integrated OpenAI Whisper for speech recognition and DeepFace for facial emotion analysis;
- Used vLLM to interpret transcribed content and extract contextual insights;
- Handled RTSP stream ingestion and video segmentation logic;
- Created and maintained internal documentation for development and onboarding;
- Participated in code reviews to ensure quality and consistency;
- Collaborated with team members on task estimation and delivery planning;
- Supported internal communication and participated in retrospectives and process improvement discussions.

## AI-POWERED FINANCIAL ANALYSIS AND RECOMMENDATION SYSTEM

AI-based cloud technology for financial advisors and wealth managers. The system uses machine learning techniques to process various content feeds in real-time and boost the productivity of users in such domains as wealth management, commercial banking, and fund distribution.

### Environment

Python, Django / Django REST, FastAPI, SQLAlchemy, PostgreSQL, Celery, Redis, BeautifulSoup4, PyTest, Next.js, Redux, Docker, Ansible, Azure, ELK.

### Project Roles

Senior Software Engineer / Lead Software Engineer

### Period

11.2020 - Till now

### Responsibilities & achievements

As Senior Software Engineer:

- Incorporated Content-Based Recommender model into the project.
- Implemented RESTful API endpoints taking into account the performance of subjacent complex database queries;
- Implemented over 15 business rules and 10 import tasks;

As Lead Software Engineer:

- Actively collaborated with the business team to have a broad understanding of the business which resulted in more qualitative outcomes and time savings. Collaborated with other teams helping to solve problems;
- Responsible for the design of subsystems and large features. Helped other team members build effective architecture growing their design skills;
- Take part in setting up monitoring of the project with ELK stack, Sentry, and Azure Monitor;
- Actively participated in code review and helped junior engineers to develop their technical skills;
- Work on the documentation of the project, make diagrams based on C4 Model approach.

## WEB APPLICATION FOR TEACHERS TO SIMPLIFY THE ASSESSMENT OF STUDENT WORK

The application is designed to minimize the time spent assessing students by scanning their handwriting and automatically grading their papers. The application allows teachers to create tests for students and answer keys for those tests. Afterward, teachers can scan the answers using a smartphone, and the application will automatically sort the answers into correct and incorrect ones and mark the student accordingly. The primary task was to ensure the stability and reliability of the application, even after making small changes to the code.

### Environment

Python, Django / Django REST, Celery, JavaScript, LaTeX, Pandas, React, Redux, Material UI, PostgreSQL, GIT, FastAPI, SQLAlchemy, AWS Cloud, Docker, Kubernetes, RedSMS API, Twilio API

### Project Roles

Senior Software Developer

### Period

06.2019 - 10.2020

### Responsibilities & achievements

- Direct communication with the customer, requirements clarification, analysis, research of different implementation options, estimate them and discuss them with the customer. Implementation of the functionality related to generation;
- Assessment of tasks and implementation options;
- Work on the custom mechanism of generating PDF blanks based on LaTeX technology;
- Architecture design of tasks, and making documentation for the project;
- Product support, bug fixes, and implementation of functionality, work on improvements;
- Development of functionality, as well as delivery of changes to the live environment;
- Integrated RedSMS for automated SMS/voice notifications for user verification and alerting;
- Initiated infrastructure migration to Twilio, preparing proof-of-concept modules for communication workflows, and aligning data handling with security and privacy standards;
- Developing and updating unit tests to test functionality.

## FINANCIAL DATA MANAGEMENT PLATFORM

The custom financial data management platform is designed to provide a comprehensive dashboard for visualizing financial information. The platform pulls large amounts of data from the customer's database and presents it clearly and concisely, available to banks and financial institutions via a paid subscription. The front part of the platform is implemented to handle the processing and display of a large amount of data, while dynamic graphs and online monitoring allow real-time monitoring of financial data and system status.

### Project Roles

Software Developer

### Period

02.2018 - 05.2019

### Responsibilities & Achievements

- Integration with ML-based solution using REST API and WebSockets interfaces;
- Work on the functionality for preparing datasets and their editing process such as removing, changing their types, etc;
- Developed an interface that triggers model training process, monitors it, and provides information in real-time;
- Collaborate with frontend developers to integrate frontend and backend components using MVC architecture and Swagger for API documentation;
- Architecture design of tasks, and make documentation for the project.
- Product support, bug fixes, and implementation of functionality work on improvements.
- Features implementation, take part in groomings/requirements clarifications;

### Environment

Python, MVC, Django REST, Swagger, Python Celery, JavaScript, React.js, Redux, PostgreSQL, Django ORM, Redis, MongoDB, Websockets, AioHTTP;

## REPORT GENERATION APPLICATION

The project aims to provide a report generation feature in Slack, enabling enterprise users to generate reports for specific periods on various aspects of channel activity. The report generation feature includes data on the number of members, the number of deleted/archived channels, the number of guest accounts, and other metrics related to channel activity.

### Project Roles

Software Developer

### Period

10.2017 - 01.2018

### Responsibilities & Achievements

- Direct communication with the client, requirements clarification;
- Make a research Slack API for Enterprise users, prepare an integration plan and architecture taking into account Slack endpoints structure and different rate limits for each endpoint;
- Working with the next Slack API features: bot, slash commands, modals, events;
- Created welcome page, endpoints to receive requests from Slack, services to process these requests and collect the necessary data (will also generate reports later).

### Environment

Python, Flask, Slack API, Celery, RabbitMQ, PostgreSQL, Jinja2

# Professional skills

| SKILLS | | EXPERIENCE IN YEARS |
|---|---|---|
| BACKEND TECHNOLOGIES | Python | 7 |
| | Django / Django REST | 7 |
| | FastAPI | 4 |
| | Flask | 3 |
| | Python Celery | 6 |
| | SQLAlchemy | 5 |
| | BeatifulSoup | 4 |
| | WebSockets | 4 |
| | AioHTTP | 1 |
| | OpenAI | 2 |
| | LlamaIndex | 1 |
| | MCP | 0.5 |
| | Multi-agent systems | 1 |
| | Tika | 2 |
| | LLM | 2 |
| FRONTEND TECHNOLOGIES | JavaScript | 6 |
| | TypeScript | 2 |
| | React.js | 4 |
| | Next.js | 3 |
| | Redux | 4 |
| | AntD | 4 |
| | SCSS/CSS | 4 |
| | HTML | 4 |
| | AG Grid | 2 |

| SKILLS | | EXPERIENCE IN YEARS |
|---|---|---|
| DATABASES AND OTHER TECHNOLOGIES | PostgreSQL | 6 |
| | pgvector | 1 |
| | MySQL | 2 |
| | Redis | 4 |
| | ElasticSearch | 3 |
| | MongoDB | 3 |
| | LogStash | 2 |
| | Kibana | 2 |
| ARCHITECTURE PRINCIPLES, PATTERNS | MVC | 7 |
| | Clean Architecture | 5 |
| | Microservices | 5 |
| | SOLID | 5 |
| | DRY | 5 |
| MESSAGE BROKERS | RabbitMQ | 5 |
| | SNS | 1 |
| | Kafka | 2 |
| DEVOPS | Docker / Kubernetes | 5 |
| | GitLab CI/CD | 5 |
| | AWS | 3 |
| | Azure | 4 |
| VERSION CONTROL SYSTEMS | Git | 7 |
| | GitLab | 4 |
| | Bitbucket | 2 |
| | GitHub | 7 |