

Aydın Demir

Gebze\Kocaeli | aynndemr@gmail.com | +905350542879 | [GITHUB](#) | [Linkedin](#)

SUMMARY

Backend Software Engineer with 2+ years of professional experience in enterprise healthcare systems, with practical, hands-on exposure to distributed systems concepts gained through a self-driven microservices project covering event-driven communication, the Outbox Pattern, Choreography Saga, and a 25-package internal NuGet framework. Building working knowledge of .NET 10.0, Clean Architecture, DDD, and cloud-native backend development.

EDUCATION

Kocaeli University
Computer Engineering

Kocaeli, Türkiye
2016 – 2021

EXPERIENCE

Full Stack Software Developer – Software Engineer Solvent Software Inc.

Gebze, Kocaeli | August 2024– Present

- Contributed to development and maintenance of a large-scale Hospital Information System (HIS) serving 500+ active users, working across 10+ clinical and administrative modules including Patient Admission, Outpatient Clinic, Billing, Cashier, Laboratory, Dental, Clinical Engineering, and Needs Management Information Systems.
- Took primary responsibility for the Billing module — the most business-critical component of the system — implementing end-to-end financial workflows covering invoicing, payment tracking, and insurance reconciliation.
- Implemented VEM (Veri Eşleştirme Modülü) compliance, designing database schemas, table structures, relationships, and column standards required by the Turkish Ministry of Health to ensure lossless data migration between healthcare software vendors.
- Integrated national health infrastructures E-Nabız and MHRS via SOAP/XML web services, ensuring full regulatory compliance and secure bidirectional patient data flow with government systems.
- Optimized critical SQL queries through indexing, query restructuring, and database normalization, achieving 5x+ improvement in backend response times across high-traffic modules.
- Contributed to UI/UX improvements across multiple modules using JavaScript, HTML, CSS, Bootstrap, and jQuery, improving accessibility and responsiveness for clinical staff.
- Participated in agile sprint cycles and maintained codebase through GitLab version control, supporting reliable and collaborative deployment workflows.

Software Engineer Intern Kocaeli University

İzmit, Kocaeli | July 2020– September 2020

- Built a full-stack mini e-commerce application using MVC architecture, covering product listing, cart, and order management flows.
- Designed and implemented RESTful APIs for frontend-backend data exchange, applying REST principles and HTTP best practices.
- Wrote and optimized SQL queries for database record management, applying relational data modeling principles to design efficient table structures.

PROJECTS

TelcoCRM Microservices Platform (.NET 10 — Enterprise Distributed System)

GitHub: <https://github.com/aydindemir1/Udemy-TelcoCrmMicroservicesMyWork>

Core Package: <https://github.com/aydindemir1/UdemyCourse-CorePackage-MyWork>

- Built a self-driven enterprise microservices ecosystem (following a structured course curriculum, then extended independently) consisting of 8 independent services (IdentityService, CustomerService, CatalogService, BasketService, SalesService, InvoiceService, SearchService, ApiGateway), each with its own dedicated database — applying the Database-per-Service pattern across SQL Server, PostgreSQL, MongoDB, Redis, and Elasticsearch.
- Built out a companion framework of 25 reusable NuGet packages (Core Package Suite) covering cross-cutting concerns including CQRS/MediatR pipeline, Clean Architecture abstractions, Outbox Pattern, JWT/2FA security stack, RabbitMQ transport, and OpenTelemetry tracing — consumed by all services as internal NuGet dependencies.
- Applied RabbitMQ and the Outbox Pattern for reliable asynchronous inter-service communication, practicing at-least-once delivery guarantees and data consistency across service boundaries without distributed transactions.
- Practiced the Choreography-based Saga pattern across the order fulfillment flow (BasketClearedEvent → SalesService → OrderPlacedEvent → InvoiceService), coordinating multi-step business processes without a central orchestrator.
- Configured Ocelot API Gateway with Polly resilience policies (Circuit Breaker, Retry, Timeout) and Steeltoe Eureka for dynamic service discovery and load balancing.
- Built a full observability stack: Prometheus metrics, OpenTelemetry distributed tracing exported to Jaeger, health check endpoints, and Grafana dashboards — enabling end-to-end request visibility across all services.
- Containerized the entire platform with Docker Compose, orchestrating 15+ containers including infrastructure dependencies (RabbitMQ, Elasticsearch, Kibana, Mailhog, Jaeger, Grafana, Prometheus).

Cinema Ticketing System (DDD & Clean Architecture)

GitHub: <https://github.com/aydindemir1/CinemaTicketingSystemDDDLesson-Aydin>

- Practiced and implemented a backend system using Domain-Driven Design (DDD) and Clean Architecture principles.
- Structured the application into Domain, Application, Persistence, and API layers to ensure clear separation of concerns.
- Implemented authentication, caching, and service bus modules as cross-cutting concerns.
- Built a scalable architecture with unit testing support and database migration pipeline.

Distributed Messaging & Queue System (.NET, RabbitMQ, Kafka, Redis)

GitHub: <https://github.com/aydindemir1/December2025QueueSystemEducation>

- Implemented messaging patterns using RabbitMQ and Kafka for asynchronous, decoupled service communication.
- Developed background processing pipelines with .NET Worker Services.
- Applied Dead Letter Queue (DLQ) and retry mechanisms for fault-tolerant message handling.
- Utilized Redis for caching and built shared messaging abstractions using .NET Channels.

Docker & Kubernetes Production-Style DevOps Project

GitHub: <https://github.com/aydindemir1/docker-kubernetes-for-developers>

- Practiced and implemented a containerized multi-service architecture using Docker, including image creation, optimization, and environment configuration.
- Deployed and orchestrated workloads on Kubernetes using Pods, Deployments, Services, ConfigMaps, and Secrets.
- Configured Ingress networking to expose and manage external application access in a Kubernetes cluster.
- Implemented CI/CD pipeline automation using GitHub Actions for building, tagging, and deploying Docker images to Kubernetes environments.

.NET Microservices Architecture Project (Event-Driven Distributed System)

GitHub: <https://github.com/aydindemir1/MicroservicesWithNetWithTwoTerm>

- Practiced and implemented a microservices-based distributed system using ASP.NET Core, focusing on service independence and scalable backend architecture.
- Enabled asynchronous inter-service communication using RabbitMQ, implementing an event-driven architecture for decoupled service interactions.
- Integrated Keycloak for centralized authentication and authorization across distributed services.
- Applied Docker-based containerization to orchestrate and run multiple services in a consistent environment.

TECHNICAL & SOFTWARE SKILLS

Programming Languages: C#, JavaScript, SQL, Java (SE)

Software Engineering Principles:

Object-Oriented Programming (OOP), SOLID Principles, Design Patterns (GoF), Clean Code, DRY, KISS, YAGNI, Separation of Concerns, Dependency Injection, Inversion of Control

Computer Science Fundamentals:

Data Structures, Algorithms & Complexity Analysis, Recursion, Sorting & Searching Algorithms, Graph Algorithms, Memory Management

Backend & Frameworks: ASP.NET CORE API, .NET CORE, Entity Framework Core, MediatR, MassTransit, Polly, FluentValidation, AutoMapper, ASP.NET CORE MVC, ASP.NET MVC, ASP.NET Web Forms, ADO.NET, .NET Framework

Databases: MS SQL Server, PostgreSQL, MySQL, MongoDB, Redis, ElasticSearch

Architecture & Design: Clean Architecture, Onion Architecture, Hexagonal Architecture, Domain-Driven Design (DDD), CQRS, Microservice Architecture, Event-Driven Architecture, RESTful API Design, SOAP API Design

Messaging & Queuing: RabbitMQ, Kafka, Redis

DevOps & Cloud: Docker, Kubernetes, CI/CD, GitHub Actions, Jenkins, Cloud Fundamentals (AWS, Azure)

Security: JWT, OAuth 2.0, OpenID Connect (Keycloak), IdentityServer

Web Services: RESTful, SOAP, gRPC

Observability: Serilog, Elastic Stack, OpenTelemetry, Prometheus, Grafana

Testing: Unit Testing, Integration Testing, Test-Driven Development (TDD)

Version Control & Collaboration:

Git, GitHub, GitLab, Branching Strategies (GitFlow), Code Review Workflows, Agile & Scrum

Frontend: HTML5, CSS3, Bootstrap, jQuery

Languages: English (B2)

Online Courses & Self-Study

- **TelcoCRM Microservices Platform (.NET 10 — Enterprise Distributed System)** – Udemy, (Microservices, Outbox Pattern, CQRS, Clean Architecture, DDD, Service Discovery, Observability, Docker)
- **Domain-Driven Design with .NET** – Backendguru, (Domain-Driven Design, CQRS, .NET)
- **Distributed Messaging System with .NET** – Backendguru, (RabbitMQ, Kafka, Redis)
- **Docker and Kubernetes for Developers** – Backendguru, (Docker, Kubernetes, GitHub Actions, Grafana, Istio, Prometheus)
- **CI/CD DevOps Pipeline** – Udemy, (Docker, Kubernetes, Jenkins)
- **REST API & gRPC Development with .NET** – Backendguru, (Entity Framework Core, xUnit, Docker, .NET, Grpc, Swagger, JWT)
- **C#.NET Core Clean Architecture & CQRS Project Infrastructure** – Udemy, (Clean Architecture, DDD, CQRS, C#, .NET Core)
- **.NET Backend Bootcamp** – Fatih Çakıroğlu, (.NET, C#, Entity Framework Core, LINQ, MS SQL Server, ASP.NET Core API)