

Title Senior Software Engineer

Reports To: Senior Software Principal Engineer

Department: Software Engineering

Role Purpose & Scope

The Senior Software Engineer is responsible for designing, developing, and deploying the core services that carry the heaviest logic and the highest load on our platform. You will work on the pricing and eligibility services at the core of our platform. This is a deep back-end position on the current Microsoft stack, covering C#, modern .NET, and Azure. This role has no front-end component.

The systems you will own are rules-driven and computationally demanding. They handle large volumes of complex, interdependent business logic where correctness is not negotiable and where a poorly chosen data structure or an unnecessary allocation shows up as latency our customers can feel. The work rewards engineers who genuinely enjoy this. Reading a profiler trace, reasoning about algorithmic complexity, restructuring a hot path, and knowing which design pattern fits the problem rather than reaching for a familiar one.

Analytical depth is the other half of the job. Much of the difficulty here is not writing the code, it is understanding intricate business rules well enough to model them cleanly, spotting where a new rule conflicts with an existing one, and reasoning about how a change propagates through a system with a great many interacting conditions. If you like decomposing genuinely complicated problems and you are comfortable being the person who must be right, this role will suit you.

We also invest seriously in AI-assisted development, and this is a chance to build that skill in a place that treats it as real engineering practice rather than an experiment. You will learn to work alongside AI tooling in a way that raises the quality of what you ship, and you will come out of this role with experience that is scarce in the market right now.

You should be able to work independently with little supervision, influence architecture decisions, and lead technical initiatives across the team.

How We Work

Written clarity matters more here than in most engineering roles. We expect engineers to think through a problem and write it, before building it.

You are accountable for everything that merges under your name. AI tooling can produce a lot of plausible-looking code quickly, and the person who submits it owns whether it is correct. Getting a change verified matters as much as building it. Tests and documentation are part of delivering work, not follow-up tasks. Sensitive parts of the system always get human review no matter what an automated check says.



We are also honest about the mix of work. A meaningful part of the team's capacity goes into extending and supporting systems that are already live, not only into new development.

Role Responsibilities

- Design, develop, and maintain the back-end services at the center of our platform using C#, modern .NET, and ASP.NET Core.
- Own the performance characteristics of the code you write. Profile, measure, and optimize for latency, throughput, memory allocation, and database efficiency rather than assuming and hoping.
- Apply design patterns and architectural approaches deliberately and be able to explain why a given pattern fits the problem and what you rejected.
- Model complex business logic in a way that stays maintainable as rules accumulate and identify where new requirements conflict with existing behavior before they get built.
- Design and implement cloud-based solutions on Microsoft Azure using the compute, data, caching, and messaging services appropriate to the workload.
- Write down an implementation approach before building, covering the components you expect to touch, edge cases, the test approach, and how the change can be reversed if needed.
- Produce the test evidence, blast radius, and handoff detail that lets QA verify a change without guessing at what it touches.
- Work with AI-assisted development tooling as part of your normal workflow and develop the judgment to know where it helps and where it should not be trusted.
- Review code carefully, including code you did not write and code that was generated. Confirm that what was built matches what was agreed and reject work that does not reconcile.
- Take ownership of scalability, reliability, and security through sound engineering practice rather than through after the fact patching.
- Troubleshoot and resolve complex technical issues in production systems, including problems that only appear under real data volumes and real concurrency.
- Mentor and guide other developers and raise the technical standard of the people around you.
- Stay current with the Microsoft ecosystem and advocate for adopting what is genuinely worth adopting.
- Collaborate with Product, QA, and other engineers across all phases of the software development lifecycle, from planning and design through testing and deployment.

Key Performance Indicators

Within your first ninety days, you can deliver meaningful changes to complex logic without close supervision, and you can explain the performance implications of what you built.



Within your first year, you are a trusted reviewer on the hardest parts of the codebase, you have measurably improved the performance or maintainability of areas you own, and other engineers come to you when a problem is genuinely difficult.

Qualifications and Skills

- Seven or more years of professional software development experience with a strong focus on the Microsoft stack.
- Bachelor's degree in Computer Science, Software Engineering, or a related field, or equivalent practical experience.
-

Technical Skills

- Deep proficiency in C# and modern .NET, including asynchronous programming, concurrency, and memory behavior. We are looking for someone who understands what the runtime is doing, not only how to make the compiler happy.
- Demonstrated experience with application performance engineering. Profiling and benchmarking, finding and fixing bottlenecks, reasoning about algorithmic complexity, and reducing allocation and query overhead in code that runs at volume.
- Strong command of design patterns, domain modeling, and separation of concerns, with the judgment to know when a pattern adds value and when it adds ceremony.
- Substantial experience with ASP.NET Core and RESTful API design and implementation.
- Strong SQL Server skills covering schema design, indexing, and query performance tuning. You should be able to read an execution plan.
- Proven experience with Microsoft Azure, including compute, managed data services, caching, and Azure DevOps.
- Experience with build and release automation, automated testing, and infrastructure as code.
- Experience working in a large existing codebase where changes have consequences beyond the file you are editing.

Analytical and Professional Skills

- Exceptional analytical ability. You can take a dense, ambiguous set of business rules, find the underlying structure, and turn it into something clean and testable.
- Ability to gather and interpret data to support a technical decision rather than arguing from preference.
- Ability to document complex business logic clearly enough that someone else can maintain it.
- Excellent problem-solving skills and real attention to detail. In this part of the system, small errors are expensive.
- Strong verbal and written communication skills.
- Ability to make sound decisions under pressure and to work independently while still operating well within a team.
- Strong organizational and time management skills, with a proactive approach to learning.



Desired Skills

- Experience with high throughput, low latency, or calculation-intensive systems.
- Experience with distributed caching, messaging, and event-driven patterns.
- Any prior exposure to AI coding assistants in a professional codebase. This is a plus, not a requirement, and we will invest in developing it if you are curious and rigorous about it.
- Test-driven development and automated testing approaches beyond basic unit tests, including property-based or invariant testing for calculation-heavy logic.
- Experience with cloud native deployment and containerization.
- Experience in financial services or another regulated industry a plus.

Competencies and tech stack

- Our primary stack is C#, modern .NET, ASP.NET Core, Entity Framework Core, LINQ, and SQL Server, running on Microsoft Azure with Azure DevOps for source control, build, and release automation.
- Some older modules still carry earlier framework technology, and modernizing them is active work rather than something we pretend is not there.
- We use AI-assisted development tooling throughout our delivery process, supported by our own internal engineering practices and documentation standards. We will go into more detail on this during the interview process.

LoanNEX is an equal opportunity employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, creed, sex, national origin, age, disability, marital status, sexual orientation, gender identity, veteran status, or any other protected status in accordance with applicable federal, state, and local laws. Applicants must be able to work without employer sponsorship. LoanNEX is committed to providing reasonable accommodations for qualified individuals with disabilities.