

Letter of recommendation

Zagreb

May 17, 2019

To whom it may concern,

This letter serves as a recommendation of Mislav Jakšić for even the most demanding of software engineering positions.

While he was finishing his studies to obtain a Master of Science in Computing degree, I was his supervisor on a project for a startup company AISoft Technology that I partner with. During the time that we worked together, I got to know him very well, and I have seen his coding skills in practice.

His assignment was to develop a backend system that exposes a REST API for managing resources of Apache Kafka clusters for our cloud infrastructure platform. The project combines many technologies, including Java, Spring Boot, REST, MVC, Apache Kafka, and JMX. The ultimate goal of the project was to develop our own Confluent Control Center from scratch, including metrics reporting. I would rate it as a demanding project and not a project for a novice developer.

First of all, Mislav was enthusiastic about everything we did and he always talked with a noticeable passion about the progress in his work. In a short period, he outperformed my expectations. He was the only developer on the project, and he has proved to be reliable and able to complete a demanding task on his own.

Part of the project called for a straightforward implementation with a great amount of coding. However, the required technology was not so straightforward, so Mislav had to research it thoroughly. He had to combine information from many resources and even fall back to the analysis of source code of open-source projects.

He writes clean code and always applies best practices. The classes he wrote were designed with separation of concerns. By applying design patterns, he made the code modular which proved important in a situation which called for an architecture change in order to reduce dependencies, as well as to allow automated project deployment on nodes in a distributed environment. Some examples of his dedication to clean code include well-named functions and variables, usage of the null-object pattern, the facade pattern, decorators, and the like.

I have enjoyed software engineering discussions with Mislav. I have worked as a teaching assistant in a computer science department for more than 10 years, and in all that time, I never met a student with whom I was able to discuss software best practices in such detail. One example is the comparison of the books *Code Complete* and *Clean Code*, and which principles they describe should be used in which situations. Many students do know of the books, but lack the "when" and "why" for application in practice.

What impressed me with Mislav is the level of evidence he provides with all his arguments. Whenever we had a discussion about system design or the implementation of the project, he always brought out the right evidence and based his work on best practices, hard facts and comparisons, and always with a reference to the source from which the solution was derived. For REST API design, he referenced *RESTful Web Services* by Leonard Richardson and Sam Ruby.

Another thing I immensely appreciated was the fact that Mislav shared his own ideas and proposals whenever there was a choice to be made about system design, with a regard to the impact on the project goal. As scalability and performance were high-priority requirements, this allowed me to guide the project in the right direction.

Even though most of our meetings were conference calls, the project goal was not jeopardized by this form of communication. I attribute it mostly to Mislav's proactive and impressively responsible attitude from the very beginning.

The only time when I saw room for improvement for Mislav was when there were open discussions about current project issues. He tended to want to take upon himself the responsibility to solve an issue, even though it would be perfectly reasonable to allow somebody else to do it. I attribute this again to his dedication to the success of the project, but this tendency might cause him to overburden himself with tasks, which might be detrimental for him in the long run. We have discussed this and he has understood my reasoning, so I am sure that it will not take him long to improve in this area as well.

To conclude - I would not recommend employing Mislav as "just another software engineer". I recommend that his future employer offer him a challenging project where he will be able to show his full potential and where he will be allowed to apply his skills to make a significant difference.

Sincerely,



Miroslav Popović, Ph.D.
Lead Solutions Architect
AISoft Technology
popovic.miroslav@gmail.com