The Zalojna Kushta

Software Architecture Document

**Version 5**

Architecture constraints and design decisions.

The application uses Spring Boot as its’ backend framework. Spring Boot offers a streamlined development experience, allowing me to quickly set up a functioning server-side application.

On the frontend, I chose React for its component-based architecture and virtual DOM, enabling an interactive and efficient user interface. React's flexibility and reusability of components aligns well with my goal of delivering a responsive and dynamic user experience.

For the database I am using MySQL. MySQL is a widely adopted relational database management system known for its reliability, scalability, and open-source nature. Additionally, MySQL integrates seamlessly with Java-based technologies, offering compatibility with Spring Boot, which is perfect for my Spring Boot application.

SOLID principles adoption

Single Responsibility:

My application follows the single responsibility principle – every class has its’ individual purpose and is used for that alone.

Interface segregation:

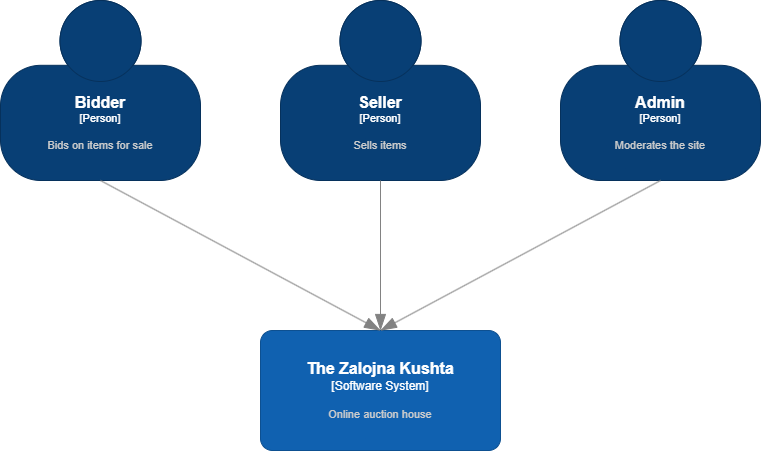
My application does not require my classes to implement interfaces they aren’t using.

Dependency inversion:

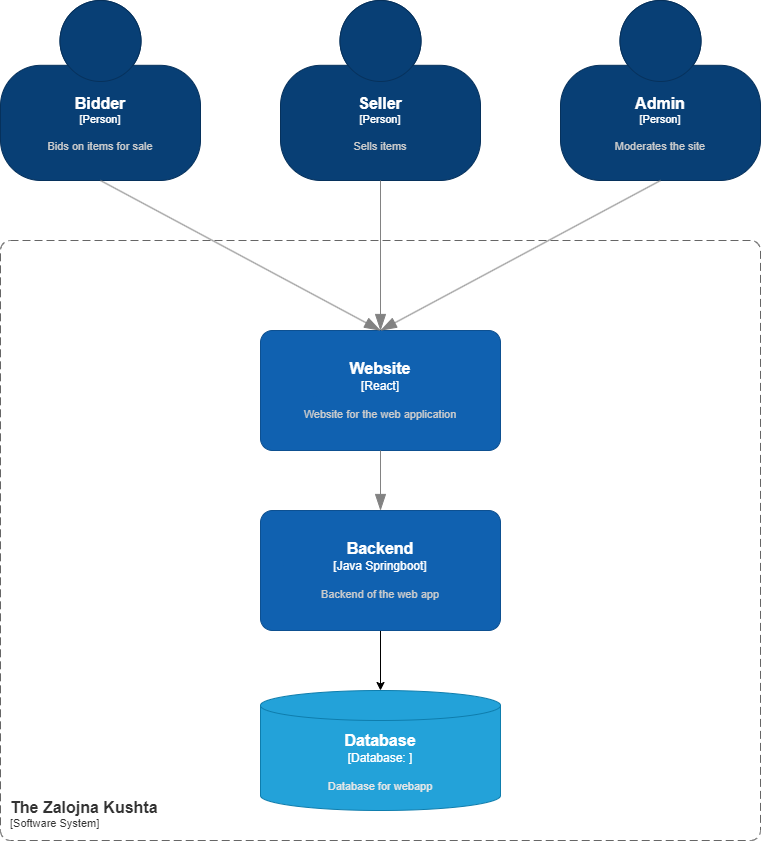
By depending on abstract interfaces instead of concrete classes, my application follows the dependency inversion principle.

Architecture diagrams

C4 - Level 1 Diagram:



C4 – Level 2 Diagram:



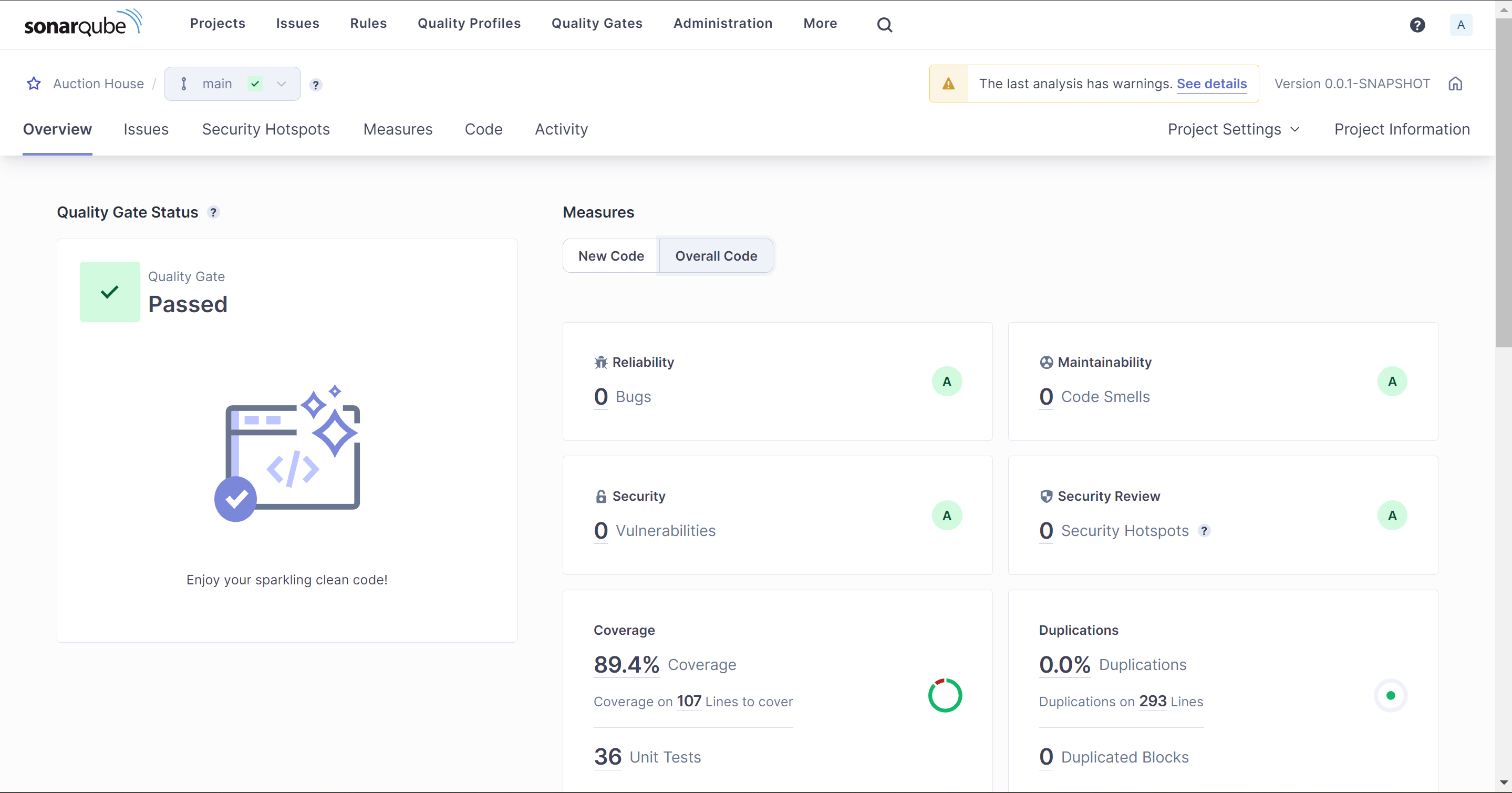
C4 – Level 3 Diagram:



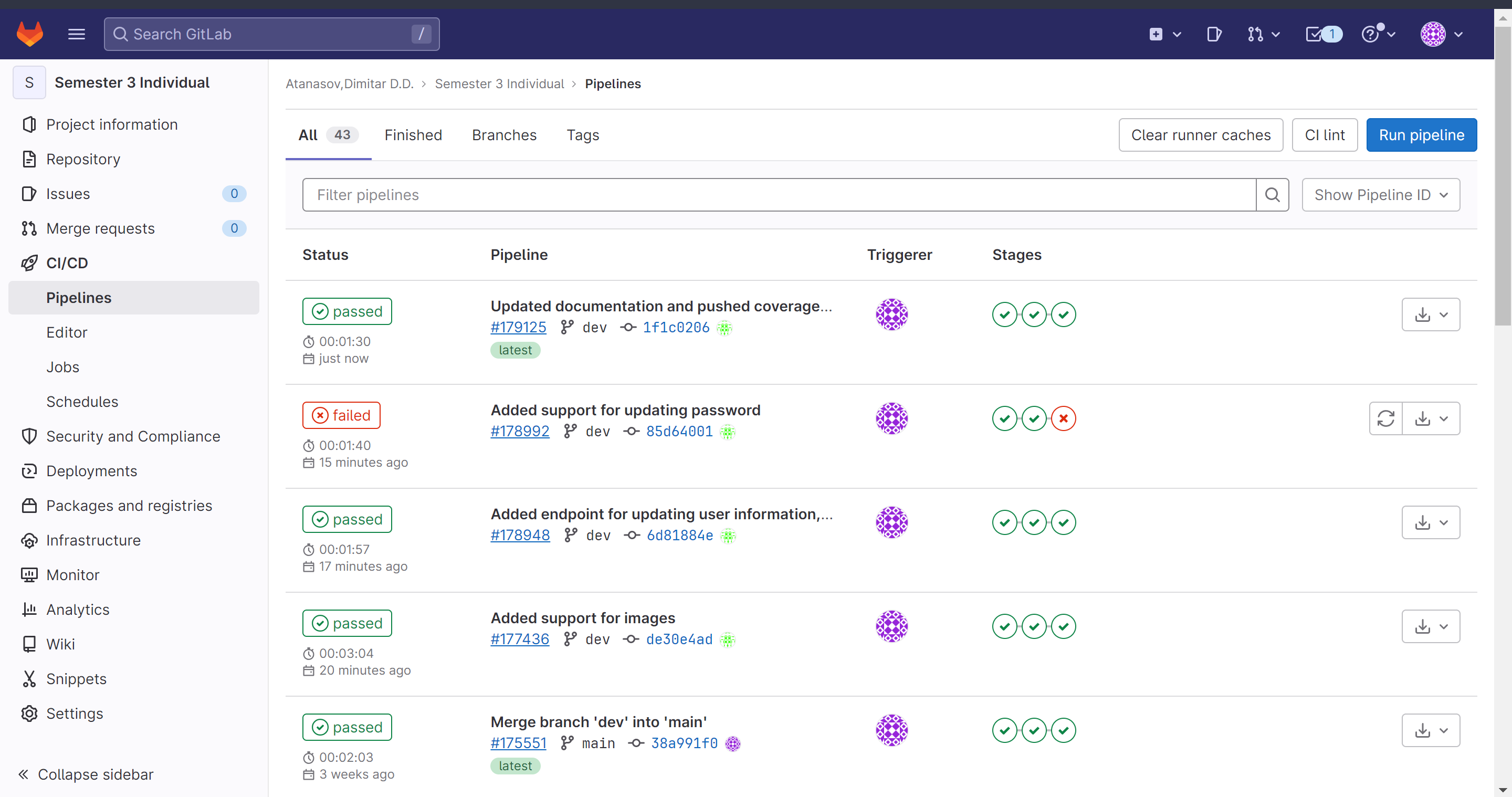
C4 – Level 4 Diagram:



SonarQube:



Git pipeline:



\* The last commit had 79.9% coverage, failing my quality gate and thus my pipeline. It was fixed with the commit that uploaded this document as well.

CI setup diagram:

