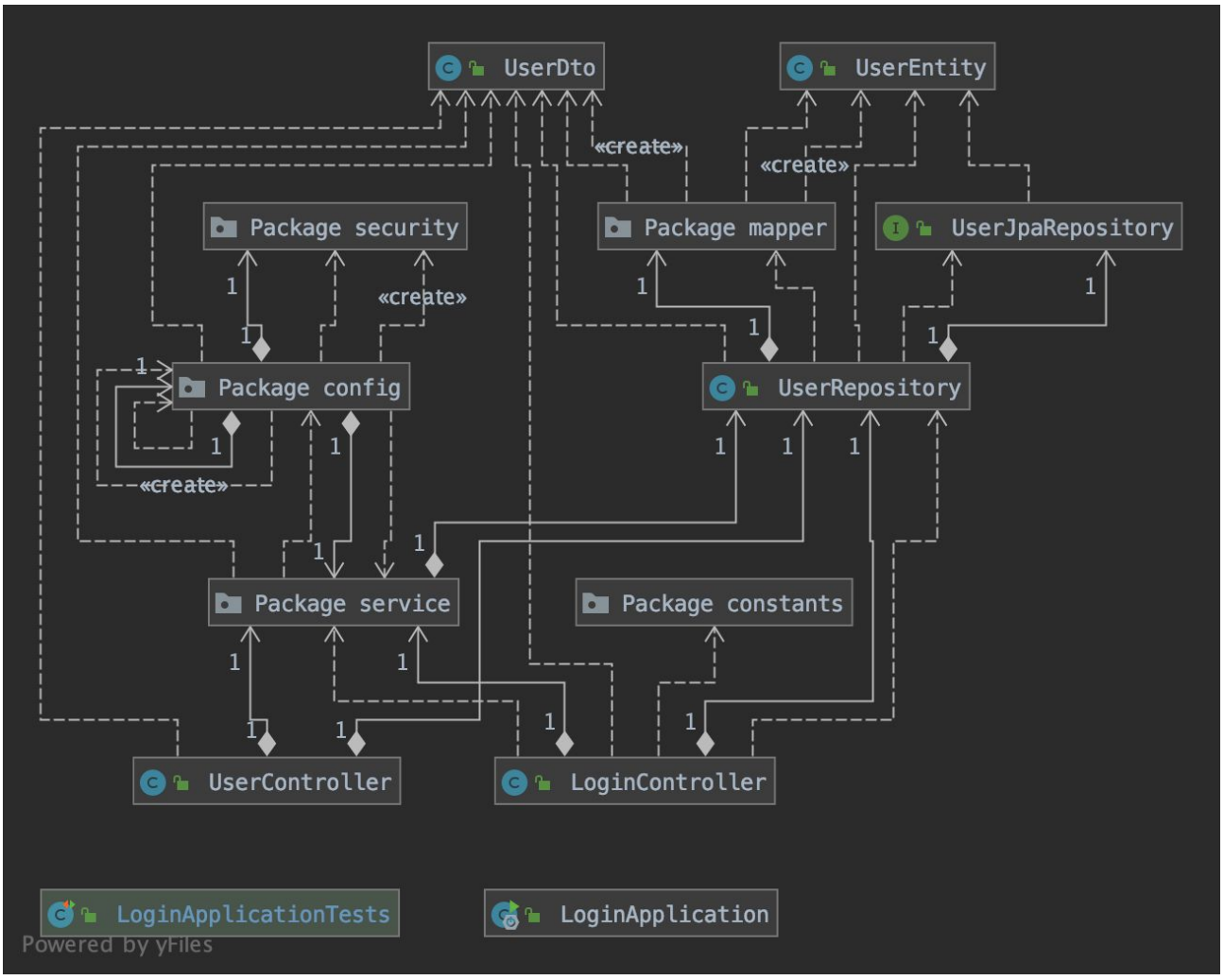
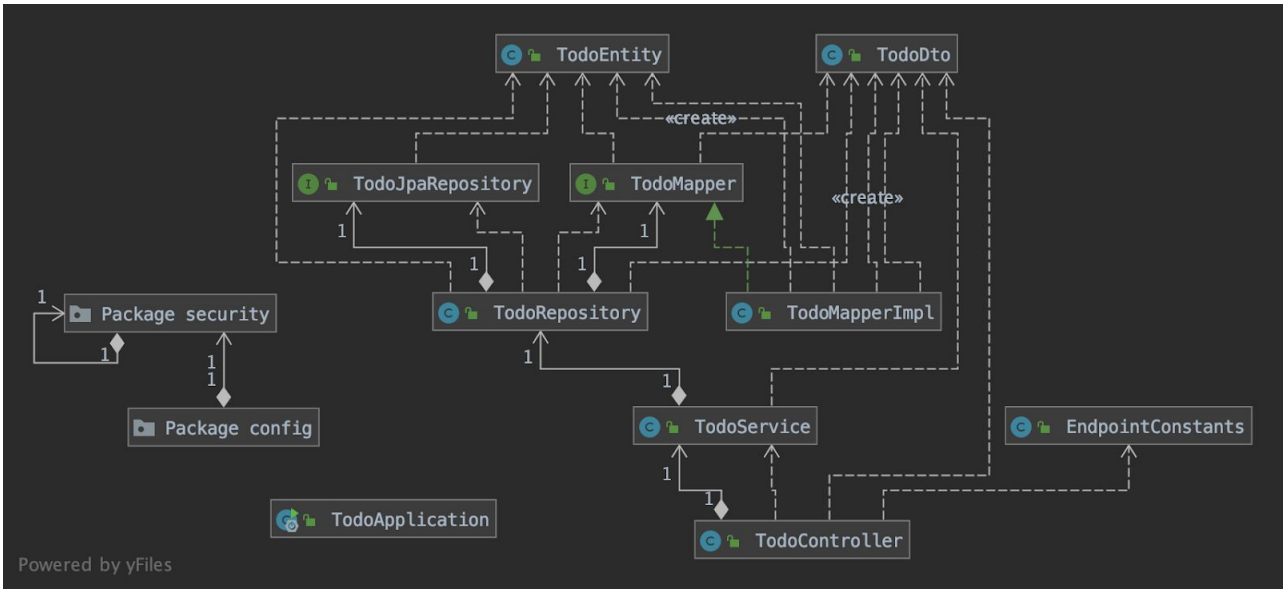
UML Diagrams

- UML diagram for Login Microservice



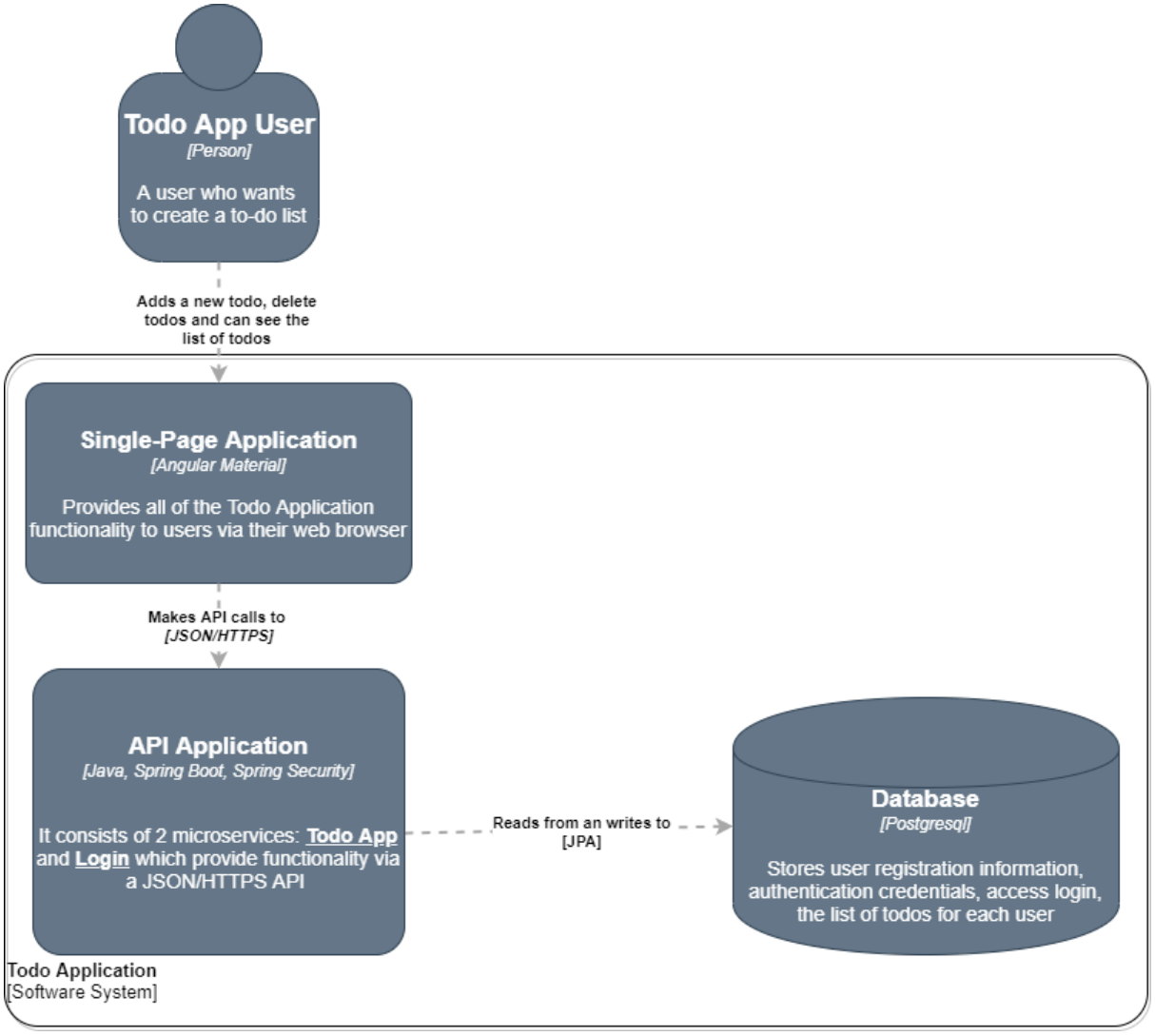
- UML diagram for TodoMicroservice



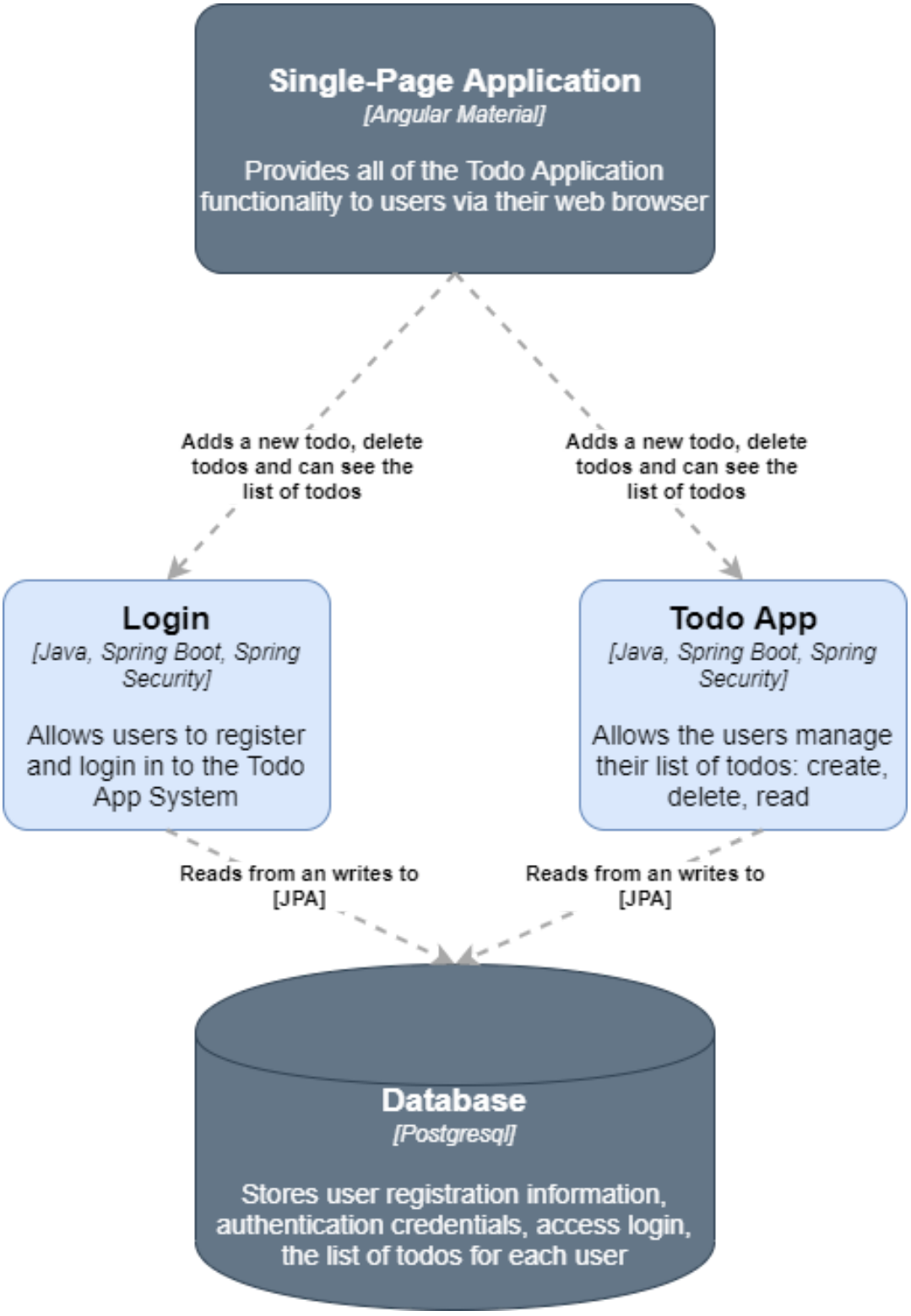
C4 models describing the system

- System context diagram

- Container diagram



- Component diagram



Microservices Architecture

The architecture based on microservices consists in the development of aapplication that contains small services, each having its own role and communicating through simple mechanisms, such as the HTTP protocol [1].

Microservices are an alternative to the monolithic application development system. This involves developing an application as a single unit. In a monolithic application, all components are strongly interconnected, making the application hard to modify and maintain. Any changes to the application will determine the process to create another version of the system. In Figure 1 the differences can be seen [1].

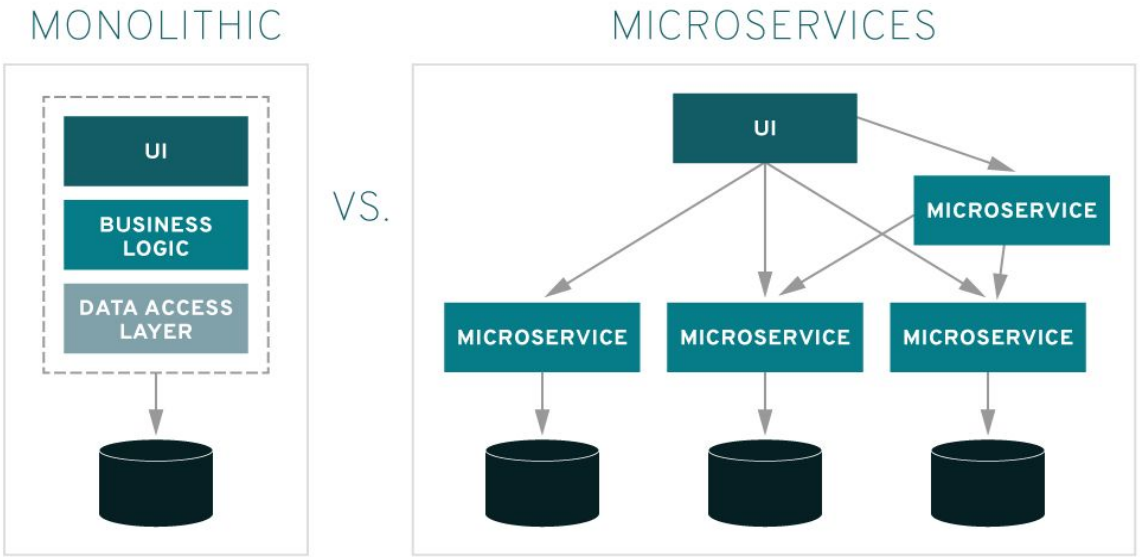


Figure 1. Differences between Monolithic and Microservice Architecture. Figure source: [2].

There are many advantages of using the microservices [1]:

- The application's components are independent, so if a failure occurs one can change a microservice without affecting the others.
- The apps can be deployed independently.
- The microservices can be developed by multiple teams.
- The code can be reused.
- Allows the developers to choose the most suitable technology for a given functionality.
- The microservices are just a small part from the entire functionality, so it's easier to understand.

In order to organize better the application’s components and layers, this project is created according to the client-server architecture. This architecture splits an application into two components: the web server and the back-end server.

The web server represents the front-end part and it is using the Angular Material. This contains all the components and services that are used to create the interface for the application.

The back-end part is formed from two microservices: Login and School. These microservices are independent and each one has its own implementation, but are working as a single unit. The web server is connected with both by sending requests in order to access the data.

The Login microservice manages all the user related operations including registration, login and security. It includes the User entity along with the dto, mappers, repositories, services and controller classes.

The TodoApp microservice has all the todo related operations in order to create and delete a todo and see the list with all todos. The security classes are also included in this microservice.

Bibliography

[1] - https://blog.webland.ro/2017/10/ce-este-arhitectura-bazata-pe-microservicii/
[2] - https://www.redhat.com/en/topics/microservices/what-are-microservices