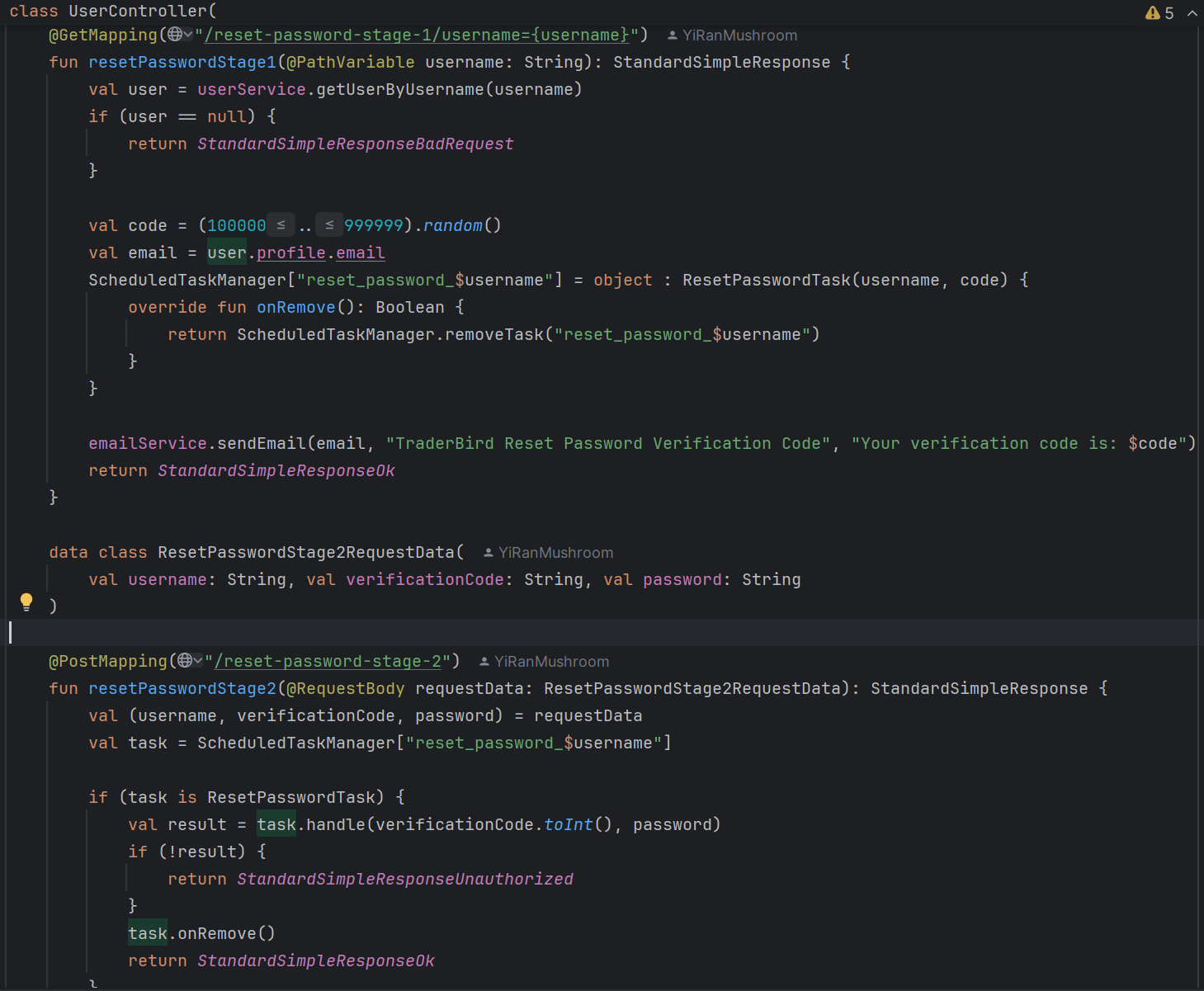
Dear Sathish,

I hope this message finds you well.

I am the designer of the backend for our project. I sincerely apologize that our team project cannot provide highly automated test cases. This limitation stems from several key challenges:

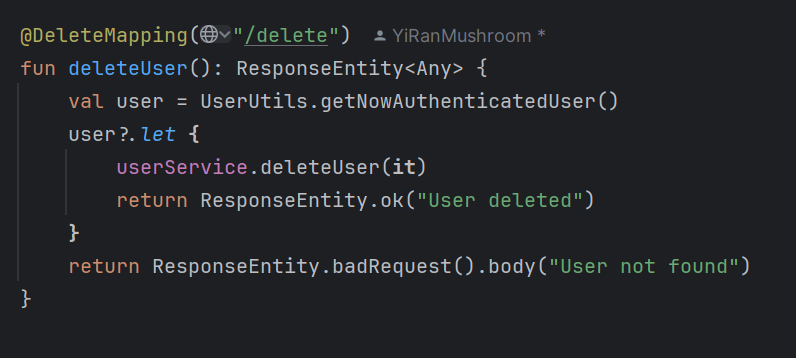
## Strong Context Dependency

Many of our components require a well-defined context to function properly. For instance, testing user password reset states depends on multiple intertwined contexts. Since these actions are atomic and context-sensitive, determining test success solely from function output is impractical.



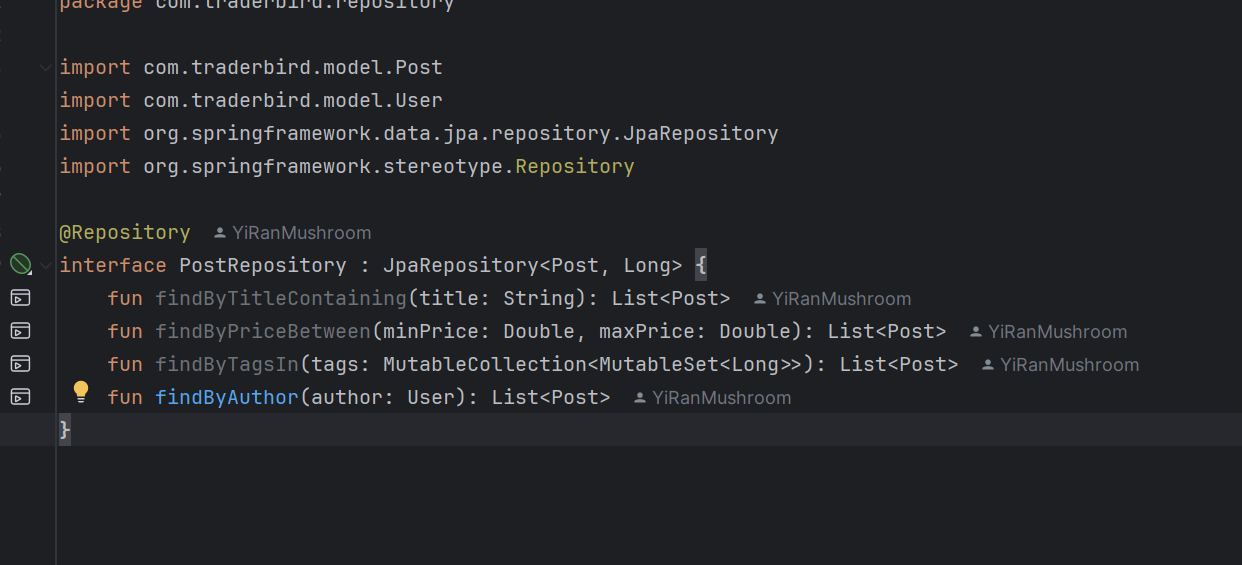
## Unrecoverable Side Effects

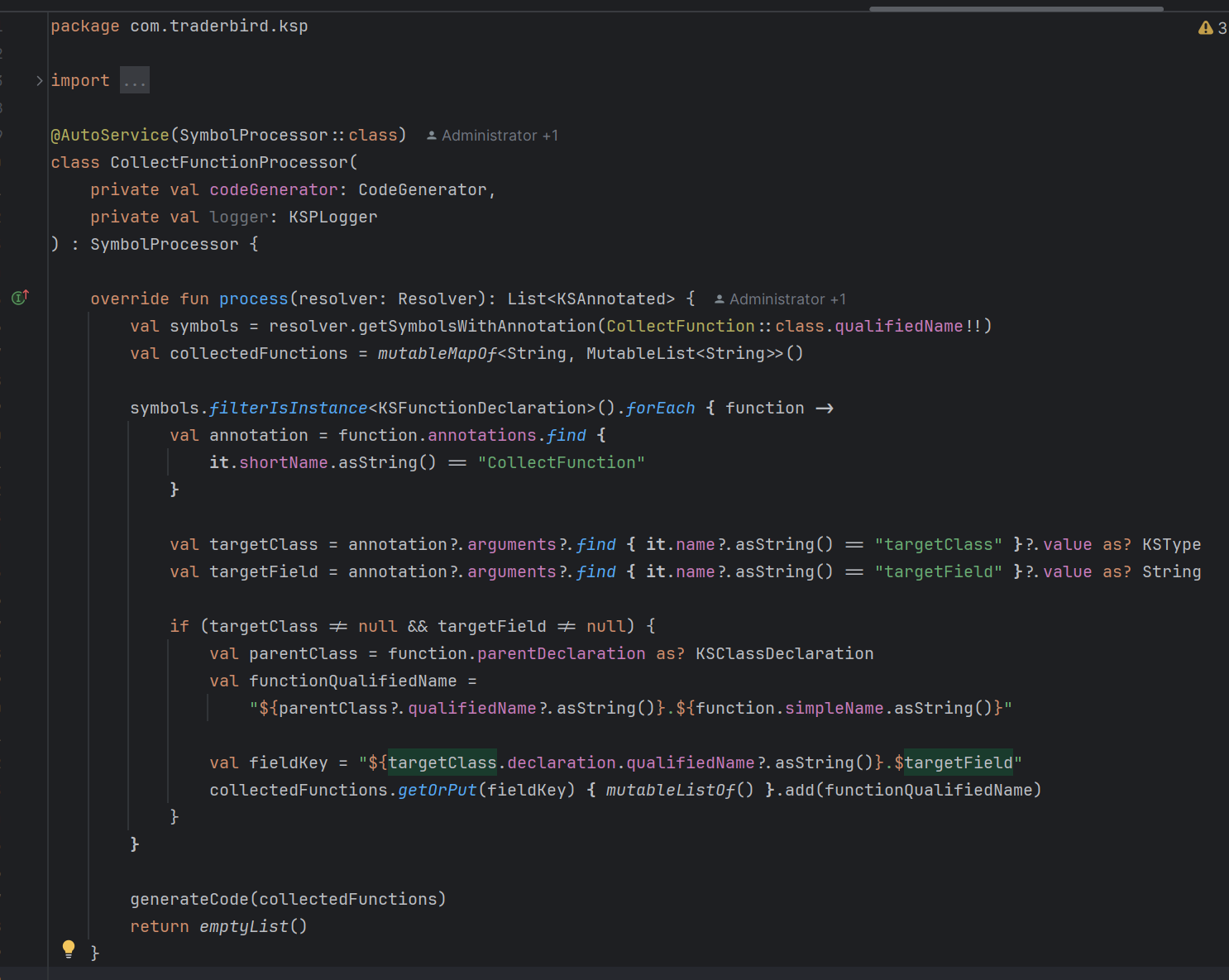
Many of our function calls can cause irreversible side effects, such as sending emails, permanently deleting files, or making unrecoverable changes to the database. These side effects complicate automated testing, as they cannot be rolled back or isolated easily.



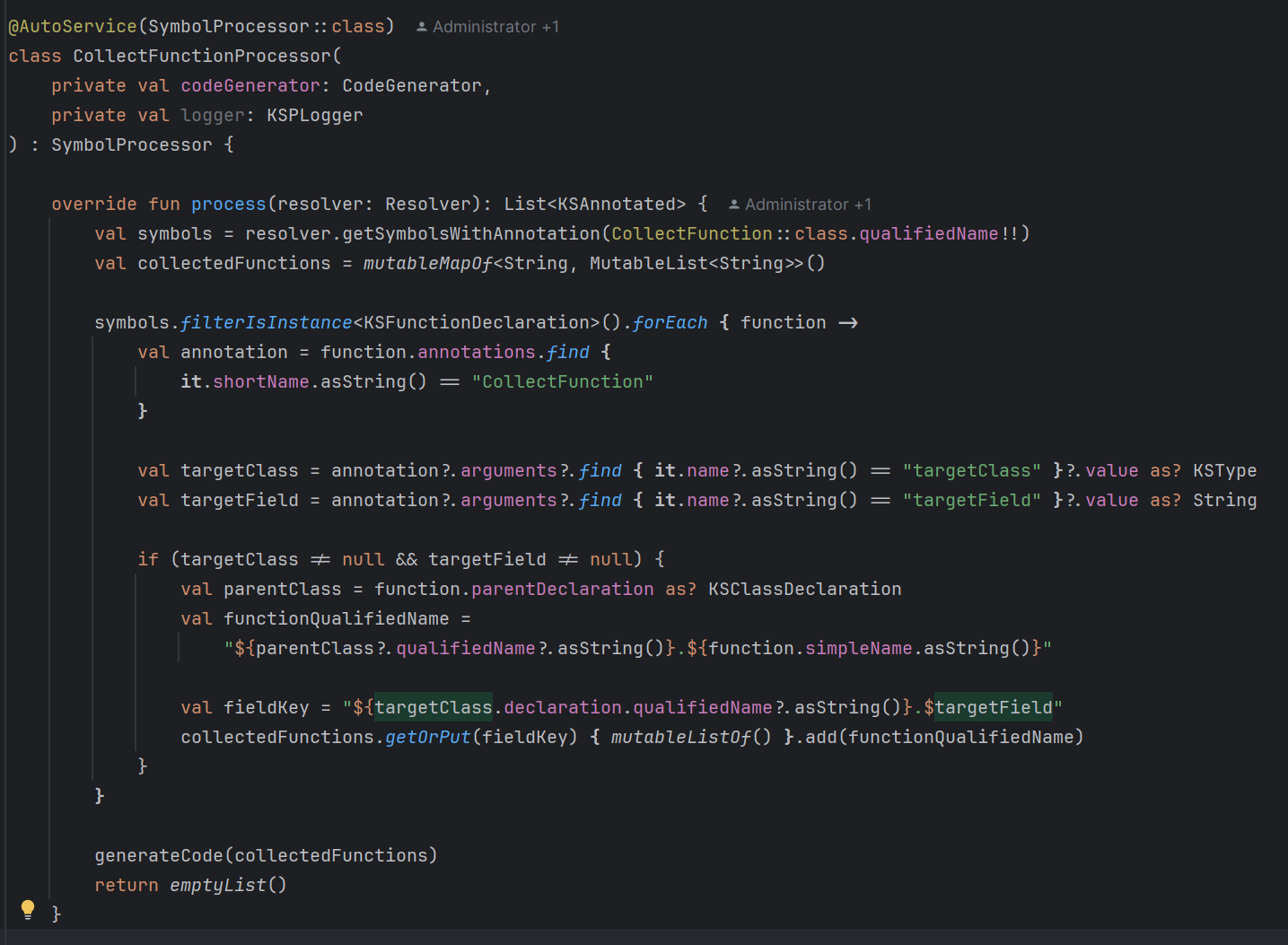
## Limited Access to Code and Framework Constraints

The project’s framework further complicates testing. Since we use the Spring Framework, much of our functionality relies on runtime mechanisms like singleton Beans and JPA repositories. For example, the PostRepository interface depends on runtime reflection to generate SQL queries. These Beans are automatically managed by Spring Boot, making it nearly impossible to manually initialize them or control their lifecycle outside of the application context. Furthermore, many of our modules depend on components like userService or entityManager, which are also framework-managed and cannot be directly instantiated.





Additionally, compile-time code generation by Kotlin Symbol Processing (KSP) adds another layer of complexity. The generated code implicitly modifies application behavior, making it infeasible to replicate or dynamically generate during testing.



## Manual Testing and Time Constraints

While Spring Test or other frameworks could partially address these issues, creating a robust test framework for such a large and complex project would require significant time and resources. Given our limited time and the scale of the project, we prioritized manual testing. All functionalities and branches have been thoroughly tested and verified to work as expected. Moreover, the project’s scope doesn’t involve complex mathematical or logical computations, further reducing the necessity for automated tests.

To summarize, our inability to provide detailed automated tests is due to the project’s inherent framework dependencies, context sensitivity, and the irreversible nature of certain operations. While we recognize the importance of automated testing, we kindly ask for your understanding and consideration of the unique challenges I faced as a backend developer. Given the short timeline and competing priorities, we hope you will take these constraints into account and reconsider some aspects of the marking criteria.

Thank you for your time and understanding.

Best regards,

Yiran.