**Vragen:**

* Welke belangrijke tradeoffs zijn er gemaakt?
* Waar zit het grootste riciso in deze architectuur?
* Met welke crosscutting concerns zijn er rekening gehouden?
* Hoe zou je logging in deze applicatie implementeren?

**Antwoorden:**

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * Separation of user interface, business logic and data access logic. (separation of concerns) * Changes can be made to a layer without having a large impact on the other layers | * Extra overhead (going through layers instead of a direct connection between parts) * Added complexity |

* 1. Everything makes use of the centralized database through the business layer. If the database fails so will the rest of the application.
  2. Transaction progressing, business rules and monitoring.
  3. Logging will be implemented on the database itself rather than on the application. For this I suggest making use of the transaction log already implemented in the database.