Allowed additional individual functionalities for Project NoSQL

Part O of the Rubrics:
The student has created an additional functionality that wasn't previously described as a requirement by the client (Garden Group).

is a conditional requirement for every group member. Because we get a lot of questions on what level of complexity of this individual part will suffice for passing, we have created a list with functionalities that are pre-approved by the consultants for NoSQL Project (Frank Dersjant en Thijs Otter).

The following individual functionalities can be chosen:

- Searching through incident/service tickets. The search functionality is able to search based on words that occur in the ticket (subject, content). To get alle points fort his functionality it is required to include AND + OR search and also to order the results by most recent on top.
- Filtering: a preloaded list of (incident) tickets can be filtered using (common) keywords. This is different from the search functionality because you will filter an already loaded list.
- Forget Password-functionality: the user should be able to reset their password using a password reset email + link. It is required that you use MVC framework (C#) or MVC pattern (Java, PHP, etc.) to implement this solution.
- API server (WARNING! FOR ADVANCED DEVELOPERS): build an API server module that makes your ticket application available to external systems. At least one of each GET/POST/PUT/DELETE query is required to be implemented.
- Escalate / close tickets: you can determine whether a ticket should be escalated (send incident to management) or closed, and the application should offer the functionality to do so.
- Sorting a ticket list: be able to sort the ticket list based on PRIORITY LEVEL of the tickets
- Transfer a ticket to another person
- Archiving the entire database (For example all tickets older than 2 years): with a simple click on a button, several entries before a certain date, are moved to a secundary (archive) database.

There are two restricitons when implementing an individual functionality:

1. You can not have two ore more members in the same group working on the same individual functionality. Because you know, it's INDIVIDUAL.
2. An individual functionality is always developed in a SEPERATE CLASS .

If you don't want to choose from the functionalities above, you can request to develop an individual functionality of your own choosing. Please email Thijs Otter (Thijs.Otter@inholland.nl) and we will determine the complexity and then approve or deny your request.