Implemented Product Features

Based on the previous documentation for the Microservices has already given an impression that we have some services ready. Each of these services works self-contained. In order to be able to test all the services together, there are still a few services that are planned to be implemented in the next few weeks. In addition, existing ones may need to be extended to ensure full coverage of all components planned in the architecture.
Below is a list of which of the defined requirements have already been implemented. Here a distinction is made between partially implemented or fully implemented requirements.

Table Of Content

1. Implemented Product Features
2. Table Of Content
3. Partially implemented Requirements
4. Fully implemented Requirements

Partially implemented Requirements

The following requirements are partially met, since the microservice is ready for this, but there is still no full connection to the system.

`/F100/` The system must allow the user to enter requests by text ~~or language~~

> Here are some NLU services that interpret the text. In order to generate an answer, we still lack a rule machine.We do not yet support voice input.

`/F200/` The ~~system~~ must be able to retrieve the Beuth Mensa menu for a specific day from the OpenMensa API
`/F201/` The ~~system~~ must be able to forward the menu from the OpenMensa API
`/F202/` The ~~system~~ must be able to filter and probe the menu according to the user's specifications

> Here we already have the microservice but since the system is not ready the requirement is not fully supported yet.

Architecturally, we have already started with some microservices, some of which are not yet full functional. An overview of the existing functions has hopefully given you the pre-recorded documentation.

Fully implemented Requirements

The following Requirements should be fully implemented our plant with our archtiecture.

/NF300/ The system should be as modular as possible
/NF301/ The system should be easily scalable
/NF302/ The system should contain easily replaceable components

After assembling it, we have found that our concept works and our implemented services work. The bot responds to corresponding requests.