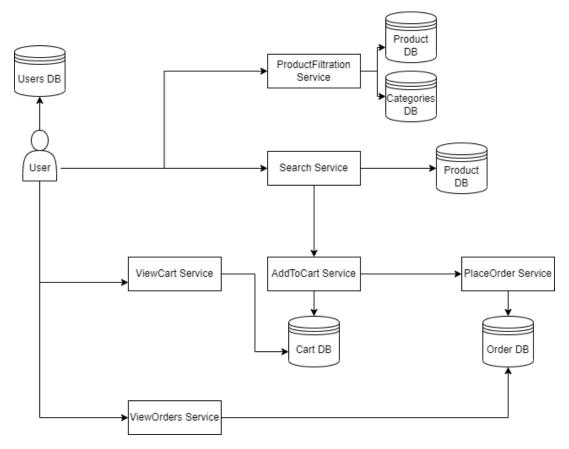
A Comparative Study: Monolithic vs. Microservices Architectures Aytan Gurbanova June 2023

The purpose of this study is to conduct a comparative analysis of monolithic vs. microservices architecture, with a primary focus on performance. To achieve this goal, a simple e-commerce application has been selected to be built in both architectural styles. The application will consist of five database tables and six main services. For better understanding of the architecture, a diagram has been included below. It is important to note that the application will only have a user-side functionality, as the administration-side will be directly controlled through the database for the sake of simplicity.



Currently, the microservices architecture has not been implemented as the decomposition process has not yet begun. However, I intend to start the decomposition process in a few weeks.

Since, the main goal of the study is to conduct performance comparison of two architectures, the research strategy is **quantitative.**

Based on the related papers I have analyzed so far, the performance compassion is done by analyzing metrics such as response time, number of threads, number of requests, and memory usage, among others. For data collection, my plan is to collect similar metrics once the applications are developed and deployed. Currently, I intend to use JMeter for this purpose along with the service logs, although I might consider alternative tools if a better option is found. Up until now, my main focus has been on application development, and data collection in my case requires both applications up and running, that's why I will initiate the data collection process after midterm.

As for data cleansing approaches, since I don't have access to the metrics data yet, I won't be able to provide any specific details. However, I believe that the output of performance measurement tools is typically clean and precise, and can be directly used in the paper. Therefore, I consider this step to be inapplicable in my study.