ERSMS - Movie recommendation system

| Role | Person |
| --- | --- |
| Leader | Krzysztof Rudnicki |
| Developer (from the biggest input to smallest) | Hubert Dwornik, Michał Sar, Michał Łezka, Jakub Mazur |
| System Architect | Jakub Mazur |
| Tester | Michał Łezka |
| Presentation Producer | Michał Sar |
| Librarian | Krzysztof Rudnicki |

We searched for the project on <https://mindmajix.com/microservices-projects>, we wanted something relatively easy to make to be able to actually finish what is for us the very first microservices project

Functionality:

1. User logs in or registers
2. Sees a grid of available movies to search through and filters to the side
3. Filters movies based on title, year of release, country of origin, genre
4. Can click on any movie and assign a rating to it (max 5 stars, min 1 star, 0.5 gradation)
5. Can click “recommend similar to this” button and receive similar grid as in the beginning, this time with movies recommended by our **simple** recommendation system

Microservices:

* Database microservice -> Retrieves data based on user input, filters and recommendation microservice
* OAuth -> Mandatory, authenticates user
* Intelligent Recommendation -> Based on user ratings on movies and their tags, recommends more movies to the user
* Web Application for displaying list of movies and potential filters we can use

Risk analysis:

Most important part of our solution is the **database**, if the database gets deleted, we have no data of user ratings, of movies and their tags, therefore we need to introduce automatic database backups, recovering database and saving database changes

Element stanu i usługi którymi zarządza

Cachowanie stanu dla przynajmniej 2 mikroserwisów

Następujące typy mikroserwisów

Ograniczone cachowanie między mikroserwisów planujemy dla tych 2 (których)?