WATER POLO SCRIMFINDER

Water polo is a niche sport in the Netherlands. Unlike football, hockey, or tennis where nearly each town might have one club of the 3 sports and big cities might have multiple within the same sport. This results in clubs easily finding opponents to play practice matches outside the main competition. Water polo on the other hand besides having a smaller group of athletes also has the problem that it is restricted to towns and cities with a swimming pool which is also competition approved. This combination leads to difficulty finding practice matches, the idea for this project is a integrated system where teams can find each other to play practice matches, contact each other, find relevant personnel like referees, keep track of matches themselves and create an environment which promotes competitive play among the regions.

Challenges

The system consists of multiple moving parts, contacting (either through email service or in application communication) and finding a match, agreeing on the terms of a match, a batch system where if more than 3 teams agree can create a league or tournament for practice, a live match updater for tracking match data, an online ranking system like ELO in chess, a location for match data to be stored and lastly if possible for my individual research into AI and ML I might integrate this into the match updating system.

Non-functional requirements

Most requirements will be researched further into when I know more about architecture and how they might relate to the final product.

- The system should be able to create a 1000 matches within X seconds.
- Email should be sent within 24hours when a match is scheduled or created.
- Matches when spectated should be updated within a minute when the referee or jury changes it.

Technology

The decision came down to Springboot Java and Django Python, Django allows for quick development cycles and easy deployment. While Springboot has a more robust security system and works better within smaller services. Front end work will be done in Vue.JS as it is what I have been more confident with and allows for easy single page applications.

For storage I will need to investigate what kind of database works best with batch/worker jobs but for static data I have a multitude of different solution with Springboots MySQL integration or cloud storage for match data.