## Challenges and Lessons Learned

# React

The React framework was used for several reasons:

* It is steadily becoming an industry standard for web applications
* React has components such as react-table and react-select that prevent developers from having to reinvent the wheel
* The developer of this project had limited exposure and it was an opportunity to explore the framework

Thus, the use of React was beneficial but also added a learning curve to development that required disciplined time management.

# Apache Karaf

The use of Apache Karaf allowed for rapid development, and could provide the front end, back end and SQL database in one polymorphic container. With this, the SQL database persisted the information and the system could be moved to a different machine without data loss. Karaf was also used so that administrators of the Gym System could easily deploy and manage the software to suit their needs. Karaf presented challenges at the beginning of the project because configuring the system to use SQL and Apache CXF took additional time. There is also limited documentation for this tool. Since it has been around for a considerable amount of time, some documentation was older and no longer relevant. Once the configuration of the system was done, the use of Karaf payed off because developers could build small parts of the system and deploy them without have to rebuild and reinstall the system.

# Conclusion

In the final weeks of the program, validation and the ability to manage different branches of the Gym System (ex: multiple gyms in one system) were added. These features were quite large and required some massive changes that were not foreseen. If the use cases and the feedback from them provided by the instructor were considered soon and at every increment, the developers could have added code with these new features in mind.

Given that some of the tools used required additional learning, the main lesson learned from the project was to use tools that developers already have familiarity with, save one or two new ones. This way, the learning curve is not as steep, and developers are not overwhelmed at the onset.