

DuelHub

Detailed design

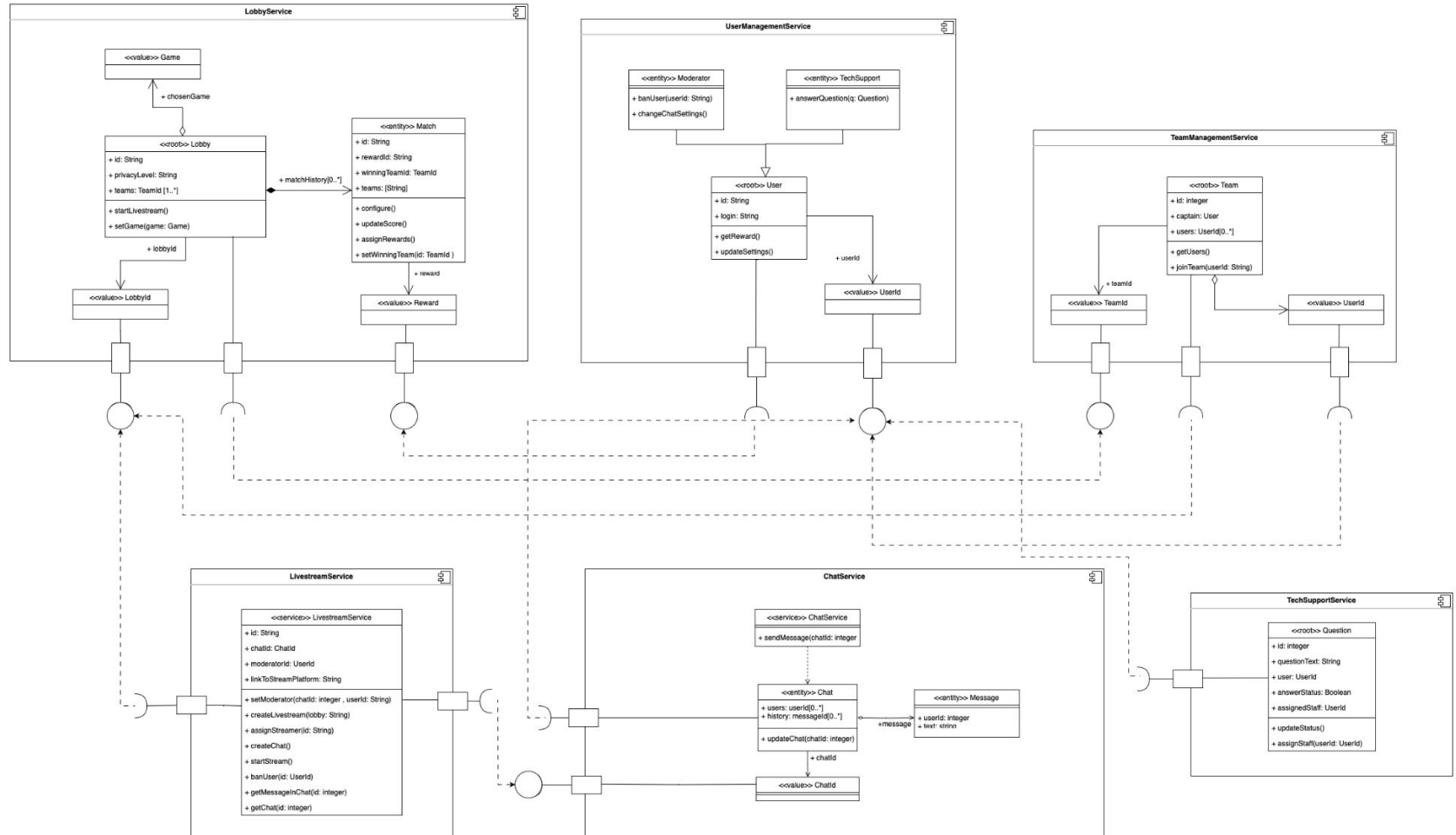
Product description

DuelHub is a web application, that provides a convenient online space for gamers to coordinate and engage in competitive matches across a wide range of computer games. By facilitating the organization of duels, our product ensures that players can easily connect with opponents and enjoy thrilling gaming experiences.

Team: Lyudmila Rezunik, Teona Sadulaeva

Repo: <https://github.com/teopalmer/duelhub>

System architecture

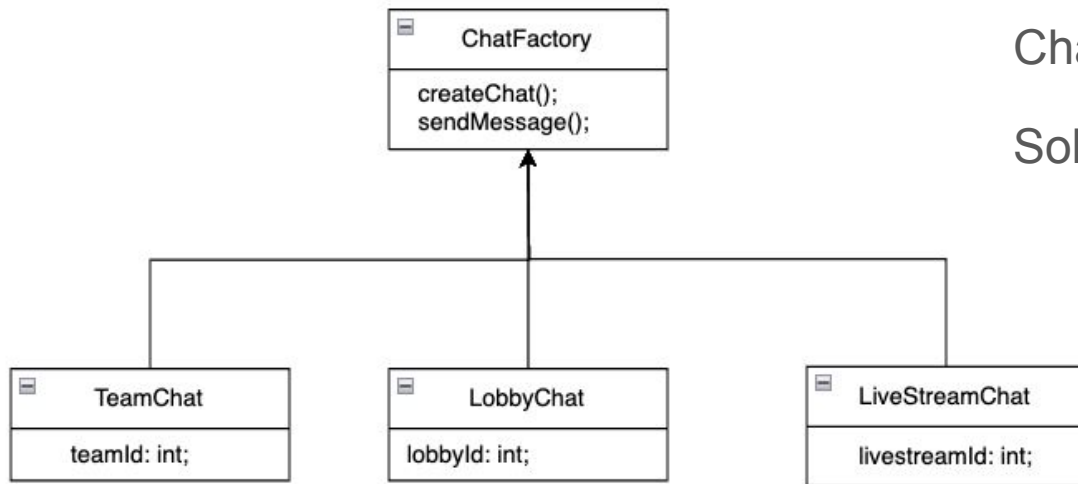


Factory pattern for messages

All chats need to share the same functionality, new types of chats should be added easily.

Chats need to be interchangeable.

Solution: Apply factory pattern

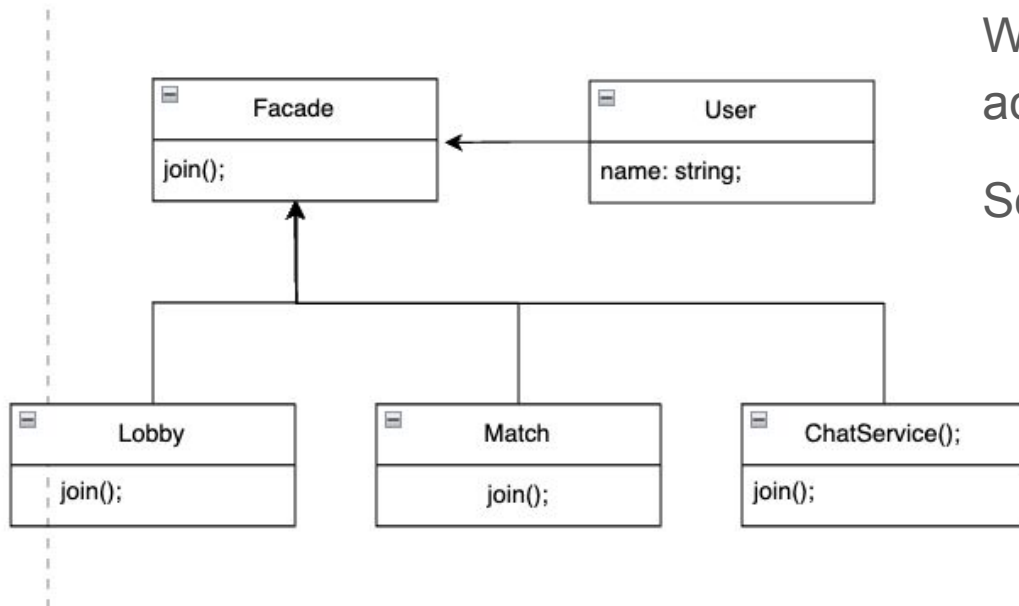


Facade pattern for joining matches

When user joins a match, he also joins lobby and its chat.

We need to ensure that all of this is accomplished.

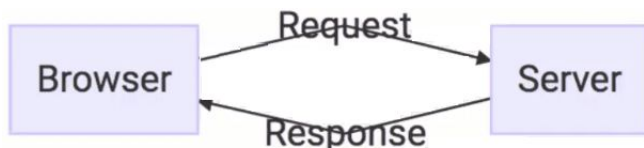
Solution: Apply facade pattern



REST API

As we apply REST architecture style we need to conform to principles to make our system RESTful

1. Uniform interface
2. Stateless technology
3. Layered system
4. Caching
5. Code on demand



Team Module

GET	/team	📋 ↩️ 🔒
GET	/team/player/{login}	📋 ↩️ 🔒
DELETE	/team/delete/id	📋 ↩️ 🔒
POST	/team/create	📋 ↩️ 🔒
POST	/team/matchmake	📋 ↩️ 🔒
PATCH	/team/edit/{id}	📋 ↩️ 🔒

Lobby Service

GET	/lobby	📋 ↩️ 🔒
GET	/lobby/player/{login}	📋 ↩️ 🔒
DELETE	/lobby/delete/id	📋 ↩️ 🔒
POST	/lobby/create	📋 ↩️ 🔒
PATCH	/lobby/edit/{id}	📋 ↩️ 🔒
GET	/lobby/spectators/join/{id}	📋 ↩️ 🔒
GET	/lobby/spectators/{id}	📋 ↩️ 🔒
POST	/lobby/join/{id}	📋 ↩️ 💡

SOLID

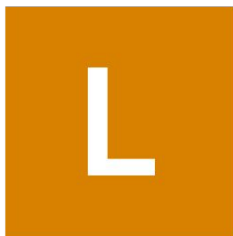
Single
responsibility



Open-Closed
Principle



Liskov
substitution



Interface
segregation



Dependency
inversion

