Data Access Object (DAO)

Presented by Team 5

What is DAO?

- The DAO Design Pattern involves encapsulating all aspects of your data storage by providing an interface to it.

- This allows you to change database details while you scale up your app without breaking the rest of your code.

Example Usage of DAO

- Imagine you had a Train Companies app that let you get companies and create more companies. The DAO interface could be:

```
public interface TrainCompanyDAO {
    public TrainCompany createInstance(String name);
    public TrainCompany getInstance(String name);
}
```

How is this interface implemented?

- You could use memory storage or text files as you start up. Once you start getting a large number of users you might switch to an SQL database.
- Most importantly, the rest of the code does not care about the implementation! It only sees the two methods in the interface.

DAO usage in our Project

- We separated our backend and frontend from the start, allowing us to transition from text files to a database with ease.

- We used interface methods such as insertUser(), insertNewTopic(), getAllTopics(), etc.

When should I use DAO?

- If you ever foresee yourself having to change how your data storage works, consider investing the time to set up DAO.

- Alternatively, if you use the Model-View-Controller (MVC) design pattern you essentially get DAO for free. (as Model)