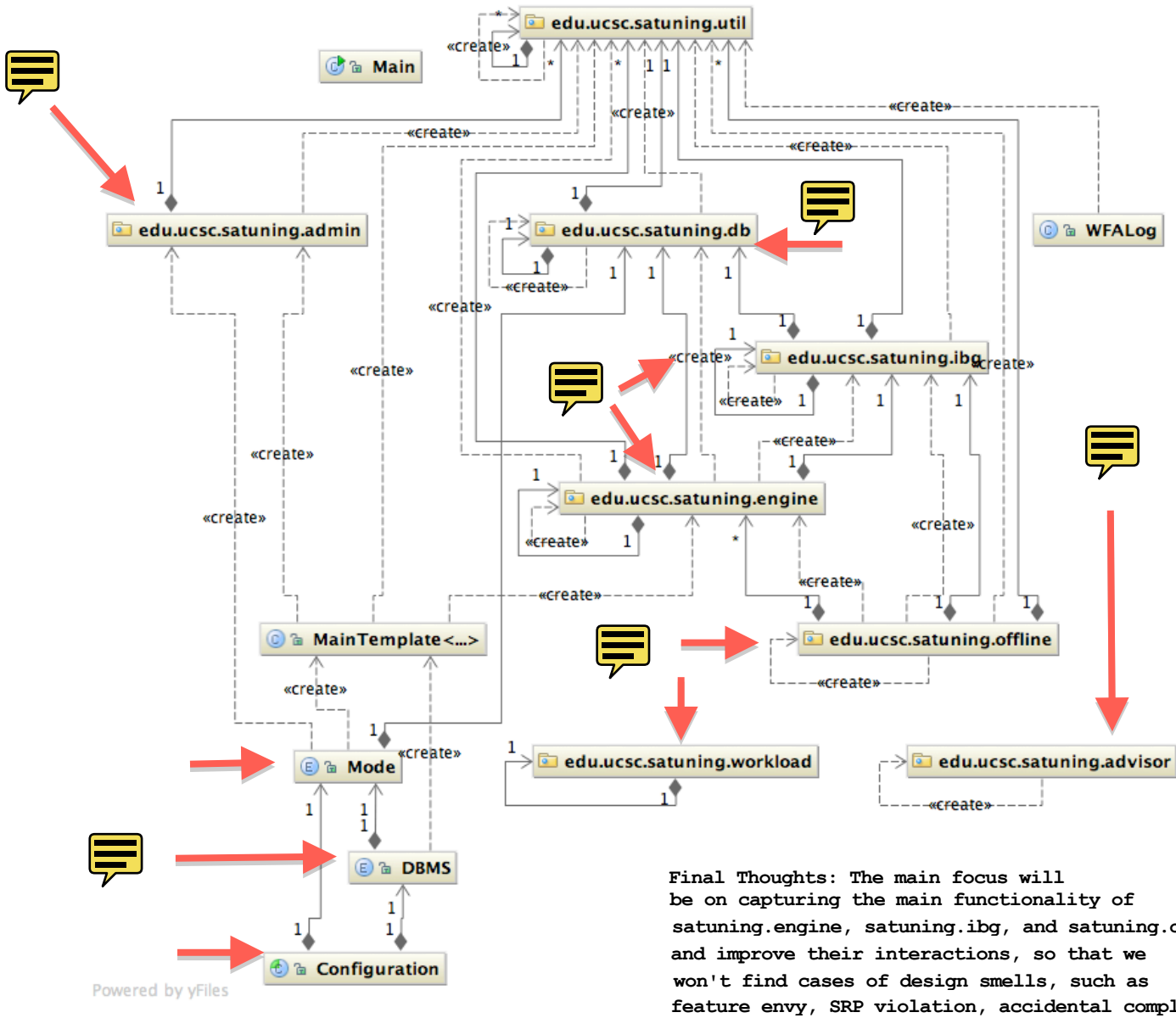
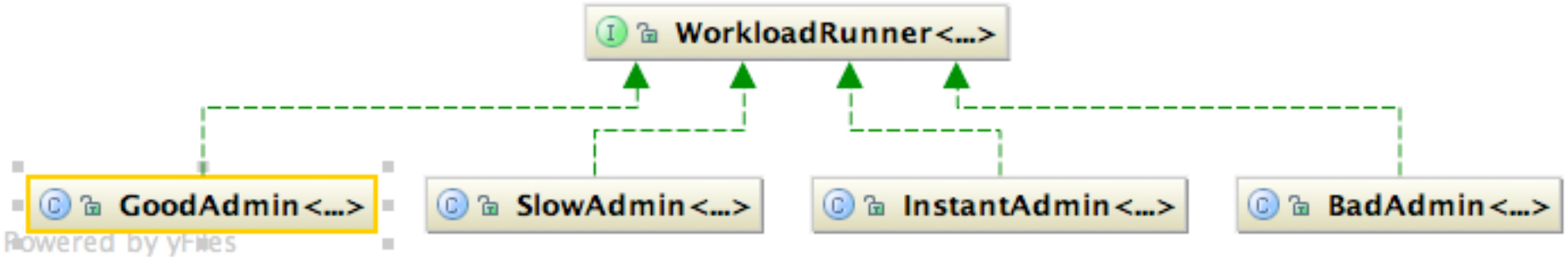
Satuning's Current Package Structure (**edu.ucsc.satuning**)



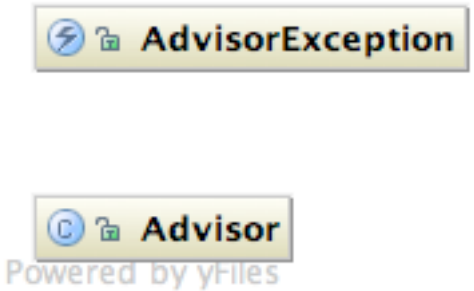
Final Thoughts: The main focus will be on capturing the main functionality of satuning.engine, satuning.ibg, and satuning.db and improve their interactions, so that we won't find cases of design smells, such as feature envy, SRP violation, accidental complexity.

Next step will be to redesign how clients of this api (client = other frameworks) should interact with our API. Most likely this interaction will be designed from scratch.

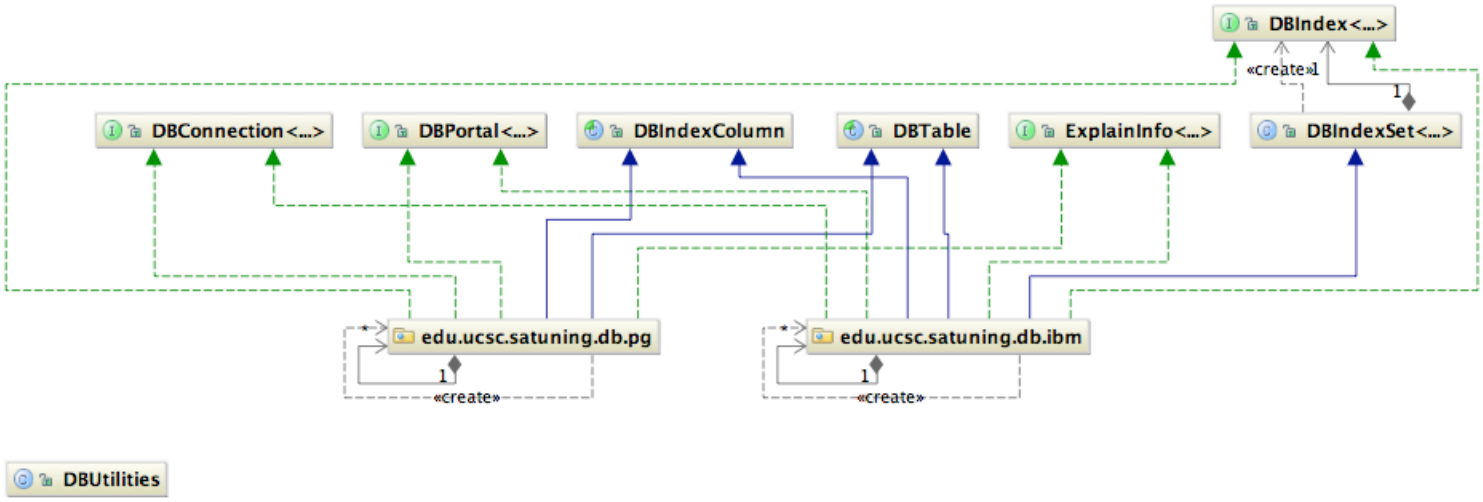
edu.ucsc.satuning.admin



edu.ucsc.satuning.advisor



edu.ucsc.satuning.db

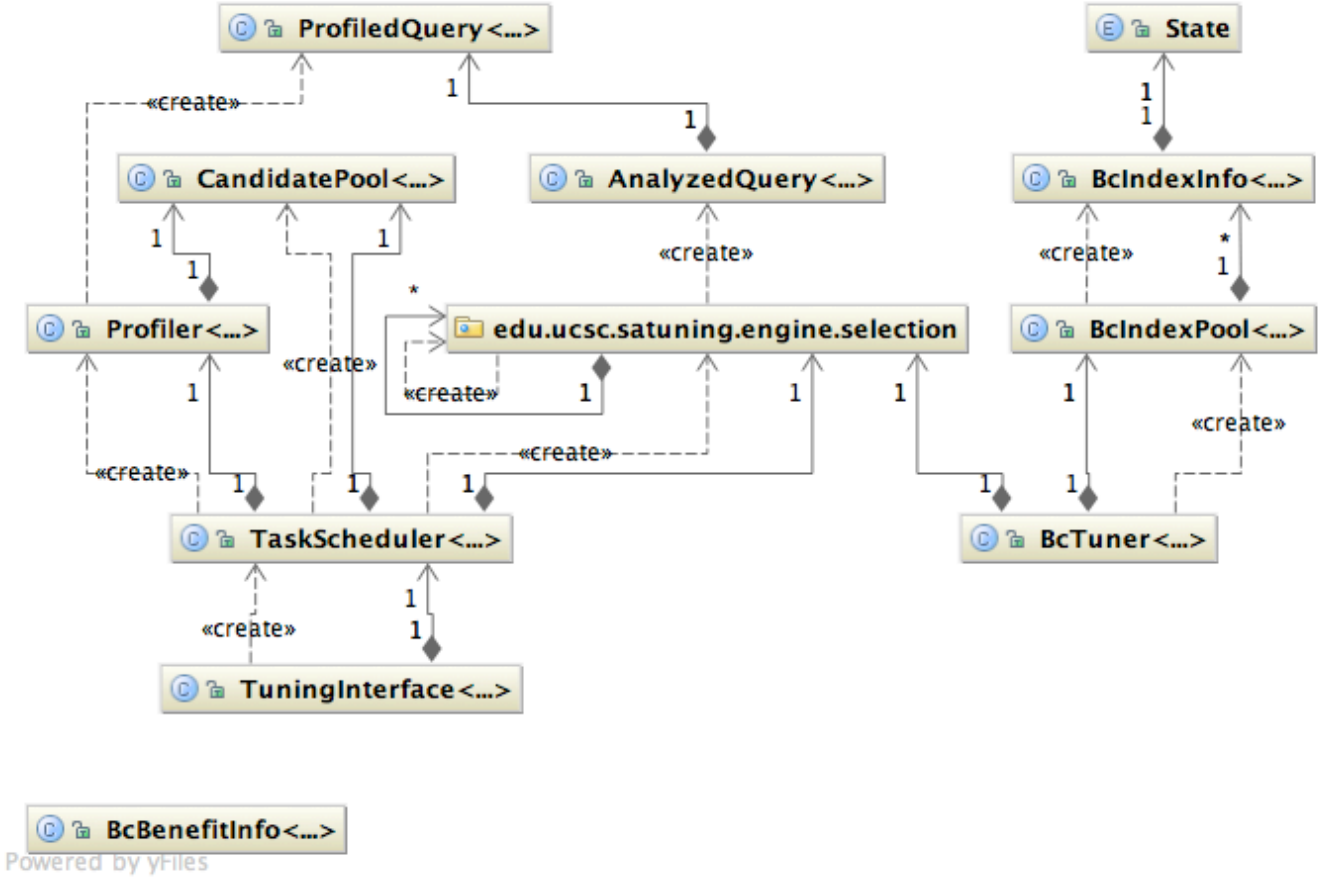


Note: the existing sub-packages represent DBMS-specific implementation of all of this package's interfaces/types.

Ideas?

1. **Redesign type/class hierarchy**
    - 1.1. **Extract functionality dealing with WhatIf Opt. from DBConnection**
    - 1.2. **Extract Session Specific Functionality from DBConnection**
    - 1.3. **Create a separate type that knows the params needed to create a database instance.**
2. **Get rid of copy/paste functionility, found in db.pg and db.ibm, by merging into a single class.**

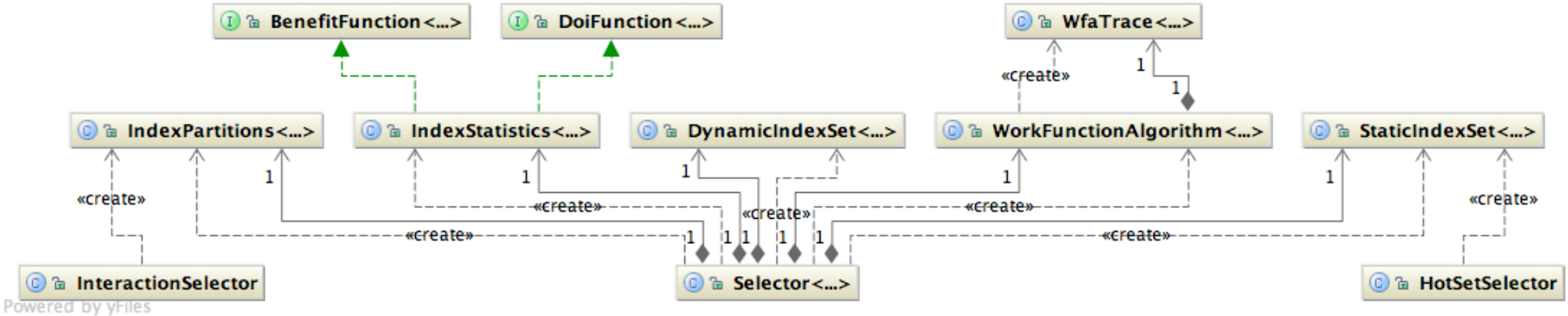
edu.ucsc.satuning.engine



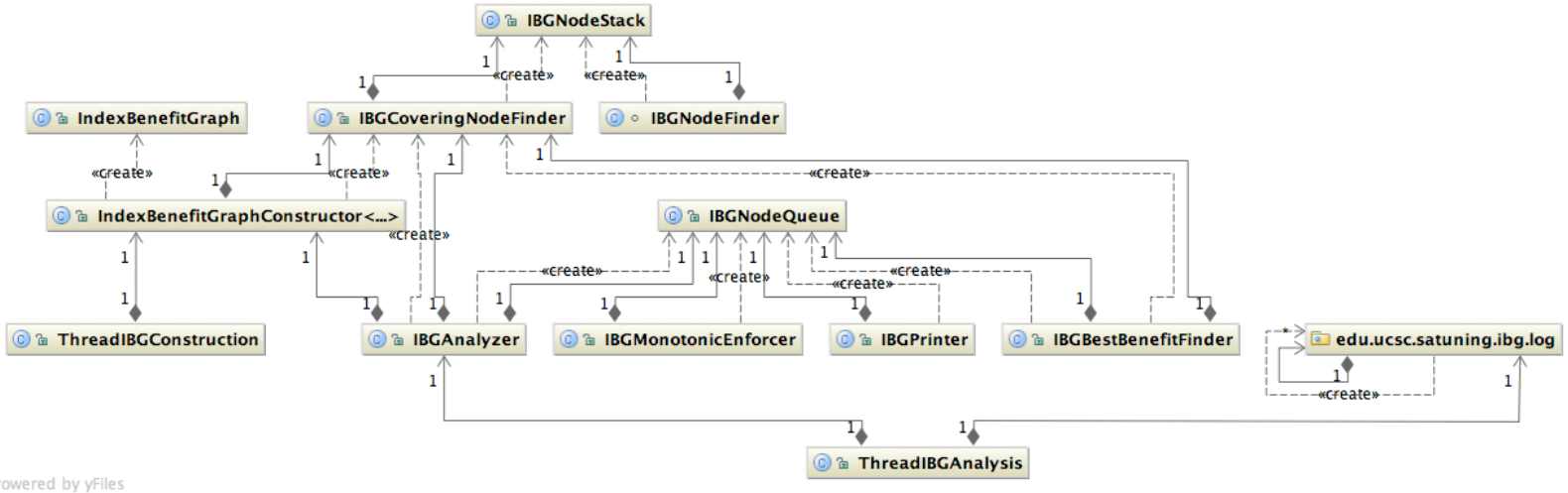
Ideas?

1. **reduce coupling between classes (e.g., minimize cyclic dependencies, use Inversion of Control design techniques)**
2. **Minimize concurrency errors/bad practices currently found in the API.**
3. **Test, Test, and more Test**

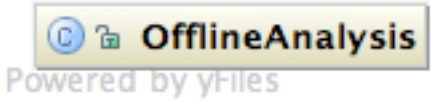
edu.ucsc.satuning.engine.selection



edu.ucsc.satuning.ibg



edu.ucsc.satuning.offline



edu.ucsc.satuning.workload