

Design Mini Paper

Baheem Ferrell

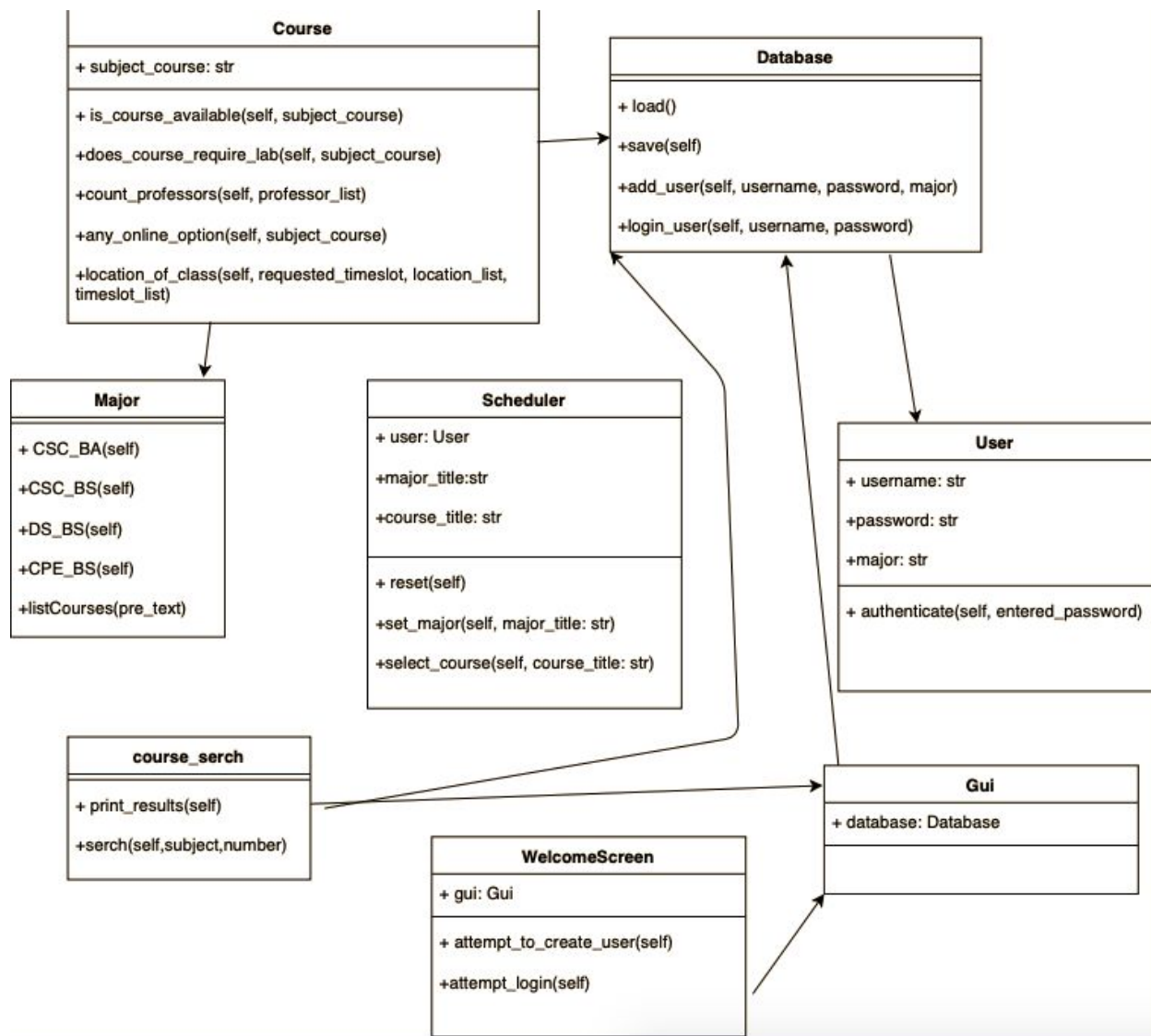
Facade is the design pattern where we can achieve any complex and difficult problem easily. The design patterns don't solve the problem that is structure predefined using themselves, they help us solve the problem in a very easy way that is the purpose of the design pattern we will see in detail one by one. Software development design is the object-oriented patterns started as best practices that we can apply again and again to the same and similar problems encountered in different contexts and different ways. Examples of common problems solved by the design pattern: How to instantiate an object properly for example interface and abstract class? How to interact between two objects and multiple objects?

The design pattern is a solution methodology approach to a common standard problem, in software engineering, a design pattern SDLC software development life cycle is a general repeatable solution again and again the design pattern commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code level. It already predefined template and description or template for how to solve a problem the easy way that can be used in many different situations and different programming languages. What are all the benefits of the design pattern? Using design patterns can speed up the program and development process and a good quality manner by providing tested, proven development paradigms. We can be reusing the design patterns to help to prevent any difficult issues that can cause major and big problems and it also improves code readability and maintainability. A design pattern provides general solutions and allows developers to communicate well-known and proper documents in a format.

Structural design means how to make relationships from one class to another class, how to define structure between multiple classes and objects. How to inherit a class and extend a class that is the purpose of a structural design pattern. A class is an entity where we can write anything. It is a logical entity and object using class we can create multiple objects and objects are also entities we can write anything. There are multiple kinds of method user-defined methods and pre-define. Users define methods we can call through the class object. This is the complete structure of the design pattern.

Lastly I am sure we are wondering how this applies to our project. It allows us to break every bit of code into small pieces for example; the scheduler and the user classes could have been combined in code but for the reuse of the code and partnership it would be better to keep it separated. Per say the application were to grow into the schools database, unfortunately the combining both of these parts of the code

would make the process slow and bulky to operate. Also, separating things that have nothing to do with one another such as the course search away from the database. Not because they are not connected but the smaller the pieces are the easier it is to read and operate it among one another.



Reference

1. Guizzo, G., Colanzi, T.E. & Vergilio, S.R. "Applying design patterns in the search-based optimization of software product line architectures." *Softw Syst Model* 18, 1487–1512 (2019).
<https://doi-org.uri.idm.oclc.org/10.1007/s10270-017-0614-9>

This article speaks about the use of different design patterns being used for search-based optimization. The article is from 2017 as far as the currency is involved. I am using this art because of its optimization and facade reference to my use of the pattern I am using for my mini paper. Two of the three authors are from the Department of Informatics at the Federal University of Parana in Brazil. I don't see any errors in the writing and the tone doesn't seem to have any bias in it. The purpose of this information is to inform you of the findings.

2. Caballero, Carlos. "Understanding Design Patterns: Facade using Pokemon and Dragonball Examples!" *Dev*, 5, Apr. 2019,
<https://dev.to/carlillo/understanding-design-patterns-facade-using-pokemon-and-dragonball-examples-5868>

This article speaks directly to the usage of facade patterns and shows an example of it in the javascript language. Also, implement it in UML diagram to show it for understanding before the coding aspect. This Dev article was created on April 5, 2019, as far as currency. The utilization of this was based on its examples for the diagram examples and its intended audience is new students and other developers looking to enhance their code methods and reusability. This information is used to teach and inform with no influence biases.

3. Serrato-Barrera, R., Rodríguez-Gómez, G., Pérez-Sansalvador, J.C. et al. Software system design based on patterns for Newton-type methods. *Computing* 102, 1005–1030 19, Sep, 2019,.
<https://doi-org.uri.idm.oclc.org/10.1007/s00607-019-00759-8>

This article speaks to a wide range of engineering applications that use optimization techniques as part of their solution process. The article was published on September 10, 2019, and has been peer-reviewed. I would say the link is intended for at least the college level for some of the formulas to incorporate fully. This article has been

peer-reviewed and doesn't seem to have any spelling errors. I don't recall there being any biases I can recall and the information is to inform.