T. Kelly, A. Kuang, M. Lee, J. Sizoo, and M. Swartz (Group 14)

Dr. David Johnson

EECS 448

Due: Sunday, April 3, 2022

Project #3: Design Pattern

Design patterns are blueprints which describe common solutions to problems that arise in software development. Design patterns are divided up into three categories: creational, structural, and behavioral. Our group's design prototype, a student and teacher resource management system, utilizes design patterns from each of the three types. However, as the project evolves it is likely that additional design patterns will be utilized.

Creational design patterns allow for multiple object creation mechanisms. Our project utilizes the Singleton design pattern, which provides universal access to an instance, while ensuring that a class has only one instance. For example, many of our data querying methods begin by establishing a single connection to the database, accessed using SQLite. This provides access to the information in the database, which can then be manipulated to present the desired information to the user.

Structural design patterns assemble objects or classes into larger structures. Our project utilizes the Bridge design pattern, which allows systems to be split into separate hierarchies, one governing abstraction and one governing implementation. Our management system operates using this split approach. The user interface layer, in which students and teacher physically interact with the resource management system, is separate from the logic and data layers, which access and manipulate the information as directed by the user.

Finally, behavioral design patterns are mostly concerned with algorithms and the relationships and division of tasks between objects. The chain of responsibility design pattern is used heavily within the Window Forms that make up the user interface. All the graphical elements, like buttons or drop-down menus, fire off event handlers upon interaction and pass along the necessary information to the next handler in the chain. Additionally, the iterator design pattern is used frequently when accessing the database. Information is accessed via lists, which can then be traversed to perform the necessary data manipulation (ex. looking for a class or student).