

Section 1: Design Pattern

Bridge Pattern is what I selected. This is because the project has two separate dimensions of change between the user interface and the system logic (Boundary vs Control) and also between the Catalogue and the user. Without Bridge, the problem is that everything would be tightly coupled and the design would be harder to maintain and update in the future. The Bridge solves this by decoupling implementation and abstraction. This allows each side to grow independently which is great for maintainability and extensibility since new interfaces can be used without modifying the main logic. It supports reusability by allowing the same code to function behind different user interfaces/accounts.

Section 2: Class Diagram

Figure 1: Design Pattern Bridge to be used in implementation of Library System

