|  |
| --- |
| Library (Singleton) |
| -Instance : Library  -users : List<User>  -courses : List<Course>  -books : List<Book> |
| +getInstance() : Library  +addUser(user:User)  +addCourse(course:Course)  +addBook(book:Book)  +assignCourseToUser(user:User,course:Course)  +borrowBook(user:User,book:Book)  +returnBook(user:User,book:Book)  +notifyingUsers(message:String)  +showCourses()  +showBooks() |

|  |
| --- |
| User (Abstract) |
| -name : String  -course : Course  -role:UserRole  -borrowedBooks:List<Book> |
| +getName():String  +getCourse():Course  +getRole():UserRole  +borrowBook(book:Book)  +returnBook(book:Book)  +notify(message : String) :void |

|  |
| --- |
| <<Enum>> UserRole |
| +STUDENT  +TEACHER  +LIBRARIAN |

|  |
| --- |
| UserFactory |
| -userRegistry : Map<String,User> |
| +getUser(name:String,role:UserRole,department:Department):User |

|  |
| --- |
| Librarian |
| +notify() |

|  |
| --- |
| Student |
| +notify() |

|  |
| --- |
| Teacher |
| +notify() |

|  |
| --- |
| Course |
| -name : String  -code : String  -bookCategories():List<BookCategory> |
| +getName() :String  +getCode() : String  +getBookCategories():List<BookCategory> |

|  |
| --- |
| PaymentContext |
| -paymentStrategy: PaymentStrategy |
| +setPaymentStrategy(strategy:PaymentStrategy):void  +pay():void |

|  |
| --- |
| PaymentStrategy |
| +pay():void |

|  |
| --- |
| Book |
| -title : String  -author :String  -category : Bookcategory  -availability : boolean |
| +getTitle() : String  +getAuthor() :String  +getCategory() :BookCategory  +isAvailable() :boolean |

|  |
| --- |
| BookCategory |
| Fiction |
| Non\_Fiction |
| CSE |
| ECE |
| EEE |
| CIVIL |
| MECHANICAL |

|  |
| --- |
| CreditCardPayment |
| +pay():void |
| CashPayment |
| +pay():void |
| OninePayment |
| +pay():void |

BASIC STRATEGY BEHIND LIBRARY MANAGEMENT SYSTEM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SNO. | CLASS | OBJECTS | METHODS | DESIGN PATTERN | COLLECTIONS |
| 1. | Library | >Instance of Library  > List<User> - users  > List<Course> - courses  > List<Book> - books | > getInstance()  > addUser(user:User)  > addCourse(course:Course)  assignCourseToUser  (user:User,course:Course)  > addBook(book:Book)  > borrowBook  (user:User,book:Book)  > returnBook  (user:User,book:Book)  >notifyingUsers  (message:String)  >showCourses()  >showBooks() | Singleton |  |
| 2. | User | > name : String  > course : Course  > role:UserRole  > borrowedBooks:List<Book> | > getName():String  > getCourse():Course  > getRole():UserRole  >borrowBook(book:Book)  >returnBook(book:Book)  > notify  (message : String) :void | Factory |  |
| 3. | UserFactory | >userRegistry : Map<String,User> | getUser  (name:String,role:UserRole  ,department:Department):  User | Singleton for Userfactory and Factory for getUser |  |
| 4. | Course | >name  >code  > bookCategories():List  <BookCategory> | >getName  >getCode  > getBookCategories():  List<BookCategory> | Singleton |  |
| 5. | UserRole | >STUDENT  >LIBRARIAN  >TEACHER |  | Strategy |  |
| 6. | PaymentContext | > paymentStrategy: PaymentStrategy | > setPaymentStrategy(strategy:PaymentStrategy):void  +pay():void | Interface between Payment Strategy and User |  |
| 7. | PaymentStrategy | >CreditCard Payment  >Cash Payment  >Online Payment | >pay() | Strategy |  |
| 8. | Book | > title : String  >author: String  > category : Bookcategory  > availability : boolean | > getTitle() : String  > getAuthor() :String  > getCategory() :BookCategory  > isAvailable() :Boolean | Strategy |  |
| 9. | BookCategory | >Fiction,Non fiction,CSE,ECE,EEE,  MECHANICAL,CIVIL |  | Strategy |  |