STEP

ACADEMY

Mao sokveasna

DESCRIPTION

This project is a compact Library Management System, tracking book details differentiating between Premium and Basic types. It also manages employee data. The system ensures security with straightforward login. Testing and documentation guarantee reliability, with future scalability in mind.

CONTENT

01 BENEFICIAL

02 STEP TO IMPLEMENT

03 LOGICAL DESIGN

04 PHYSICAL DESIGN

BENEFICIAL

01 **Why**
User-friendly
Efficiency
Reliability

02 **Who**
Librarian and library staff
Admin

03 **How**
Time-saving
Promoting library operation
Enhanced accessibility

STEP
TO
IMPLEMENT

Architecture

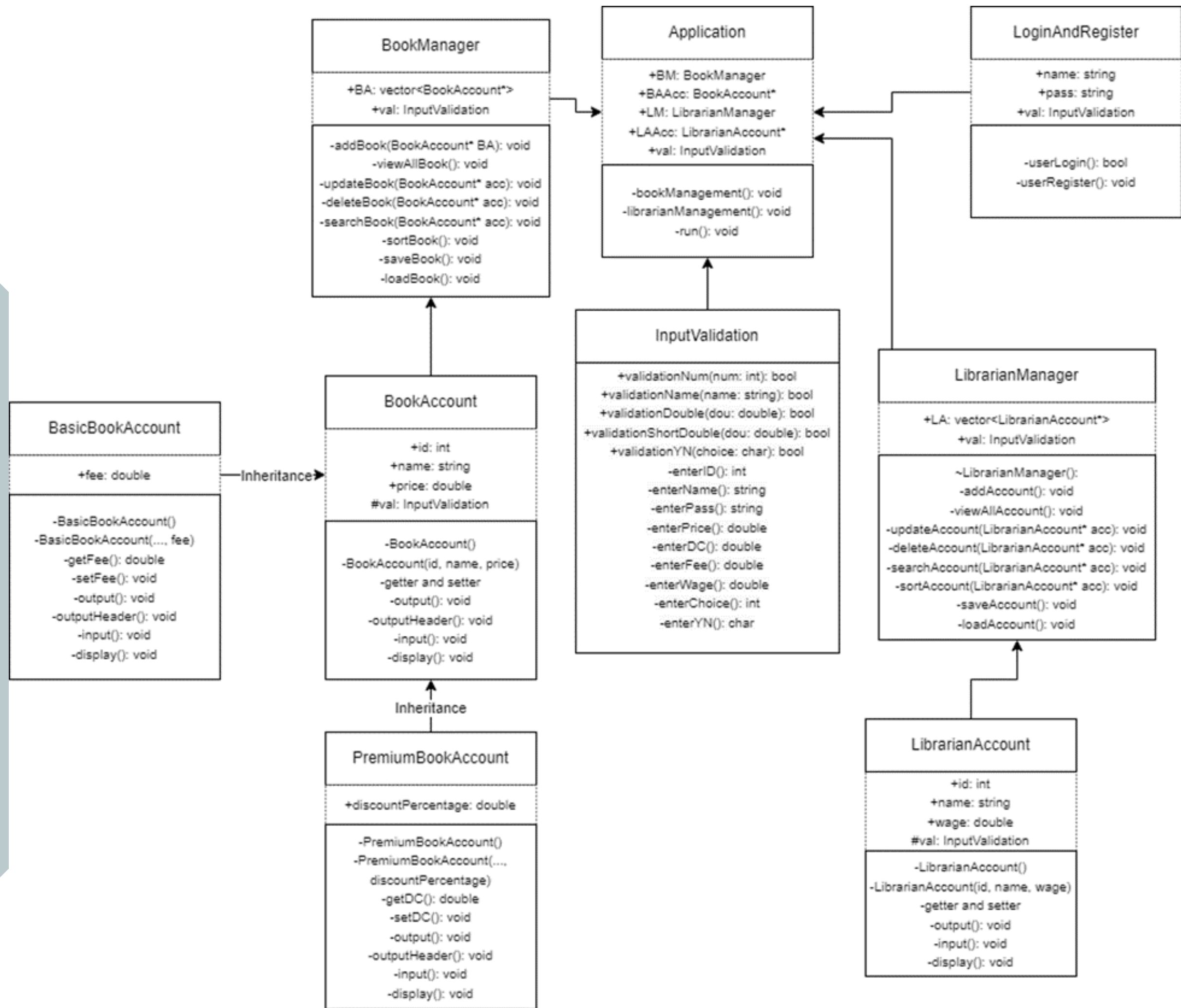STANDALONE

Methodology

AGILE

Tool and Technologies

FILE.DAT

VISUAL STUDIO CODE

C++

24H

LOGICAL DESIGN

BookManager

+BA: vector<BookAccount*>
+val: InputValidation

-addBook(BookAccount* BA): void
-viewAllBook(): void
-updateBook(BookAccount* acc): void
-deleteBook(BookAccount* acc): void
-searchBook(BookAccount* acc): void
-sortBook(): void
-saveBook(): void
-loadBook(): void

Application

+BM: BookManager
+BAAcc: BookAccount*
+LM: LibrarianManager
+LAAcc: LibrarianAccount*
+val: InputValidation

-bookManagement(): void
-librarianManagement(): void
-run(): void

LoginAndRegister

+name: string
+pass: string
+val: InputValidation

-userLogin(): bool
-userRegister(): void

BasicBookAccount

+fee: double

-BasicBookAccount()
-BasicBookAccount(..., fee)
-getFee(): double
-setFee(): void
-output(): void
-outputHeader(): void
-input(): void
-display(): void

—Inheritance→

BookAccount

+id: int
+name: string
+price: double
#val: InputValidation

-BookAccount()
-BookAccount(id, name, price)
-getter and setter
-output(): void
-outputHeader(): void
-input(): void
-display(): void

InputValidation

+validationNum(num: int): bool
+validationName(name: string): bool
+validationDouble(dou: double): bool
+validationShortDouble(dou: double): bool
+validationYN(choice: char): bool
-enterID(): int
-enterName(): string
-enterPass(): string
-enterPrice(): double
-enterDC(): double
-enterFee(): double
-enterWage(): double
-enterChoice(): int
-enterYN(): char

LibrarianManager

+LA: vector<LibrarianAccount*>
+val: InputValidation

~LibrarianManager():
-addAccount(): void
-viewAllAccount(): void
-updateAccount(LibrarianAccount* acc): void
-deleteAccount(LibrarianAccount* acc): void
-searchAccount(LibrarianAccount* acc): void
-sortAccount(LibrarianAccount* acc): void
-saveAccount(): void
-loadAccount(): void

Inheritance

PremiumBookAccount

+discountPercentage: double

-PremiumBookAccount()
-PremiumBookAccount(...,
discountPercentage)
-getDC(): double
-setDC(): void
-output(): void
-outputHeader(): void
-input(): void
-display(): void

LibrarianAccount

+id: int
+name: string
+wage: double
#val: InputValidation

-LibrarianAccount()
-LibrarianAccount(id, name, wage)
-getter and setter
-output(): void
-input(): void
-display(): void

PHYSICAL DESIGN

PHYSICAL DESIGN

```
==================Librarian Manager Menu==================
1. Create new account
2. View all account
3. Update account
4. Delete account
5. Search account
6. Sort account information
7. Save data
8. Back
0. Exit
Enter choice: 1
How many you want to add??
Enter choice: 1
Enter ID(1-999): 1
Enter name: idk
Enter wage($/day): 10
```

```
==================Librarian Manager Menu==================
1. Create new account
2. View all account
3. Update account
4. Delete account
5. Search account
6. Sort account information
7. Save data
8. Back
0. Exit
Enter choice: 2
---------------------------------------LIBRARIAN ACCOUNT----------
ID      Name                            Wage($/day)
-----------------------------------------------------------------
1       idk                             $10
```

THANK YOU

For listening