**Aseel Shaheen**

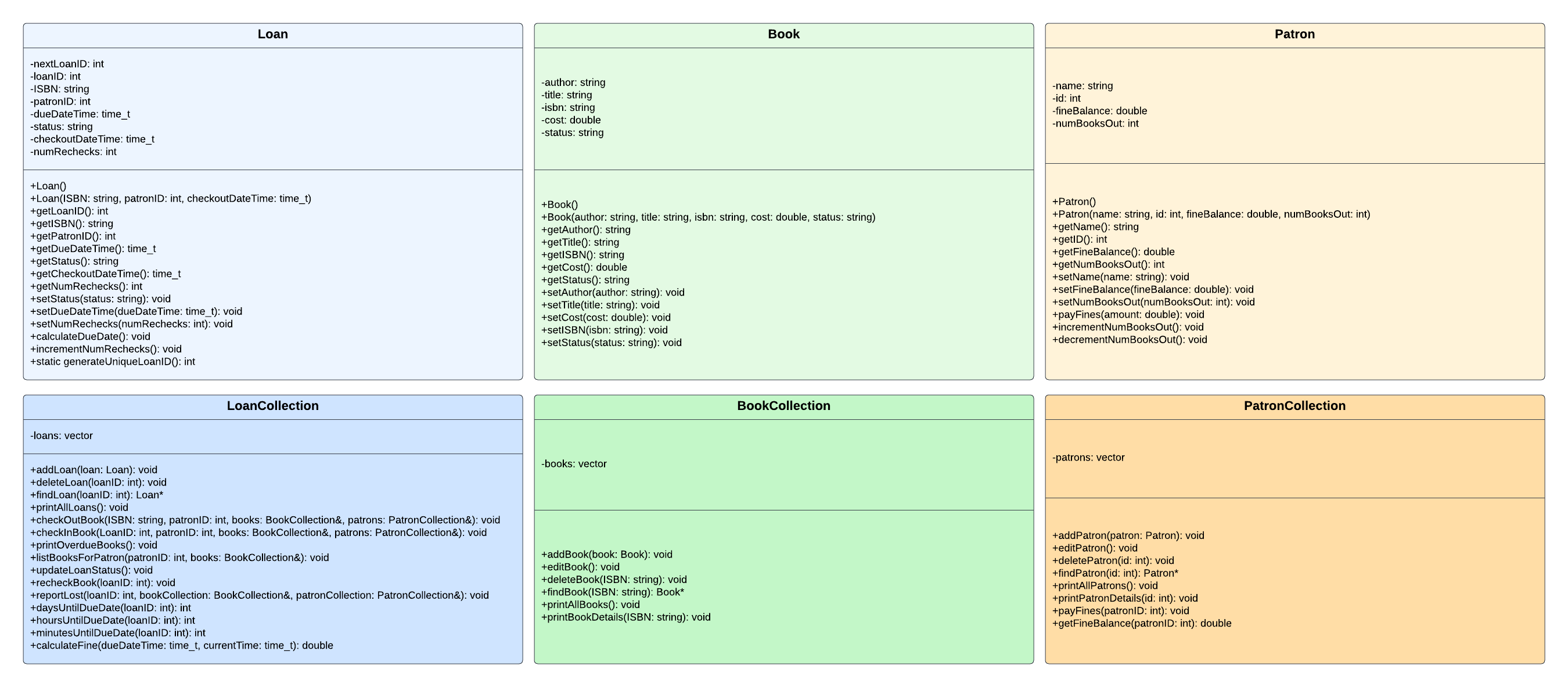
**CSCE 1040.501**

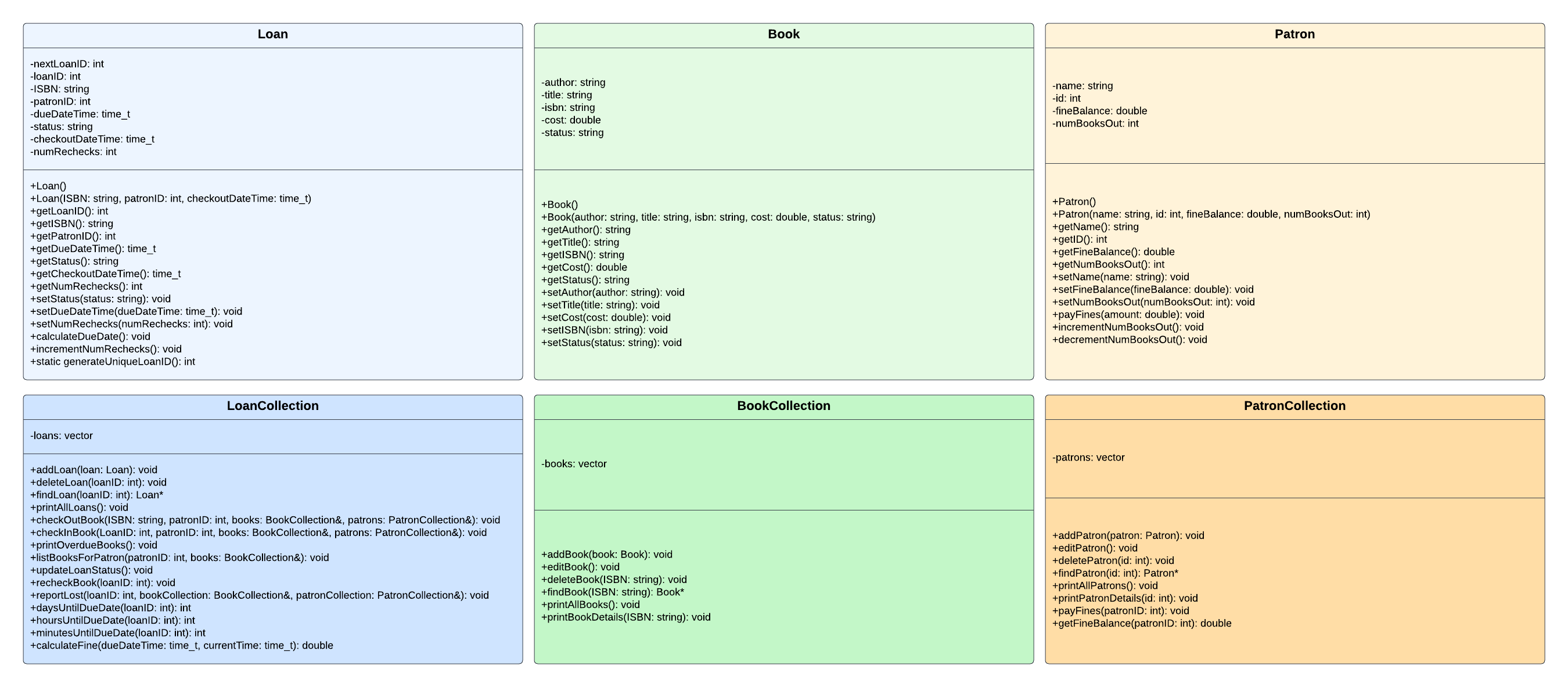
**Homework 2**

**HWK2 Design Document**

A diagram of a loan

Description automatically generated





A screenshot of a computer screen

Description automatically generated

**Psuedocode:**

**Loan Class:**

**- setStatus(status: string): void**

Set the status attribute of the loan object to the new status.

**- getStatus(): string**

Return the status attribute of the loan object.

**- printLoanDetails(): void**

**Print the details of the loan.**

Output the loan ID, book ID, patron ID, due date and time, and current status of the loan.

**LoanCollection Class:**

**- addLoan(loan: Loan): void**

Add the provided loan to the collection.

**- deleteLoan(loanID: int): void**

Delete the loan with the given ID from the collection.

**- findLoan(loanID: int): Loan\***

Find and return the loan with the given ID from the collection.

**- printAllLoans(): void**

Print the details of all loans in the collection.

**- checkOutBook(ISBN: string, patronID: int, books: BookCollection&, patrons: PatronCollection&): void**

Check out a book with the given ISBN to the patron with the given ID.

**- checkInBook(LoanID: int, patronID: int, books: BookCollection&, patrons: PatronCollection&): void**

Check in the book associated with the given loan ID.

**- printOverdueBooks(): void**

Print the details of all overdue books.

**- listBooksForPatron(patronID: int, books: BookCollection&): void**

List all books currently checked out by the patron with the given ID.

**- updateLoanStatus(): void**

Update the status of all loans based on current date and time.

**- recheckBook(loanID: int): void**

Recheck the book associated with the given loan ID.

**- reportLost(loanID: int, bookCollection: BookCollection&, patronCollection: PatronCollection&): void**

Report the book associated with the given loan ID as lost.

**- daysUntilDueDate(loanID: int): int**

Calculate the number of days until the due date for the loan with the given ID.

**- hoursUntilDueDate(loanID: int): int**

Calculate the number of hours until the due date for the loan with the given ID.

**- minutesUntilDueDate(loanID: int): int**

Calculate the number of minutes until the due date for the loan with the given ID.

**- calculateFine(dueDateTime: time\_t, currentTime: time\_t): double**

Calculate the fine for a loan based on the due date and the current time.

**Book Class:**

**- setName(name: string): void**

Set the name attribute of the book object to the provided name.

**- getName(): string**

Return the name attribute of the book object.

**- setIDNumber(idNumber: int): void**

Set the ID number attribute of the book object to the provided ID number.

**- getIDNumber(): int**

Return the ID number attribute of the book object.

**- setFineBalance(fineBalance: double): void**

Set the fine balance attribute of the book object to the provided fine balance.

**- getFineBalance(): double**

Return the fine balance attribute of the book object.

**- setCurrentBooksOut(currentBooksOut: int): void**

Set the current books out attribute of the book object to the provided value.

**- getCurrentBooksOut(): int**

Return the current books out attribute of the book object.

**- setAuthor(author: string): void**

Set the author attribute of the book object to the provided author.

**- getAuthor(): string**

Return the author attribute of the book object.

**- setTitle(title: string): void**

Set the title attribute of the book object to the provided title.

**- getTitle(): string**

Return the title attribute of the book object.

**- setISBN(ISBN: string): void**

Set the ISBN attribute of the book object to the provided ISBN.

**- getISBN(): string**

Return the ISBN attribute of the book object.

**- setLibraryID(libraryID: string): void**

Set the library ID attribute of the book object to the provided library ID.

**- getLibraryID(): string**

Return the library ID attribute of the book object.

**- setCost(cost: double): void**

Set the cost attribute of the book object to the provided cost.

**- getCost(): double**

Return the cost attribute of the book object.

**- setStatus(status: string): void**

Set the status attribute of the book object to the new status.

**- getStatus(): string**

Return the status attribute of the book object.

**- printBookDetails(): void**

Print the details of the book.

Output the author, title, ISBN, library ID number, cost, and current status of the book.

**BookCollection Class:**

**- addBook(book: Book): void**

Add the provided book to the collection.

**- editBook(): void**

Edit the details of a book in the collection.

**- deleteBook(ISBN: string): void**

Delete the book with the given ISBN from the collection.

**- findBook(ISBN: string): Book\***

Find and return the book with the given ISBN from the collection.

**- printAllBooks(): void**

Print the details of all books in the collection.

**Patron Class:**

**- setName(name: string): void**

Set the name attribute of the patron object to the provided name.

**- getName(): string**

Return the name attribute of the patron object.

**- setID(id: int): void**

Set the ID attribute of the patron object to the provided ID.

**- getID(): int**

Return the ID attribute of the patron object.

**- setFineBalance(fineBalance: double): void**

Set the fine balance attribute of the patron object to the provided fine balance.

**- getFineBalance(): double**

Return the fine balance attribute of the patron object.

**- setCurrentBooksOut(currentBooksOut: int): void**

Set the current books out attribute of the patron object to the provided value.

**- getCurrentBooksOut(): int**

Return the current books out attribute of the patron object.

**- payFines(amount: double): void**

Add the provided amount to the fine balance attribute of the patron object.

**- incrementNumBooksOut(): void**

Increment the count of current books out by 1 for the patron object.

**- decrementNumBooksOut(): void**

Decrement the count of current books out by 1 for the patron object.

**PatronCollection Class:**

**- addPatron(patron: Patron): void**

Add the provided patron to the collection.

**- editPatron(): void**

Edit the details of a patron in the collection.

**- deletePatron(id: int): void**

Delete the patron with the given ID from the collection.

**- findPatron(id: int): Patron\***

Find and return the patron with the given ID from the collection.

**- printAllPatrons(): void**

Print the details of all patrons in the collection.

**Design experience:**

I initially had to plan out each part on paper, breaking down the tasks into smaller steps. This helped me organize everything neatly and ensure that my pseudocode methods work with one another. I must admit it required more time and effort than I had assumed, considering I had to grasp the concepts of UML diagrams. I decided to use smart Draw for the UML diagrams as it gave neater diagrams than for example word would’ve.