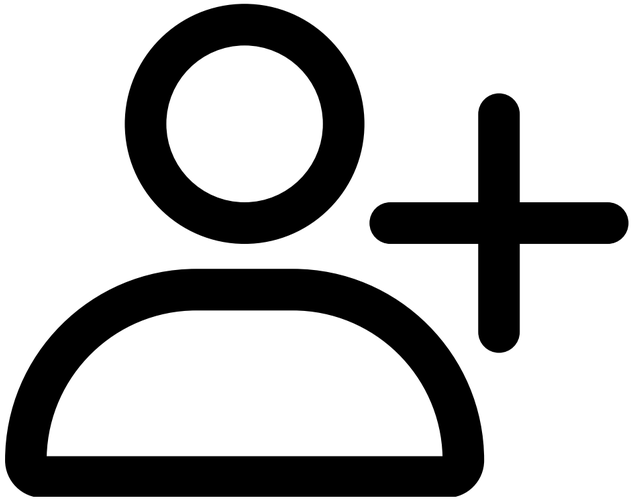
OOP-TEST

Libarry Management System

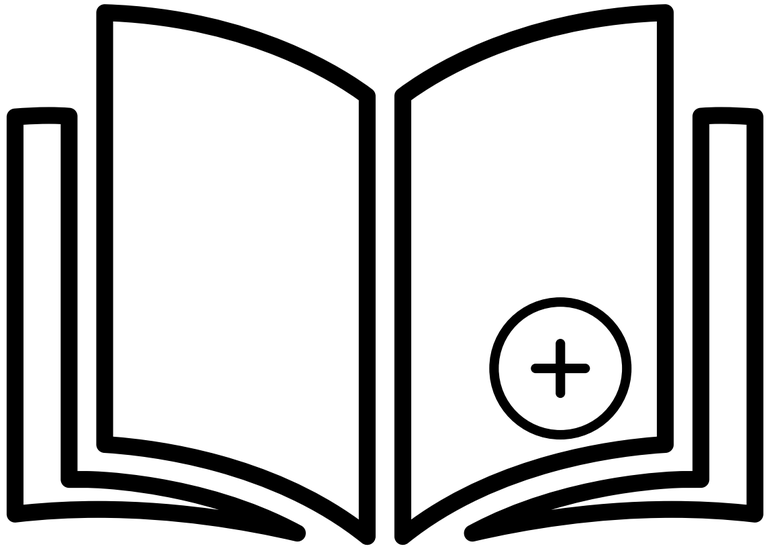
Feture ของระบบ

-(การยืม-คืนหนังสือ)

-(การเพิ่มสมาชิก)

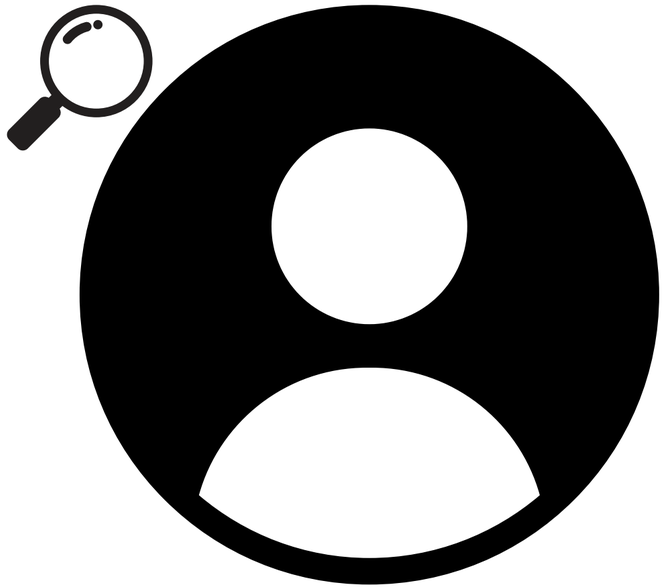


-(การเพิ่มหนังสือ)

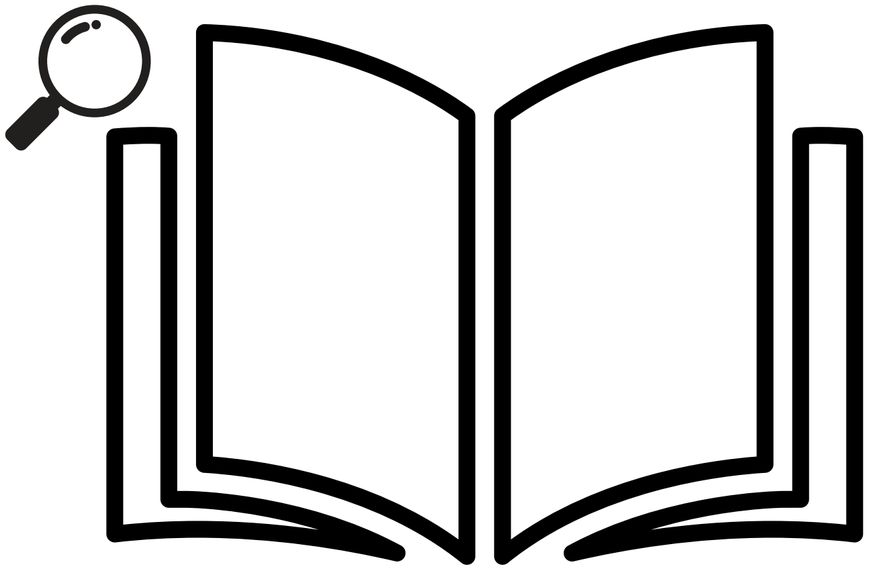


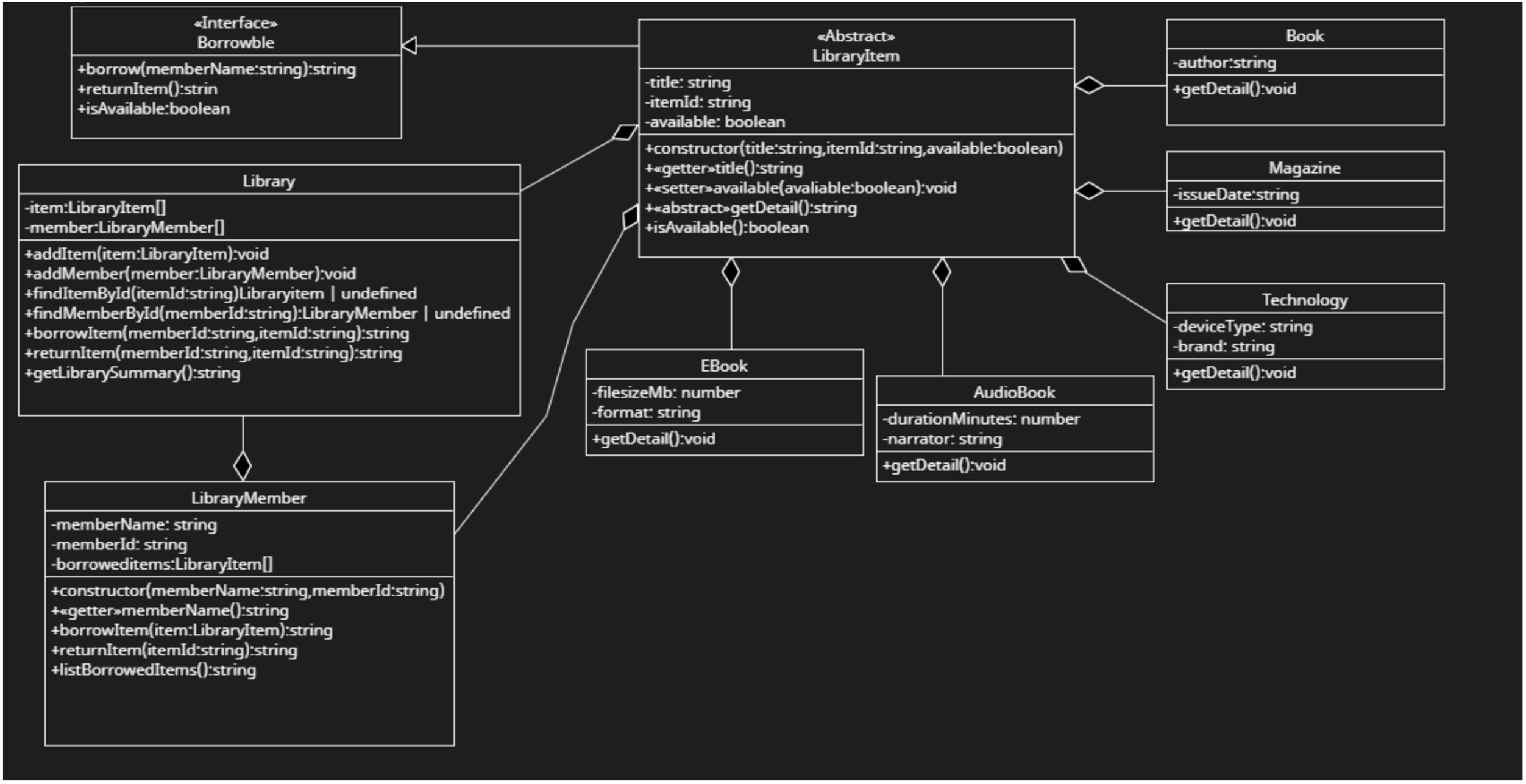
Feture ของระบบ

-(การค้นหาสมาชิก)



-(การค้นหาหนังสือ)




«Interface»
Borrowble
+borrow(memberName:string):string
+returnItem():strin
+isAvailable:boolean
«Abstract»
LibraryItem
-title: string
-itemId: string
-available: boolean
+constructor(title:string,itemId:string,available:boolean)
+«getter»title():string
+«setter»available(avaliable:boolean):void
+«abstract»getDetail():string
+isAvailable():boolean
Book
-author:string
+getDetail():void
Magazine
-issueDate:string
+getDetail():void
Technology
-deviceType: string
-brand: string
+getDetail():void
Library
-item:LibraryItem[]
-member:LibraryMember[]
+addItem(item:LibraryItem):void
+addMember(member:LibraryMember):void
+findItemById(itemId:string)Libraryitem | undefined
+findMemberById(memberId:string):LibraryMember | undefined
+borrowItem(memberId:string,itemId:string):string
+returnItem(memberId:string,itemId:string):string
+getLibrarySummary():string
EBook
-filesizeMb: number
-format: string
+getDetail():void
AudioBook
-durationMinutes: number
-narrator: string
+getDetail():void
LibraryMember
-memberName: string
-memberId: string
-borroweditems:LibraryItem[]
+constructor(memberName:string,memberId:string)
+«getter»memberName():string
+borrowItem(item:LibraryItem):string
+returnItem(itemId:string):string
+listBorrowedItems():string

interface เพราะตั้งสัญญาไว้แล้ว อะไรที่จะเอาไปใช้ได้ต้องมี borrow,ReturnItem,isAvailable

```typescript
interface Borrowable {
  borrow(memberName: string): string;
  returnItem(): string;
  isAvailable: boolean;
}
```

inheritance คลาสพวกนี้สืบทอดมาจากคลาสหลักคือ LibraryObject มี class book , class Magazine , class ebooks , class Soundbook , class technologyequipment

```typescript
class Book extends LibraryObject {
  private author: string;

  constructor(title: string, itemId: string, available: boolean, author: string) {
    super(title, itemId, available);
    this.author = author;
  }

  getDetail(): string {
    return `Book: ${this.title}, Author: ${this.author}`;
  }
}
```

```typescript
class Magazine extends LibraryObject {
  private issueDate: string;

  constructor(title: string, itemId: string, available: boolean, issueDate: string) {
    super(title, itemId, available);
    this.issueDate = issueDate;
  }

  getDetail(): string {
    return `Magazine: ${this.title}, Issue Date: ${this.issueDate}`;
  }
}
```

```
class EBooks extends LibraryObject {
  private filesizeMb: number;
  private format: string;

  constructor(title: string, itemId: string, available: boolean, filesizeMb: number, format: string) {
    super(title, itemId, available);
    this.filesizeMb = filesizeMb;
    this.format = format;
  }

  getDetail(): string {
    return `EBook: ${this.title}, Size: ${this.filesizeMb}MB, Format: ${this.format}`;
  }
}
```

```
class SoundBook extends LibraryObject {
  private durationMinutes: number;
  private narrator: string;

  constructor(title: string, itemId: string, available: boolean, durationMinutes: number, narrator: string) {
    super(title, itemId, available);
    this.durationMinutes = durationMinutes;
    this.narrator = narrator;
  }

  getDetail(): string {
    return `AudioBook: ${this.title}, Duration: ${this.durationMinutes} mins, Narrator: ${this.narrator}`;
  }
}
```

```
class TechnologyEquipment extends LibraryObject {
  private deviceType: string;
  private brand: string;

  constructor(title: string, itemId: string, available: boolean, deviceType: string, brand: string) {
    super(title, itemId, available);
    this.deviceType = deviceType;
    this.brand = brand;
  }

  getDetail(): string {
    return `Technology: ${this.title}, Type: ${this.deviceType}, Brand: ${this.brand}`;
  }
}
```

```
class Library {
  private items: LibraryObject[];
  private members: LibraryMembers[];
```

Aggregation เพราะ มันมี LibraryObject และ LibraryMembers

หรือก็คือ Library รู้จัก LibraryObject และ LibraryMembers

Encapsulation คือการใช้ private

```
class Book extends LibraryObject {
  private author: string;
```

```
class Library {
  private items: LibraryObject[];
  private members: LibraryMembers[];

  constructor() {
    this.items = [];
    this.members = [];
  }

  addItem(item: LibraryObject): void {
    this.items.push(item);
  }

  addMember(member: LibraryMembers): void {
    this.members.push(member);
  }

  findItemById(itemId: string): LibraryObject | undefined {
    return this.items.find((item) => item["itemId"] === itemId);
  }

  findMemberById(memberId: string): LibraryMembers | undefined {
    return this.members.find((member) => member["memberId"] === memberId);
 borrowItem(memberId: string, itemId: string): string {
   const member = this.findMemberById(memberId);
   const item = this.findItemById(itemId);
   if (member && item) {
     return member.borrowItem(item);
   }
   return "Member or Item not found.";
 }

 returnItem(memberId: string, itemId: string): string {
   const member = this.findMemberById(memberId);
   if (member) {
     return member.returnItem(itemId);
   }
   return "Member not found.";
 }

 getLibrarySummary(): string {
   return this.items.map((item) => item.getDetail()).join("\n");
 }
}
```

5. สร้าง class ชื่อ **Library** (Aggregation โดยเก็บ **LibraryItem** และ **LibraryMember**):

- **Attributes**:
  - private items: LibraryItem[] - รายการสิ่งของในห้องสมุด
  - private members: LibraryMember[] - รายการสมาชิก
- **Methods**:
  - addItem(item: LibraryItem): void - เพิ่มสิ่งของลง items
  - addMember(member: Library Member): void - เพิ่มสมาชิกลง members
  - findItemById(itemId: string): LibraryItem | undefined - หาสิ่งของจาก ID
  - findMemberById(memberId: string): LibraryMember | undefined - หาสมาชิกจาก ID
  - borrowItem(memberId: string, itemId: string): string - หา member และ item แล้วเรียก borrowItem ของ member (คืนข้อความผลลัพธ์)
  - returnItem(memberId: string, itemId: string): string - หา member แล้วเรียก returnItem ของ member
  - getLibrarySummary(): string - คืนค่าสรุปรายชื่อสิ่งของและสมาชิก (ใช้ getDetails() หรือ getMemberName())

จากภาพ โค้ดจะมีการ สร้าง Attributes และ Method ตาม Requirement ครบถ้วนสมบูรณ์

logic

LibraryObject ทุกสิ่งของมีสถานะ true/false การยืม การคืน LibraryMembers เก็บรายการจากคนที่ยืม Library เก็บรายการห้องสมุดกับสมาชิก

สามารถขยายระบบในอนาคต

ขยายชนิดของสิ่งของใน Library ขยายฟีเจอร์ของสมาชิก ขยาย Library เพิ่มฟีเจอร์ใหม่ที่ไม่ซ้ำอันเดิม