TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công nghệ thông tin và Truyền thông

Tài liệu mô tả thiết kế phần mềm

(Software Design Description)

Phiên bản 1

**Library Management System**

Môn: Thiết kế xây dựng phần mềm

Nhóm 07

Cai Việt Hoàng

Nguyễn Thanh Hoà

Nguyễn Minh Hoàng

Đinh Nho Hùng

*Hà Nội, 08/05/2018*

Mục lục

Mục lục 1

1. Giới thiệu 3

1.1. Mục đích 3

1.2. Phạm vi 3

1.3. Từ điển thuật ngữ 3

1.4. Tham khảo 3

2. Thiết kế kiến trúc 3

2.1. Lựa chọn kiến trúc phần mềm 3

2.2. Thiết kế tổng quan 6

2.3. Thiết kế chi tiết gói 6

2.4. Biểu đồ tương Tbc 8

3. Thiết kế giao diện 10

3.1. Giao diện với thiết bị phần cứng 10

3.2. Giao diện với phần mềm khác 10

3.3. Giao diện người dùng 10

***3.3.1.*** Biểu đồ dịch chuyển màn hình 10

***3.3.2.*** Thiết kế giao diện 11

4. Thiết kế lớp 17

4.1. Biểu đồ lớp thiết kế 17

4.2. Thiết kế lớp chi tiết 17

***4.2.1.*** Thiết kế lớp AlertScene 17

***4.2.2.*** Thiết kế lớp DatabaseHandler 18

***4.2.3.*** Thiết kế lớp Preferences và Setting 18

***4.2.4.*** Thiết kế lớp BookAddController 19

***4.2.5.*** Thiết kế lớp MemberAddController 20

***4.2.6.*** Thiết kế lớp BookListController 21

***4.2.7.*** Thiết kế lớp MemberListController 21

***4.2.8.*** Thiết kế lớp IssuedListController 22

***4.2.9.*** Thiết kế lớp Login Controller 23

***4.2.10.*** Thiết kế lớp Util 23

5. Thiết kế mô hình dữ liệu 24

5.1. Biểu đồ thực thể liên kết 24

5.2. Chuẩn hoá 24

***5.2.1.*** Các bước chuẩn hoá 24

***5.2.2.*** Biểu đồ liên kết giữa các bảng 24

5.3. Thiết kế chi tiết bảng 24

***5.3.1.*** Bảng BOOK 24

***5.3.2.*** Bảng ISSUE 25

***5.3.3.*** Bảng MEMBER 25

# Giới thiệu

## Mục đích

Thiết kế một hệ thống quản lí thư viện (Library Management System – LMS) hoàn chỉnh.

## Phạm vi

Phần mềm thiết kế theo ngôn ngữ mô hình hoá cấu trúc UML. Áp dụng mô hình hoá vào việc lập trình, cụ thể ở đây là ngôn ngữ Java.

## Từ điển thuật ngữ

<Đưa ra các thuật ngữ được sử dụng trong tài liệu này. Không được tự ý mặc định về kinh nghiệm hoặc kiến thức của người đọc>

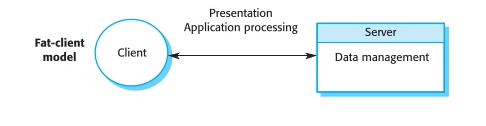
## Tham khảo

* Lập trình Java với JDBC
* Giao diện JavaFX

# Thiết kế kiến trúc

## Lựa chọn kiến trúc phần mềm

Phần mềm có sử dụng kiến trúc hai tầng cụ thể là loại "Tính toán tập trung ở máy khách" fat client – thin server

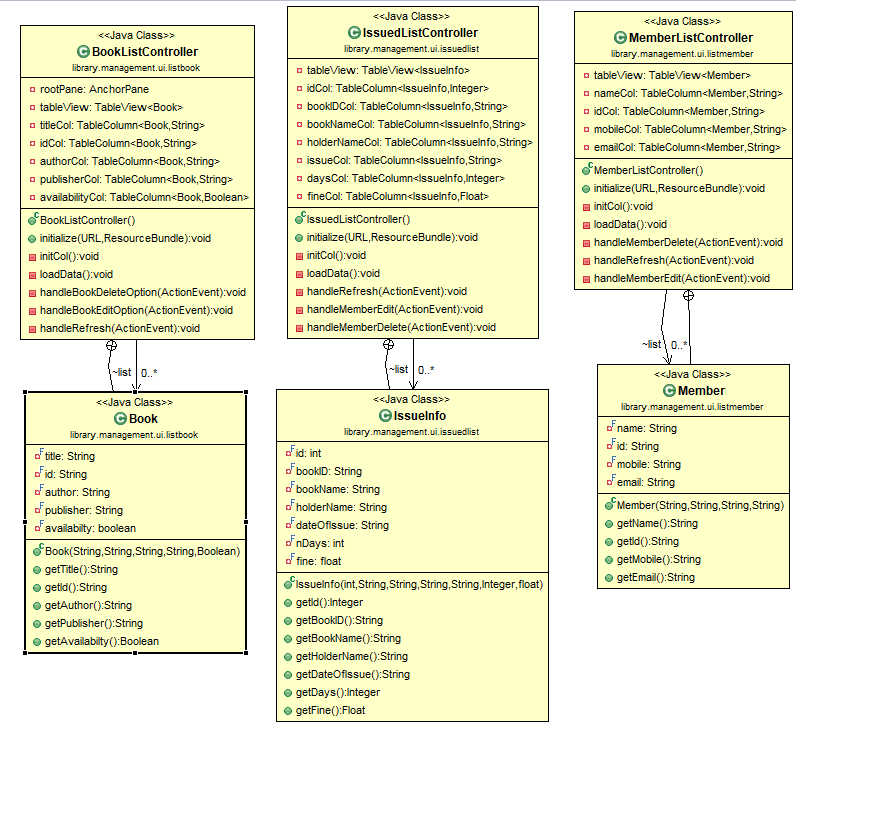


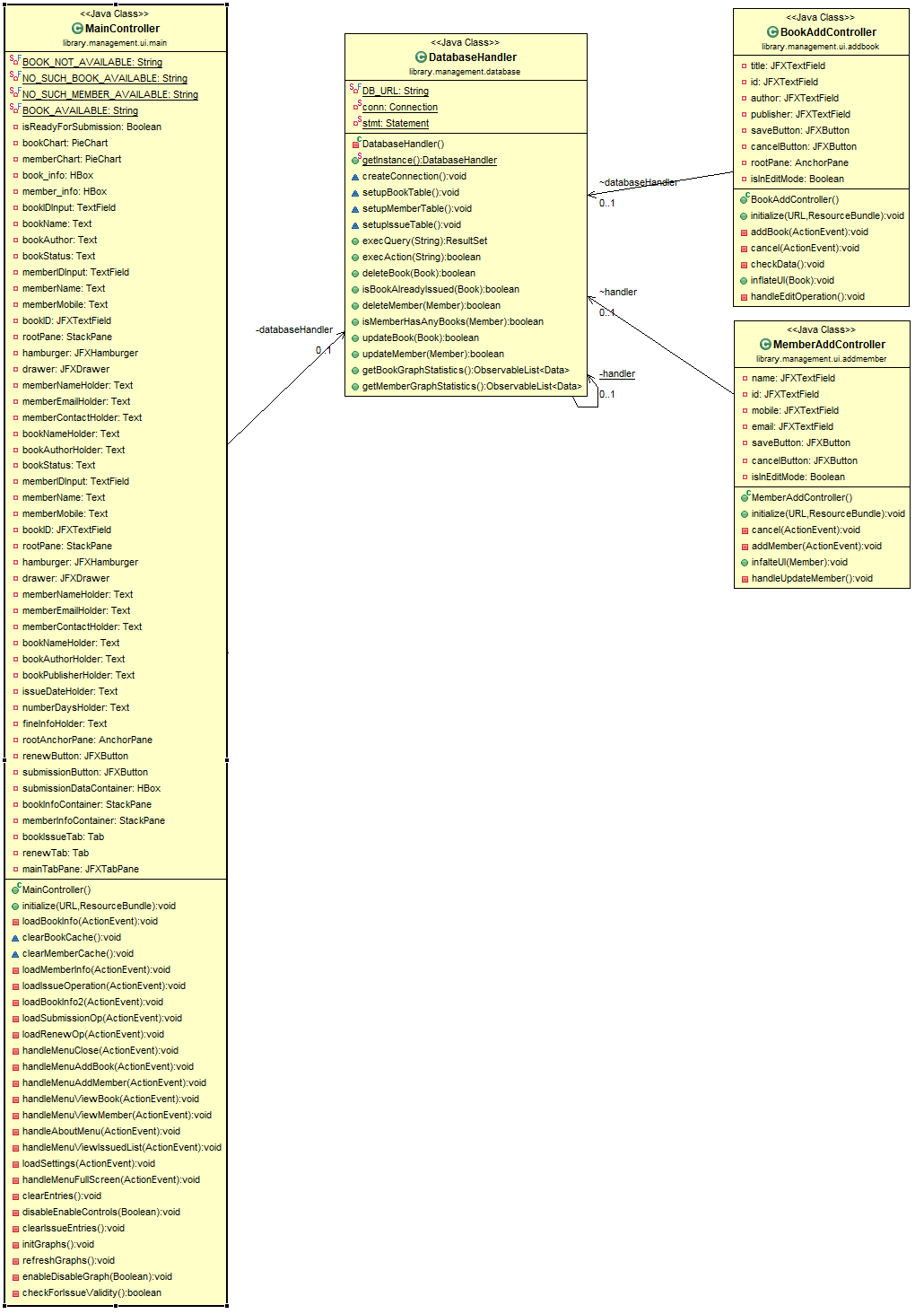
Tổng quan về kiến trúc 2 tầng loại "Tính toán tập trung ở máy khách" fat client – thin server: Ở loại này phải đảm nhận cả 2 tầng là tầng giao diện và tầng chức năng, còn máy chỉ làm nhiệm vụ của tầng cơ sở dữ liệu (CSDL).

Ưu điểm của mô hình loại này là đơn giản thích hợp cho các ứng dụng nhỏ và vừa. Tuy nhiên nó cũng có một số nhược điểm:

* Do phần tính toán nghiêng về phía máy khách nhiều nên đòi hỏi máy khách có cấu hình đủ mạnh.
* Cũng do phần chức năng nằm bên phía máy khách nên khi chương trình cần nâng cấp thì sẽ rất khó khăn do đòi hỏi phải cập nhật lại toàn bộ chương trình nằm ở toàn bộ tất cả các máy khách.
* Do mọi thao tác trên cơ sở dữ liệu đều thông qua mạng giữa máy khách và máy chủ nên tốc độ chương trình sẽ chậm đi.

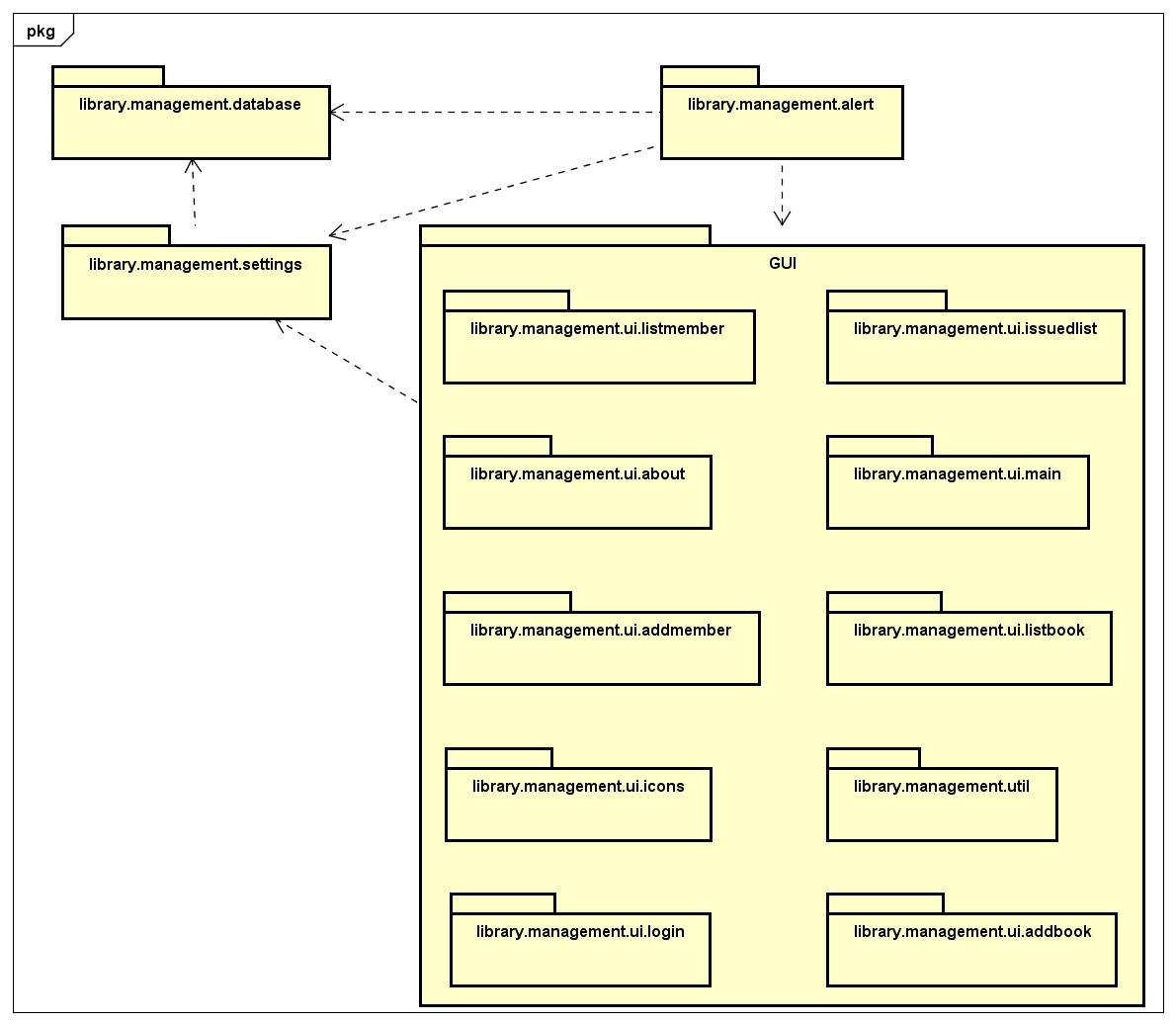
Một ví dụ trong phần mềm:





Ở đây chúng tôi đã xây dựng lớp giao diện và lớp điều khiển của mỗi chức năng chung vào một lớp. Như ở lớp BookAddController đã xây dựng giao diện và các hàm trong lớp BookAddController có chức năng xử lý nghiệp vụ, các hàm ấy không nằm trong một lớp controller riêng lẻ như trong kiến trúc 3 tầng hay tích hợp cùng lớp database như loại "Tính toán tập trung ở máy chủ". Ở trên, những class có mang từ cuối là "Controller" đều là tích hợp của lớp UI và Controller trong kiến trúc 3 tầng mặc dù tưởng như những class đó chỉ đảm nhận vai trò xử lý tính toán.

## Thiết kế tổng quan



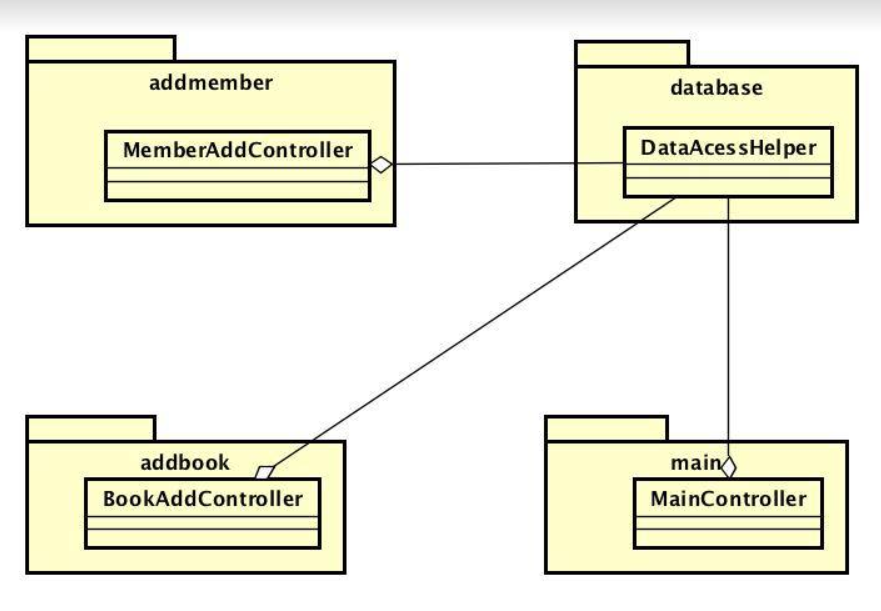
Hình 1 Ví dụ biểu đồ phụ thuộc gói.

## Thiết kế chi tiết gói

Sinh viên thiết kế và lần lượt vẽ biểu đồ thiết kế cho từng package, hoặc một nhóm các package liên quan để giải quyết một vấn đề gì đó. Khi vẽ thiết kế gói, sinh viên chỉ cần đưa tên lớp, không cần chỉ ra các thành viên phương thức và thuộc tính. SV tham khảo ví dụ minh họa trong Hình 2.

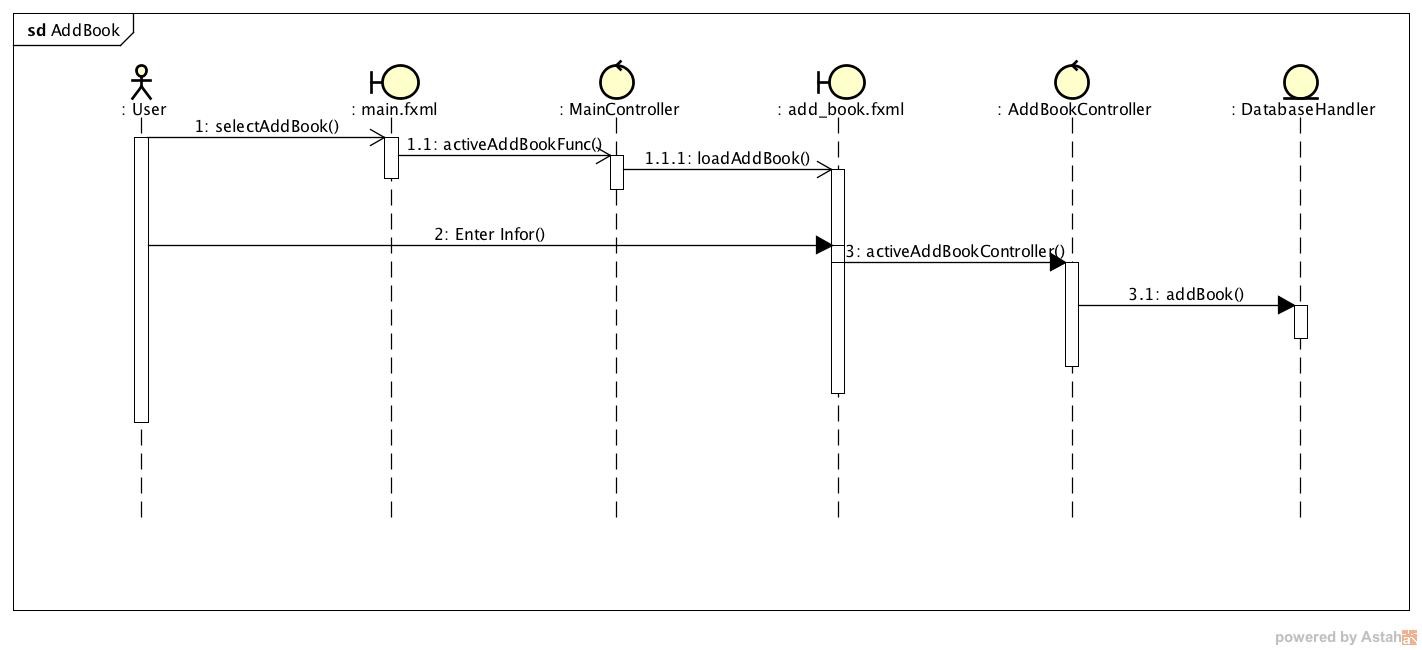
Sinh viên cần vẽ rõ ràng quan hệ giữa các lớp trong biểu đồ. Các quan hệ bao gồm: phụ thuộc (dependency), kết hợp (association), kết tập (aggregation), hợp thành (composition), kế thừa (inheritance), và thực thi (implementation). Các quan hệ này đều đã được minh họa trong Hình 2.

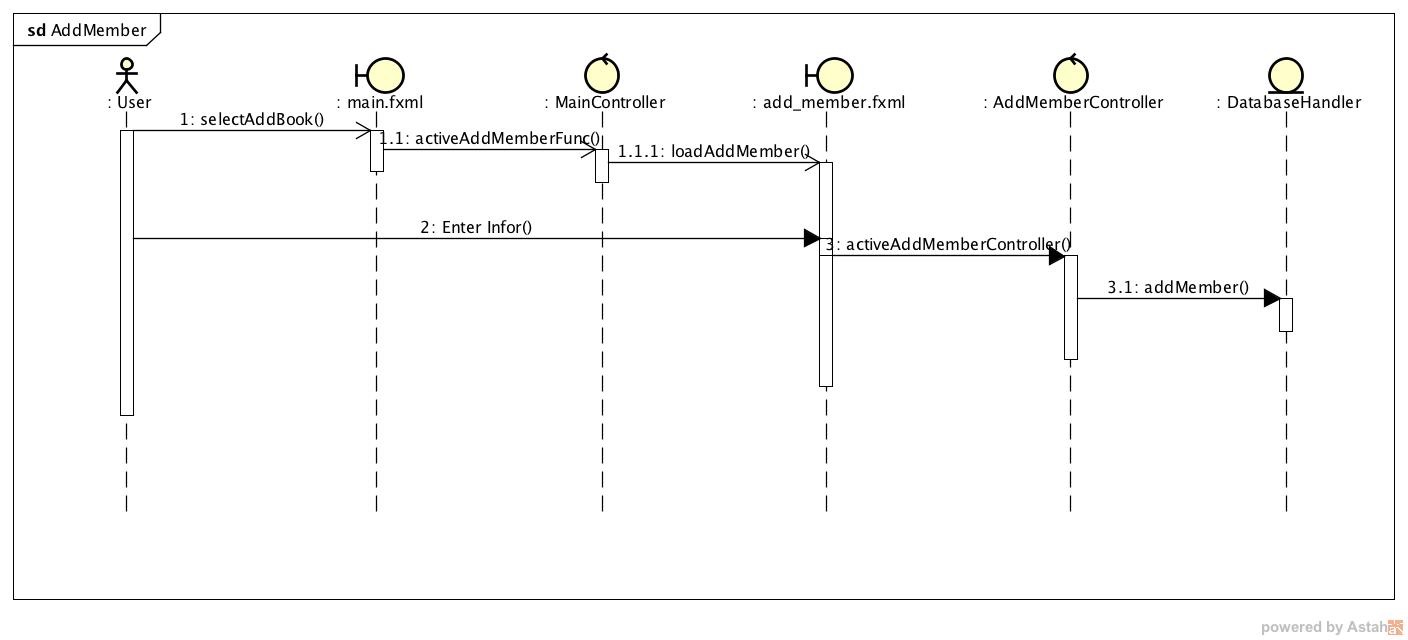
Sau khi vẽ hình minh họa, sinh viên cần giải thích ngắn gọn về thiết kế của mình.

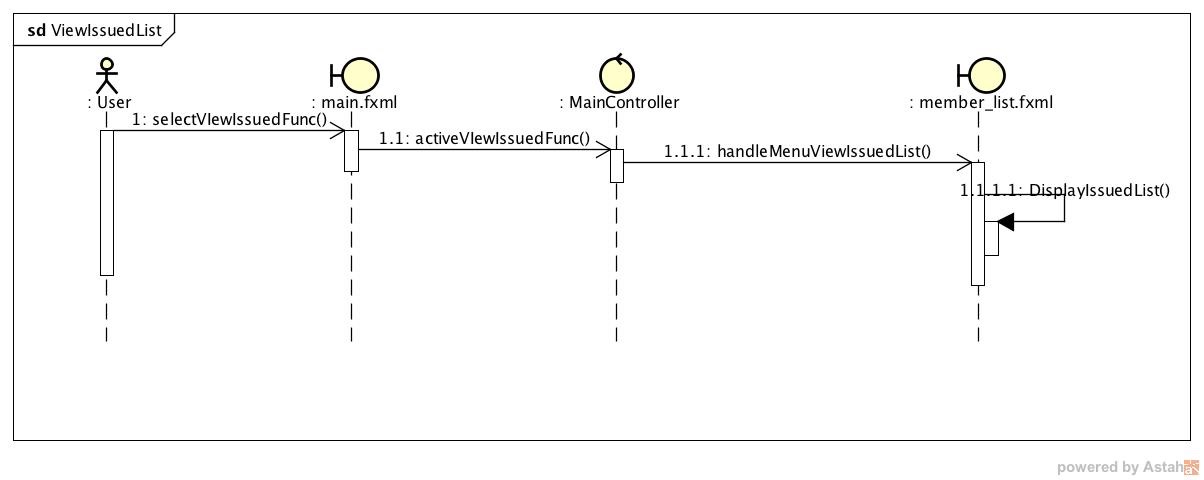


Hình 2 Ví dụ thiết kế gói.

## Biểu đồ tương Tbc







# Thiết kế giao diện

## Giao diện với thiết bị phần cứng

<Nếu phần mềm có tương tác với một thiết bị phần cứng nào đó, thì cần chỉ rõ cách thức tương tác, định dạng dữ liệu / thông điệp, giao thức truyền thông giữa thiết bị phần cứng và phần mềm. Phần này cần thiết kế định dạng dữ liệu (VD: File text có định dạng cấu trúc như thế nào, nếu là XML thì cần định nghĩa cấu trúc tài liệu XML…) hoặc định dạng thông điệp, mô tả rõ về cách thức tương tác / giao thức>

## Giao diện với phần mềm khác

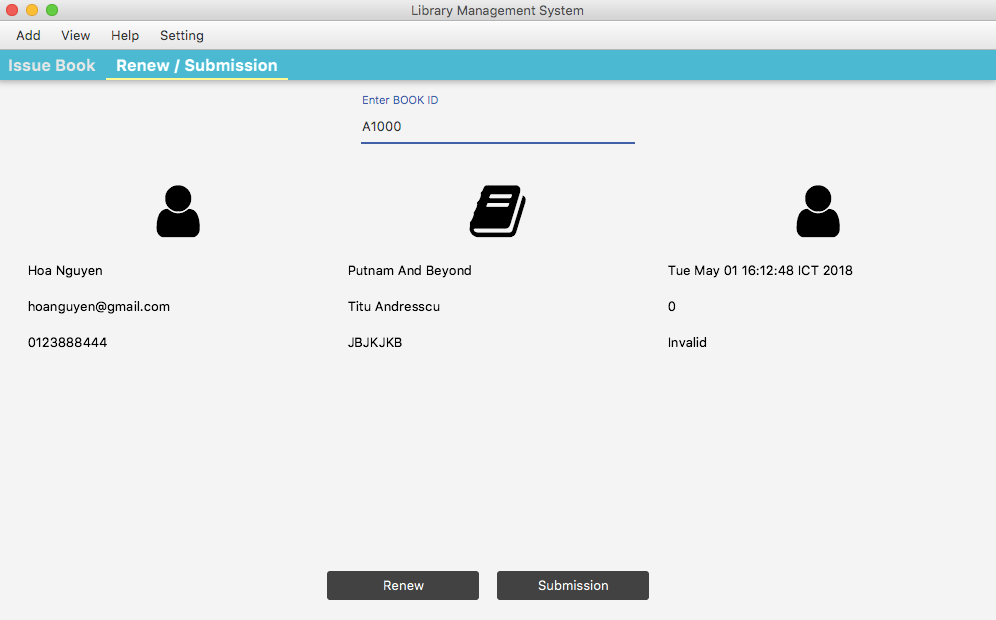
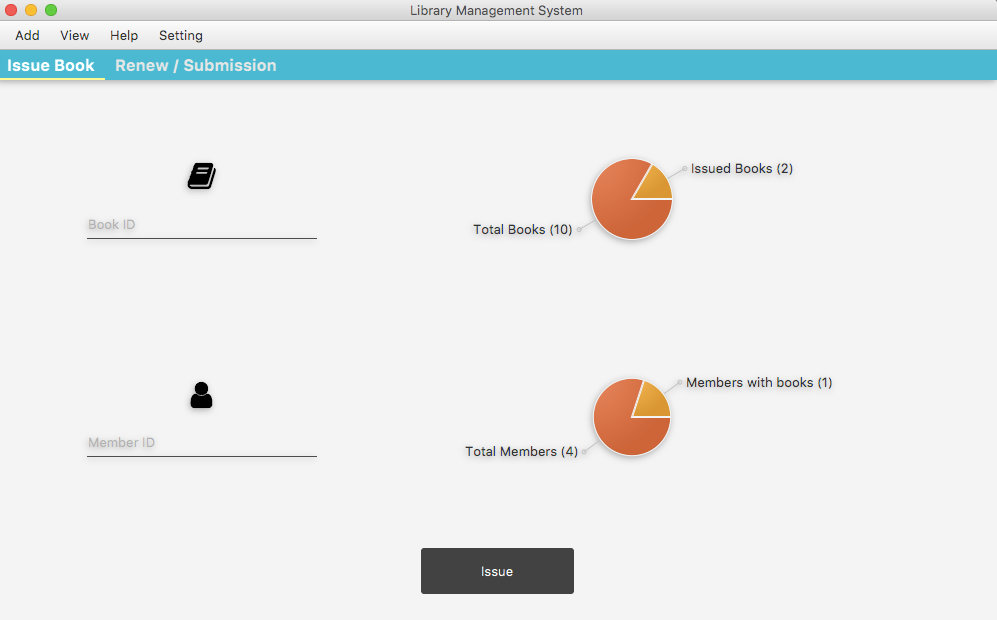
<Nếu phần mềm có tương tác với một phần mềm nào đó, thì cần chỉ rõ cách thức tương tác, định dạng dữ liệu / thông điệp, giao thức truyền thông giữa hai phần mềm. Phần này cần thiết kế định dạng dữ liệu (VD: File text có định dạng cấu trúc như thế nào, nếu là XML thì cần định nghĩa cấu trúc tài liệu XML…) hoặc định dạng thông điệp, mô tả rõ về cách thức tương tác / giao thức>

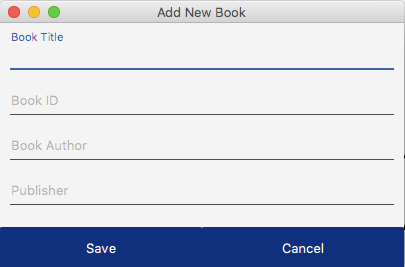
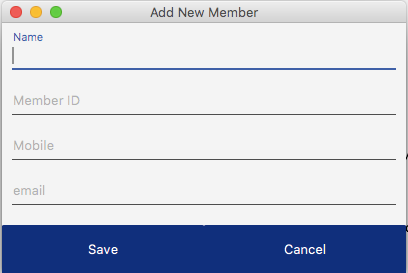
## Giao diện người dùng

### Biểu đồ dịch chuyển màn hình

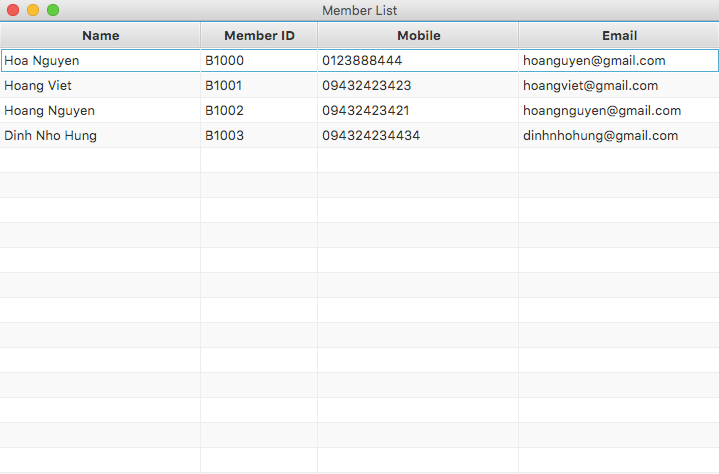
<Vẽ biểu đồ dịch chuyển giữa các màn hình, tương ứng với các lớp biên trong biểu đồ lớp tương tác gộp. Trong phần này, chỉ đưa ra các lớp biên tương tác với tác nhân là con người>

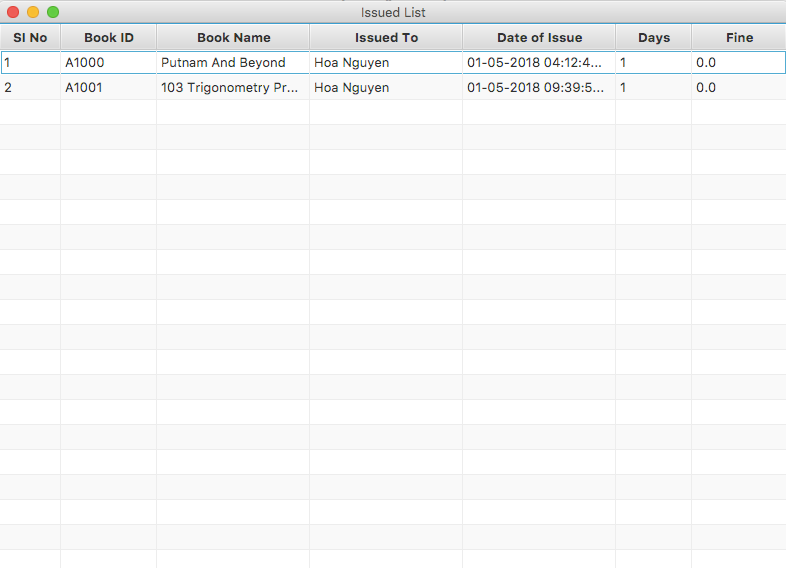
### Thiết kế giao diện

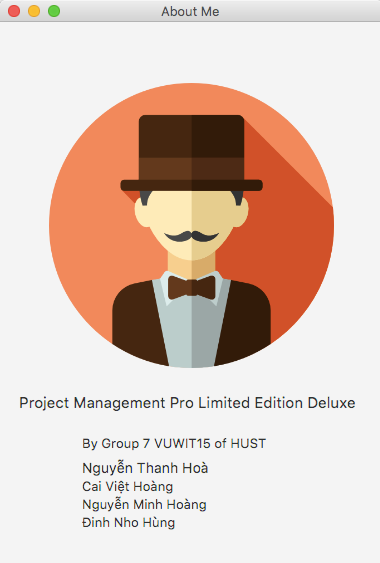












# Thiết kế lớp

## Biểu đồ lớp thiết kế

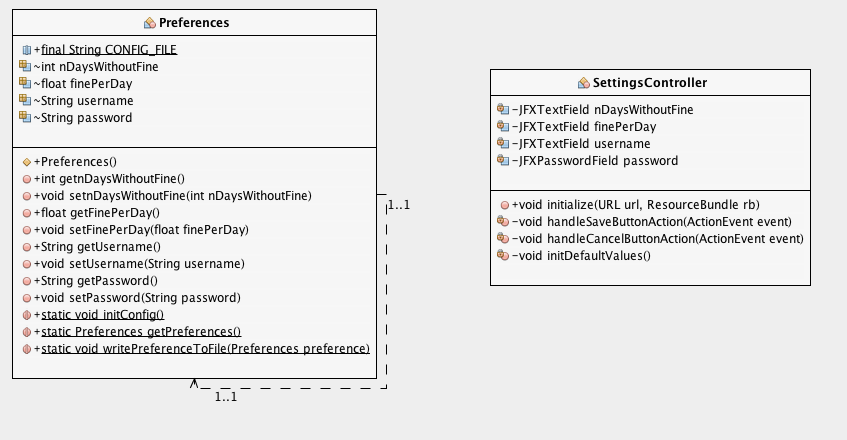
<Ánh xạ từ biểu đồ lớp phân tích sang biểu đồ lớp thiết kế. Có thể chỉnh sửa, thêm hoặc xoá một số lớp. Trong biểu đồ này, cần liệt kê các thuộc tính và kiểu dữ liệu của nó, chữ ký các thao tác theo ngôn ngữ lập trình đã chọn>

## Thiết kế lớp chi tiết

### Thiết kế lớp AlertScene

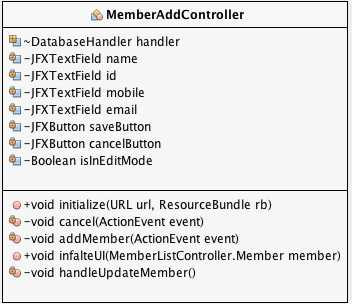
### Thiết kế lớp DatabaseHandler

### Thiết kế lớp Preferences và Setting

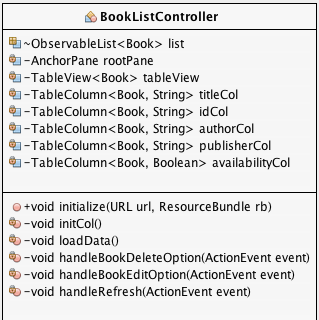


### Thiết kế lớp BookAddController

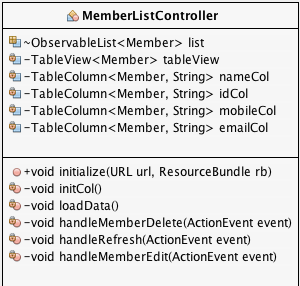
### Thiết kế lớp MemberAddController



### Thiết kế lớp BookListController

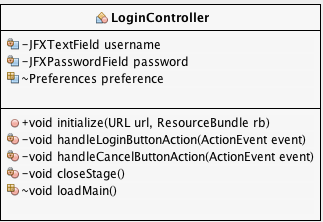


### Thiết kế lớp MemberListController

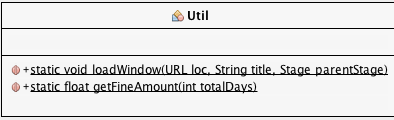


### Thiết kế lớp IssuedListController

### Thiết kế lớp Login Controller



### Thiết kế lớp Util



# Thiết kế mô hình dữ liệu

## Biểu đồ thực thể liên kết

<Vẽ biểu đồ thực thể liên kết bằng cách lấy toàn bộ các lớp thực thể cần lưu trữ lâu dài trong biểu đồ lớp. Tìm mối liên hệ giữa chúng. Nếu là mối quan hệ nhiều-nhiều, cần thêm thực thể mô tả mối quan hệ này>

## Chuẩn hoá

<Với CSDL quan hệ, các thực thể được ánh xạ thành các bảng và cài đặt liên kết bằng ràng buộc khoá ngoại. Thực hiện các bước chuẩn hoá để thu được cơ sở dữ liệu ở dạng chuẩn 3. Lựa chọn một hệ quản trị CSDL để thực hiện việc thiết kế CSDL trên đó (VD: SQLite, MySQL, SQL server…)>

### Các bước chuẩn hoá

### Biểu đồ liên kết giữa các bảng

## Thiết kế chi tiết bảng

### Bảng BOOK

CREATE TABLE BOOK

bookID varchar(200) primary key,

memberID varchar(200),

issueTime timestamp default CURRENT\_TIMESTAMP,

renew\_count integer default 0,

FOREIGN KEY (bookID) REFERENCES BOOK(id),

FOREIGN KEY (memberID) REFERENCES MEMBER(id)

### Bảng ISSUE

CREATE TABLE ISSUE

bookID varchar(200) primary key,

memberID varchar(200),

issueTime timestamp default CURRENT\_TIMESTAMP,

renew\_count integer default 0,

FOREIGN KEY (bookID) REFERENCES BOOK(id),

FOREIGN KEY (memberID) REFERENCES MEMBER(id)

### Bảng MEMBER

CREATE TABLE MEMBER

varchar(200) primary key,

name varchar(200),

mobile varchar(20),

email varchar(100)\