
NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

BỘ BÀI TẬP

TẬP THỂ TÁC GIẢ:

TS. VŨ THỊ HƯƠNG GIANG

TS. BÙI THỊ MAI ANH

TS. NGUYỄN NHẬT HẢI

TS. TRẦN NHẬT HÓA

TS. TRỊNH THÀNH TRUNG

THS. NGUYỄN MẠNH TUẤN

ĐƠN VỊ: KHOA KHOA HỌC MÁY TÍNH

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HÀ NỘI, 6/2023

MỤC LỤC

DANH MỤC BẢNG	3
DANH MỤC HÌNH VẼ	4
CHƯƠNG 2 VÒNG ĐỜI PHẦN MỀM.....	9
CHƯƠNG 3 PHƯƠNG PHÁP AGILE.....	15
CHƯƠNG 4 QUẢN LÝ DỰ ÁN PHẦN MỀM	20
CHƯƠNG 5 QUẢN LÝ CẤU HÌNH PHẦN MỀM.....	28
CHƯƠNG 6 KỸ NGHỆ YÊU CẦU PHẦN MỀM.....	39
CHƯƠNG 7 THIẾT KẾ PHẦN MỀM.....	49
CHƯƠNG 8 XÂY DỰNG PHẦN MỀM.....	80
CHƯƠNG 9 ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM.....	90
TÀI LIỆU THAM KHẢO	111

DANH MỤC BẢNG

Bảng 2-1: Thông tin chung của dự án	9
Bảng 3-1: Bảng ví dụ Product Backlog	16
Bảng 3-2: Bảng ví dụ Product Backlog	17
Bảng 4-1: Mô tả rủi ro dự án	25
Bảng 4-2: Đánh giá xác suất rủi ro dự án	26
Bảng 4-3: Đánh giá tác động rủi ro dự án	26
Bảng 6-1: Đặc tả chi tiết UC Tạo khoản thu	46
Bảng 7-1: Đặc tả màn hình giao diện đăng nhập - trường hợp người dùng có thể tự đăng ký tài khoản sử dụng phần mềm	65
Bảng 7-2: Đặc tả giao diện màn hình đăng nhập với vai trò biết trước	66
Bảng 7-3: Một số thông tin hướng dẫn thiết kế	72
Bảng 7-4: Các thuộc tính về tính dùng được của mẫu thử	78
Bảng 9-1: Các test case	97
Bảng 9-2: Các kết quả có thể	98
Bảng 9-3: Bảng rút gọn các kết quả có thể	98
Bảng 9-4: Kết quả thực hiện các test case	99
Bảng 9-5: Các test case kiểm thử hộp trắng với tạo khoản thu	103
Bảng 9-6: Bảng quyết định cho tạo khoản thu	104
Bảng 9-7: Các test case kiểm thử hộp đen với tạo khoản thu	105
Bảng 9-8: Các test case kiểm thử hộp trắng với thêm khoản thu	108
Bảng 9-9: Bảng quyết định với thêm khoản thu	109
Bảng 9-10: Các test case kiểm thử hộp đen với thêm khoản thu	110

DANH MỤC HÌNH VẼ

Hình 1-1: Ví dụ về mẫu giấy tờ thu chi đang thực hiện thủ công.....	7
Hình 2-1: Ví dụ về danh sách thành viên tham gia dự án	12
Hình 2-2: Ví dụ về master plan của dự án	13
Hình 3-1: Vào phần tạo Project trong Github.....	18
Hình 3-2: Tạo bảng công việc theo mô hình Scrum	18
Hình 3-3: Tạo bảng công việc theo mô hình Scrum	19
Hình 4-1: Cấu trúc phân cấp công việc theo sản phẩm bàn giao	23
Hình 4-2: Cấu trúc phân cấp công việc theo quy trình Waterfall	24
Hình 4-3: Cấu trúc phân cấp công việc theo quy trình Scrum.....	24
Hình 5-1: Tổ chức thư mục dự án theo các sản phẩm của dự án.....	30
Hình 5-2: Tổ chức thư mục dự án theo các pha trong tiến trình phát triển phần mềm	31
Hình 5-3: Lập bảng quản lý các mục cấu hình	31
Hình 5-4: Ví dụ tổ chức thư mục mã nguồn phần mềm.....	34
Hình 5-5: Tạo kho lưu trữ mới trên Github	34
Hình 5-6: Thông tin kho lưu trữ trên Github	35
Hình 5-7: Thêm thành viên vào kho lưu trữ trên Github.....	36
Hình 6-1: Nghiệp vụ quản lý nhân khẩu và hộ khẩu.....	41
Hình 6-2: Phân rã chức năng nghiệp vụ quản lý nhân khẩu và hộ khẩu	42
Hình 6-3: Nghiệp vụ quản lý thu phí và đóng góp.....	42
Hình 6-4: Phân rã chức năng nghiệp vụ quản lý thu phí và đóng góp	43
Hình 6-5: Usecase tổng quan module quản lý thu phí	44
Hình 6-6: Biểu đồ UC quản lý khoản thu	44
Hình 6-7: Biểu đồ Usecase quản lý đợt thu phí	44
Hình 6-8: Cấu trúc đặc tả chi tiết usecase	45
Hình 6-9: Biểu đồ hoạt động UC Tạo khoản thu	48
Hình 7-1: Phân rã usecase "Thêm khoản thu"	50
Hình 7-2: Biểu đồ trình tự cho usecase "Thêm khoản thu".....	51
Hình 7-3: Sơ đồ lớp phân tích cho usecase tổng hợp "Quản lý khoản thu"	52
Hình 7-4: Sơ đồ lớp phân tích cho usecase tổng hợp "Quản lý thu phí"	53
Hình 7-5: Mẫu kiến trúc MVC.....	54
Hình 7-6: Biểu đồ gói cho usecase “Thêm khoản thu”	54
Hình 7-7: Sơ đồ thực thể - quan hệ	58
Hình 7-8: Lược đồ các bảng dữ liệu.....	58
Hình 7-9: Sơ đồ lớp chi tiết cho ca sử dụng “Quản lý khoản thu”	61
Hình 7-10: Bố cục 1 - Khung cửa sổ 1120 * 645.....	63
Hình 7-11: Bố cục 2 - Khung cửa sổ 1440 * 1024.....	63
Hình 7-12: Bố cục 3 - khung cửa sổ 1440 * 1024	64
Hình 7-13: Màn hình đăng nhập theo vai trò người dùng đăng ký.....	66
Hình 7-14: Đăng nhập theo vai trò thiết lập sẵn	67
Hình 7-15: Màn hình trang chính theo bố cục 2	68
Hình 7-16: Màn hình trang chính theo bố cục 3	68
Hình 7-17: Màn hình tạo khoản thu theo bố cục 1.....	69
Hình 7-19: Màn hình thống kê theo bố cục 2	70
Hình 7-18: Màn hình thu phí theo bố cục 1.....	70
Hình 7-20: Màn hình thống kê theo bố cục 3	71
Hình 7-21: Các lựa chọn phối màu của màn hình đăng nhập.....	73
Hình 7-22: Các lựa chọn phối màu của màn hình trang chính	73
Hình 7-23: Các lựa chọn phối màu của màn hình tạo khoản thu.....	73
Hình 7-24: Các lựa chọn phối màu cho màn hình thu phí.....	74

Hình 7-25: Các lựa chọn phối màu của màn hình thống kê	74
Hình 7-26: Luồng duyệt tin theo cách phối màu thứ nhất.....	75
Hình 7-27: Luồng duyệt tin theo cách phối màu thứ hai.....	76
Hình 7-28: luồng duyệt tin theo cách phối màu thứ ba	77
Hình 7-29: Các bước thực hiện đánh giá mẫu thử	78
Hình 8-1: Thiết kế biểu đồ lớp cho các chức năng Khoản thu và Nộp tiền	82
Hình 8-2: Minh họa một đoạn chú thích bằng Javadocs	87
Hình 8-3: Tài liệu được sinh ra từ chú thích tài liệu Javadocs	87
Hình 8-4: Biểu đồ lớp sau khi refactor.....	89
Hình 8-5 : Bộ công cụ Inspect and Transform trong NetBeans.....	89
Hình 9-1: Cấu trúc của chương trình.....	91
Hình 9-2: Cơ sở dữ liệu trên phpMyAdmin	92
Hình 9-3: Màn hình chức năng login	93
Hình 9-4: TC2 cho chức năng login.....	98
Hình 9-5: Giao diện thêm khoản thu.....	100
Hình 9-6: Dữ liệu khoản thu	103
Hình 9-7: Giao diện đóng phí	106
Hình 9-8: Dữ liệu nộp tiền.....	108

GIỚI THIỆU BÀI TOÁN

Chung cư BlueMoon tọa lạc ngay ngã tư Văn Phú được khởi công xây dựng năm 2021 và hoàn thành vào 2023. Chung cư được xây dựng trên diện tích 450m², gồm 30 tầng, tầng 1 làm kiot, 4 tầng để, 24 tầng nhà ở và 1 tầng penhouse. Khi sở hữu nhà chung cư, hộ gia đình hoặc chủ sở hữu sẽ phải bỏ ra một khoản kinh phí đóng định kỳ để thực hiện vận hành và bảo dưỡng thường xuyên về cơ sở vật chất. Các hoạt động quản lý và thu phí ở chung cư BlueMoon được thực hiện bởi Ban quản trị chung cư do nhân dân sinh sống ở đây bầu ra.

Hàng tháng Ban quản trị chung cư lập **danh sách các khoản phí cần đóng với mỗi hộ gia đình** và gửi thông báo thu tiền. **Các khoản phí chung cư gồm** nhiều loại:

+ **Phí dịch vụ chung cư**, đây là loại phí bắt buộc nộp **theo tháng**, ban quản lý chung cư để chi trả vào các việc như: Lau dọn vệ sinh và bảo dưỡng các khu vực chung, làm đẹp cảnh quan của các khu vực chung; thu gom rác thải, bảo dưỡng sân vườn; đảm bảo an ninh... **Phí dịch vụ chung cư** được tính theo diện tích căn hộ sở hữu, hiện nay dao động từ 2.500 đồng tới 16.500 đồng/m²/tháng.

+ **Phí quản lý chung cư**, đây cũng là chi phí bắt buộc nộp **theo tháng**, dùng cho tất cả các hoạt động quản lý cũng như vận hành nhà chung cư. Chi phí này phụ thuộc vào tiêu chuẩn, chất lượng của dự án chung cư đó ví dụ như chung cư cao cấp, chung cư thường hay nhà chung cư giá rẻ. Với chung cư BlueMoon phí quản lý ở mức từ 7.000 đồng/m².

+ Các khoản đóng góp mà ban quản trị phối hợp với chính quyền địa phương, tổ dân phố để thực hiện thu (ví dụ quỹ vì người nghèo, quỹ biển đảo, quỹ từ thiện,...). Các khoản đóng góp này thu theo từng đợt, không bắt buộc và thu theo tinh thần tự nguyện.

Ban quản trị hiện đang quản lý việc thu phí theo phương thức thủ công, có sử dụng một số công cụ hỗ trợ như Excel nhưng hiệu quả quản lý chưa cao. Hiện tại Ban quản trị có nhu cầu xây dựng một phần mềm quản lý thu các loại phí tại chung cư BlueMoon.

Ví dụ một mẫu sổ quản lý thu các **khoản đóng góp**:

UBND PHƯỜNG KHUÔNG DINH
KHU DÂN CƯ 2

DANH SÁCH CÁC HỘ GÓP QUỸ TỰ NGUYỆN NĂM 2020...

TT	Họ và tên	Địa chỉ	Quỹ vì người nghèo	Quỹ vì biển đảo VN	Quỹ lễ Đình Gung	Quỹ tổ DP	Quỹ vì trẻ thơ	Quỹ nhân đạo, từ thiện	Quỹ tình nghĩa	Quỹ khuyến học	Quỹ người cao tuổi	Ký nhận
1	Phạm Văn Hải	50/1345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
2	Phạm Văn Hải	5/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
3	Phạm Văn Hải	5/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
4	Ngô Văn Hải	7/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
5	Ngô Văn Hải	11/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
6	Nguyễn Văn Hải	15/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
7	Trần Văn Hải	19/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
8	Ngô Văn Hải	23/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
9	Ngô Văn Hải	27/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
10	Trần Văn Hải	31/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
11	Trần Văn Hải	35/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
12	Trần Văn Hải	39/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
13	Trần Văn Hải	43/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
14	Trần Văn Hải	47/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
15	Trần Văn Hải	51/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
16	Trần Văn Hải	55/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
17	Trần Văn Hải	59/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
18	Trần Văn Hải	63/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
19	Trần Văn Hải	67/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
20	Trần Văn Hải	71/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
21	Trần Văn Hải	75/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
22	Trần Văn Hải	79/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
23	Trần Văn Hải	83/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
24	Trần Văn Hải	87/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
25	Trần Văn Hải	91/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
26	Trần Văn Hải	95/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
27	Trần Văn Hải	99/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
28	Trần Văn Hải	103/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
29	Trần Văn Hải	107/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị
30	Trần Văn Hải	111/50/345				100.000	20.000	20.000	20.000	20.000	20.000	Thị

Hình 1-1: Ví dụ về mẫu giấy tờ thu chi đang thực hiện thủ công

Trong phiên bản v1.0 của phần mềm, các chức năng cơ bản cần xây dựng cho Ban quản trị bao gồm: quản lý thông tin các khoản thu phí đóng góp, quản lý thu phí của các hộ gia đình. Ngoài ra phần mềm cũng cần cung cấp chức năng tra cứu, tìm kiếm và một số thông tin thống kê cơ bản giúp Ban quản trị nắm được hiện trạng các khoản thu. Nhằm giúp cho các hoạt động quản lý khác ở chung cư được thuận tiện và thông suốt, Ban quản trị muốn xây dựng thêm các chức năng quản lý thông tin cơ bản về các hộ gia đình (hộ khẩu) và nhân dân (nhân khẩu) đang sinh sống tại BlueMoon. Các chức năng này giúp Ban quản trị có thể cung cấp thông tin (chi tiết về hộ gia đình, nhân khẩu trong hộ, các hoạt động biến đổi nhân khẩu, tạm vắng, tạm trú,...) cho cơ quan chức năng khi được yêu cầu. Các chức năng này chỉ truy cập được sau khi Ban quản trị đăng nhập thành công với tài khoản đã cung cấp. Ban quản trị cũng có thể quản lý các thông tin cá nhân và thay đổi mật khẩu đăng nhập.

Trong phiên bản v2.0 phát triển tiếp theo của phần mềm, Ban quản trị muốn xây dựng thêm chức năng quản lý các khoản thu: Phí gửi xe ở chung cư: thu từng tháng

theo thông tin phương tiện đăng ký của hộ gia đình, trong đó phí gửi xe máy hàng tháng là 70.000/xe/một tháng và phí gửi ô tô là 1.200.000 nghìn đồng/xe/một tháng. Chi phí điện, nước, internet, đây là các khoản phí mà Ban quản trị thu hộ từng tháng theo thông báo từ các công ty cung cấp dịch vụ tương ứng.

Phần mềm dự kiến được phát triển dưới dạng một ứng dụng desktop với công nghệ Java, dữ liệu của phần mềm được lưu trữ tập trung trên MySQL server.

Luồng nghiệp vụ cần xử lý:

1. Đăng kí tài khoản
2. Tạo khoản thu
3. Thu phí
4. Thống kê các khoản đóng góp

CHƯƠNG 2 VÒNG ĐỜI PHẦN MỀM

Bài 2.1 Tổng hợp và phân tích dữ liệu của dự án

Hướng dẫn:

Tiến hành tổng hợp các thông tin cơ bản nhất của dự án. Đây là tài liệu ban đầu, xuyên suốt trong dự án. Về thông tin tổng hợp, có thể bao gồm nhiều khía cạnh, tùy theo độ phức tạp hay kiểu hình của dự án. Thông thường, tài liệu tổng hợp thông tin ban đầu trùng hay có nhiều nội dung sẽ được kế thừa dùng làm bản “Tuyên ngôn dự án” (Project charter)

Tài liệu này thường được thực hiện thông bởi người quản lý của đơn vị (hoặc cũng có thể là quản lý dự án), sau khi tìm hiểu và thu thập các thông tin, sẽ tiến hành báo cáo và gửi cho các đơn vị liên quan

Tài liệu này có thể được sử dụng với những tên khác như “Điều lệ dự án”, “Tờ trình chủ trương”, “Báo cáo nghiên cứu khả thi”.

Bảng 2-1: Thông tin chung của dự án

1. Thông tin chung	
Tên dự án	Tên dự án được sử dụng
Nhà tài trợ dự án	Tên đơn vị tài trợ, đơn vị cung cấp kinh phí cho dự án
Đơn vị thực hiện	Tên đơn vị tiến hành thực hiện dự án (có thể trùng với tên đơn vị tài trợ)
Phạm vi của dự án:	Mô tả phạm vi áp dụng của dự án hay những chức năng quan trọng hướng tới, hay hiệu quả mà sản phẩm dự án mang lại
Một số chú ý	Thông tin chú ý liên quan tới đặc thù của dự án.

Ngoài ra cũng cần cung cấp bổ sung nếu có như các thông tin về mục tiêu, hoàn cảnh ra đời của dự án, các thông tin về các bên liên quan hay đại diện các đầu mối sẽ phối hợp trong quá trình phát triển dự án.

Đây là tài liệu ban đầu, mang tính tổng hợp thông tin về dự án và khái quát hóa các tính năng, mục tiêu mà dự án hướng tới.

Bài 2.2 Tổng hợp và chuẩn bị các tài liệu phân tích kỹ thuật của dự án

Hướng dẫn:

Từ các thông tin thu thập được từ phần trước, tiến hành xây dựng các tài liệu mô tả kỹ thuật của dự án. Các tài liệu có thể bao gồm

- Tài liệu SRS (Software Requirement Specification): Là tài liệu yêu cầu có cấu trúc và chi tiết, gồm các yêu cầu về chức năng (function requirements), phi chức năng (non-function requirements) và tất cả các điều kiện khác mà phần mềm cần đáp ứng. Ví dụ trong bài toán của tài liệu này
 - + Về mặt chức năng/tính năng ghi rõ danh sách các chức năng của hệ thống quản lý thu phí trong chung cư. Việc này thường được mô tả thông qua các biểu đồ use-case của dự án (*Xem ví dụ ở phần Bài 6 của tài liệu này*)
 - + Để làm rõ về mặt tính năng, chức năng: có thể sử dụng bổ sung các bảng mô tả từng trường hợp use-case cụ thể; hoặc dùng các biểu đồ UML như biểu đồ sequence diagram để mô tả luồng nghiệp vụ tương ứng với use-case hay chức năng mong muốn.
 - + Về các yêu cầu phi chức năng: như việc phân quyền, độ mật của thông tin, thời gian phục vụ của hệ thống 24/7....
 - + Về các yêu cầu dữ liệu và thiết kế cơ sở dữ liệu dùng trong bài toán: Bao gồm sơ đồ liên kết, bảng mô tả rõ các trường dữ liệu cùng với kiểu dữ liệu tương ứng.

.....

Có thể tham khảo Chương 6 của tài liệu này để biết chi tiết cách thức xây dựng các biểu đồ use-case, chức năng sẽ được lưu trữ trong tài liệu này.

Bài 2.3 Đề xuất dự kiến quy trình, kế hoạch phát triển phần mềm phù hợp

Hướng dẫn:

Bước 1: Lựa chọn hình thức, mô hình quy trình phát triển phần mềm sẽ áp dụng trong dự án:

Tiến hành phân tích và trả lời một số câu hỏi để xác định và đề xuất mô hình sẽ áp dụng trong quá trình phát triển dự án. Các câu hỏi có thể ví dụ như sau:

1. *Phần mềm mới hay không?*
2. *Phạm vi áp dụng như thế nào?: phần mềm nội bộ, thay đổi cách thức quản lý thông tin thu phí và thông tin hộ dân cư ở chung cư Blue Moon*
3. *Vai trò của các bên liên quan ra sao?*
4. *Quy trình nghiệp vụ cụ thể đã rõ ràng chưa?: tương đối rõ khi chuyển đổi cách quản lý thu phí thông thường qua phần mềm. Tuy nhiên, cần phải làm rõ và điều chỉnh phù hợp*
5. *Kích thước phần mềm sẽ như nào?: nhỏ,*
6. *Đội ngũ lập trình cần thiết cho dự án khoảng bao nhiêu*

Sau khi trả lời các câu hỏi trên sẽ đề xuất mô hình phù hợp, song song với mô hình quản lý sẽ là các thành viên tham ra dự án cùng với các vai trò của thành viên. Ví dụ, áp dụng mô hình phát triển phần mềm linh hoạt (Agile) có thể có các đề xuất về thành viên như sau:

Giới thiệu thành viên dự án AEON

• Thẻ chế
~90 người

Asia-PM	TuanTA		Asia常駐サポーター JPMメンバー
Asia-QAM	HuyenNT		齋藤(Saito)
Asia-PL	HungNH		刈田(Karita)
			菅原(Sugawara)

	T1	T2	T3	T4	T5	T6	T7	デザイン	インフラ
Asia-BSE	DungNN	SonNH	ThinhNH	HungNH	HungNH	SonNH	DungNN	DungNT	
Asia-Devl	HungND	CuongNQ	DungNT	HungND	HungPV	DungNT	ThanhVV		
Asia-Devl-JP	稲葉 (Inaba)	稲葉 (Inaba)				稲葉 (Inaba)	稲葉 (Inaba)		
	NinhCTH	HaIDT	GiangNTT	HangNM	HoaiNTT	NinhCTH	NinhCTH		
Asia-QAL	HuongTT	GiangNTT		NinhCTH		HuongTT	HuongTT		
Asia-Dev	6	5	4	6	3	5	1	2	
Asia-QA	3	2.5	2	3	2	2.5	2.5		

Hình 2-1: Ví dụ về danh sách thành viên tham gia dự án
 Như vậy, chúng ta cần phải chỉ rõ các vai trò của thành viên trong dự án. Sau khi có thành viên, hiểu rõ nguồn lực mà dự án có thể sử dụng, chúng ta có thể tiến hành song song các bước 2-1 và 2-2 về kế hoạch tổng thể và các quy định cần chú ý trong quy trình.

Bước 2-1: Đề xuất kế hoạch tổng thể ban đầu của dự án:

Một bảng kế hoạch ban đầu có thể như hình bên dưới. Bao gồm các mốc thời gian đối với từng công đoạn trong các pha của phát triển phần mềm

dd/mm yyyy	11/12 2017	21/12 2017	12/1 2018		2/2 2018	15/2 2018	20/2 2018	22/2 2018	28/2 2018	1/3 2018	22/3 2018	23/3 2017	29/3 2018
設計 Design													
開発/単体 Code/Test		Milestone 1			Milestone 2								
結合 IT													
修正 FIX													
UAT													
dd/mm yyyy	11/12 2017	21/12 2017	12/1 2018		2/2 2018	15/2 2018	20/2 2018	22/2 2018	28/2 2018	1/3 2018	22/3 2018	23/3 2017	29/3 2018

Hình 2-2: Ví dụ về master plan của dự án

Bước 2-2: Đề nghị các quy định áp dụng cụ thể đối với quy trình

- Bao gồm các quy định về phân chia quản lý trách nhiệm
- Định nghĩa rõ các công việc/nhiệm vụ như thế nào được coi là xong. Quá trình xác định một công việc hay nhiệm vụ xong cần phải qua các giai đoạn nào. Ví dụ:
 - B1: nhận công việc, nhiệm vụ
 - B2: tiến hành phát triển
 - B3: Tiến hành kiểm thử, rà soát
 - B4-1: chuyển sang trạng thái hoàn thành nếu kiểm thử thành công
 - B4-2: Nếu kiểm thử không thành công thì chuyển sang B2.

(Chú ý: đây chỉ là ví dụ, tùy bài toán và yêu cầu chặt chẽ có thể quy định chi tiết hơn)

- Quy định về quản lý mã nguồn, quản lý phiên bản: Cần chỉ rõ công cụ, môi trường phát triển và quản lý phiên bản mã nguồn
- Quy định về quyền người được tiến hành deploy sản phẩm. Điều này là cần thiết đối với các dự án có nhiều thành viên. Khi đó, việc deployment cần phải là người có trách nhiệm và kinh nghiệm. Thường là người quản lý dự án.
- Quy định về vấn đề giao tiếp: giao tiếp qua kênh nào, công cụ nào, thời hạn phải cung cấp thông tin hay mức độ chi tiết về thông tin cần có....

- Các quy định khác nếu có: như các vấn đề liên quan đến khen/thưởng nội bộ nhóm, giờ giấc ,....

Chú ý: Tất cả các vấn đề trên cần phải được văn bản hóa và lưu lại trong quá trình thực hiện dự án.

Bài 2.4 Các tài liệu liên quan trong quá trình vận hành phát triển dự án

Hướng dẫn:

Xem xét và cập nhật tài liệu dự án thường xuyên. Quá trình vận hành của dự án cần liên tục cập nhật các thông tin:

- Điều này sẽ giữ cho các tài liệu có liên quan, chính xác và cập nhật.
- Sử dụng quy trình phân phối và cập nhật tài liệu, chẳng hạn như yêu cầu thay đổi, thông báo hoặc phản hồi.
- Lưu ý về quá trình kết thúc dự án, cùng với việc lưu trữ hoặc chuyển giao tài liệu sau đó.

Ví dụ:

- Cần cập nhật trạng thái của các công việc, nhiệm vụ thông qua biểu đồ Burn-Chart (hay các biểu đồ tương tự để theo dõi tiến độ dự án)
- Cập nhật các mô tả yêu cầu công việc ở mức chi tiết hơn, hay các phản hồi đánh giá đối với kết quả của từng công việc trong bảng quản lý công việc (Bảng quản lý thường theo mô hình Scum hay Kanban)

CHƯƠNG 3

PHƯƠNG PHÁP AGILE

Trong phần tài liệu này sẽ hướng dẫn tạo bảng quản lý theo Agile-Scrum thông qua công cụ Github (<http://github.com>). Như vậy, việc quản lý mã nguồn cũng như tiến độ công việc của dự án có thể thực hiện thông qua công cụ Github này.

Chú ý: Các hướng dẫn ở trong Chương này sẽ giả thiết người đọc đã có tài khoản Github, do đó sẽ không mô tả các bước tạo tài khoản, hay đăng ký tài khoản.

Bài 3.1 Lập danh sách các công việc cho Product Backlog

Hướng dẫn:

Khi một nhóm phát triển dự án có dự định sử dụng Framework Scrum vào quản lý dự án, thì trước khi có thể bắt đầu Sprint đầu tiên cần xây một Product Backlog. Đây là một danh sách ưu tiên (được sắp xếp 1, 2, 3, ...) của các tính năng hướng-khách hàng (customer-centric). Nghĩa là mức độ ưu tiên của các tính năng này sẽ dựa trên giá trị mà nó đem lại cho khách hàng. Ví dụ như: tạo ra giá trị sử dụng ngay, tạo ra các tính năng cơ bản trước để quảng cáo, hay tạo ra trang giới thiệu,....

Product Backlog tồn tại và có thể liên tục được cập nhật, thay đổi trong suốt quá trình phát triển sản phẩm. Tại bất cứ thời điểm nào, Product Backlog cũng là một nơi duy nhất và mang tính quyết định của danh sách công việc nhóm phát triển cần hoàn thành theo thứ tự ưu tiên.

Cách tạo ra Product Backlog

Bước 1: Liệt kê các Story, có thể mô tả bằng lời.

Ví dụ:

Story 1: Là một người sống trong chung cư, tôi muốn nộp một khoản lệ phí

Story 2: là một người sống trong chung cư, tôi muốn xem lịch sử các khoản phí mà tôi đã đóng

Story 3:

Bước 2: Sắp xếp các story vào bảng danh sách ưu tiên, đồng thời có thể bổ sung các thông tin về ước lượng công sức để hoàn thành nếu cần thiết.

Từ các thông tin của bảng này, chúng ta sẽ tiến hành áp dụng với công cụ Github được đề xuất.

Bảng 3-1: Bảng ví dụ Product Backlog

Độ ưu tiên	Hạng mục	Ước tính (ngày công)
1	Là người quản lý thu phí, tôi muốn tạo một khoản thu	15
2	Là người quản lý thu phí, tôi muốn tiến hành ghi nhận các khoản thu	20
3	Là người quản lý thu phí, tôi muốn tiến hành thống kê các khoản đóng góp của từng hộ gia đình	25
4	..	
5	..	
..	..	

Chú ý Product Backlog bao gồm nhiều hạng mục khác nhau, chủ yếu là các tính năng mới cho khách, hay các mục tiêu cải tiến, kỹ thuật trọng yếu....

Bài 3.2 Lập danh sách các công việc cho Sprint Backlog

Hướng dẫn:

Để lập kế hoạch cho Sprint chúng ta cần phải làm 2 việc chính:

- (1) Việc cần làm: Thành phần tham dự bao gồm: Product Owner, Nhóm Phát triển, ScrumMaster.
- (2) Thực hiện như thế nào: thành phần tham dự gồm : Nhóm Phát triển, ScrumMaster, Product Owner (không bắt buộc nhưng cần phải sẵn sàng để trả lời các thắc mắc).

Đối với công việc đầu tiên (1), có thể đưa ra mục tiêu của Sprint, hạng mục công việc trong 1 sprint cần có độ lớn bao nhiêu là vừa. Sau đó mỗi hạng mục cần phải tiếp tục

chia đủ nhỏ để con số ước tính thường thường một thành viên có thể thực hiện trong 1-2 ngày, tối ưu là thực hiện được trong ngày.

Đối với phần công việc thứ hai (2), nhìn vào lượng công việc đã sắp xếp theo thứ tự ưu tiên trong Product Backlog, nhóm phát triển lựa chọn công việc. Nghĩa là nhóm Phát triển quyết định lượng công việc mà họ sẽ hoàn thành, thay vì được giao bởi Product Owner.

Với mỗi hạng mục công việc, Nhóm Phát triển tạo một danh sách các nhiệm vụ nhỏ tương ứng. Các nhiệm vụ này có thể là được phân rã từ các hạng mục Product Backlog hoặc là chính bản thân các hạng mục Product Backlog nếu chúng đủ nhỏ để chỉ cần thực hiện trong vài giờ. Danh sách công việc cần hoàn thành trong Sprint được gọi là Sprint Backlog.

Bảng 3-2: Bảng ví dụ Product Backlog

Product Backlog	Sprint Backlog	Người thực hiện	Ước tính
Là người quản lý thu phí, tôi muốn tạo một khoản thu	Chỉnh sửa cơ sở dữ liệu	Nguyễn Văn A	2
	Tạo giao diện	Nguyễn Văn A	3
	Tạo thao tác bắt sự kiện qua Javascript	Nguyễn Văn A	5
	Cập nhật thông tin khoản thu vào csld	Nguyễn Văn A	5
Là người quản lý thu phí, tôi muốn tiến hành ghi nhận các khoản thu	Thiết kế giao diện nhập thông tin khoản thu	Nguyễn Văn B	
	Tạo thao tác bắt sự kiện nhập nhằm định dạng	Nguyễn Văn B	
	Lưu vào CSDL	Nguyễn Văn B	

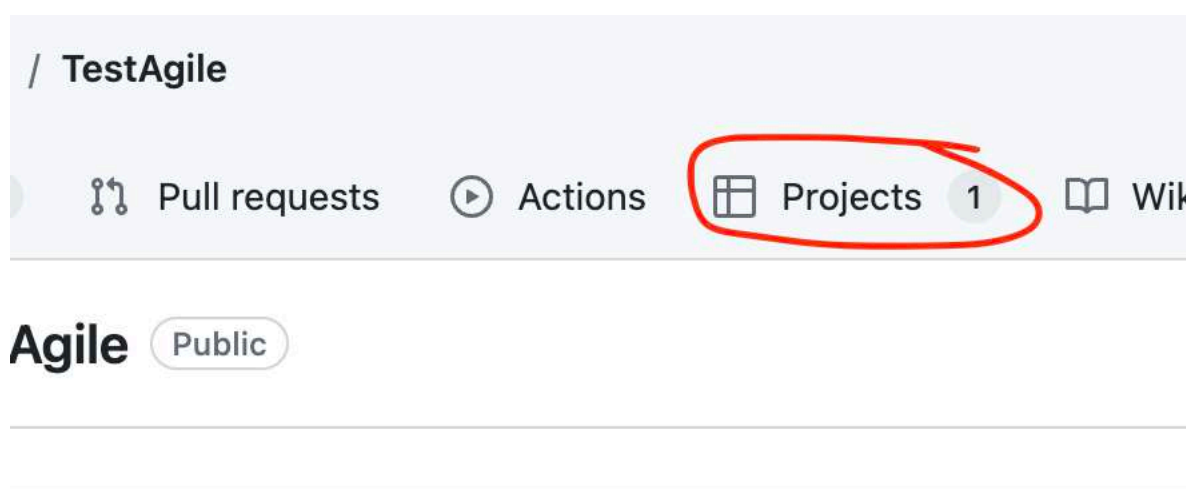
...	...		

Bài 3.3 Thiết lập bảng quản lý công việc trên Github

Bước 1: tạo 1 repository trên Github. Ví dụ “TestAgile”

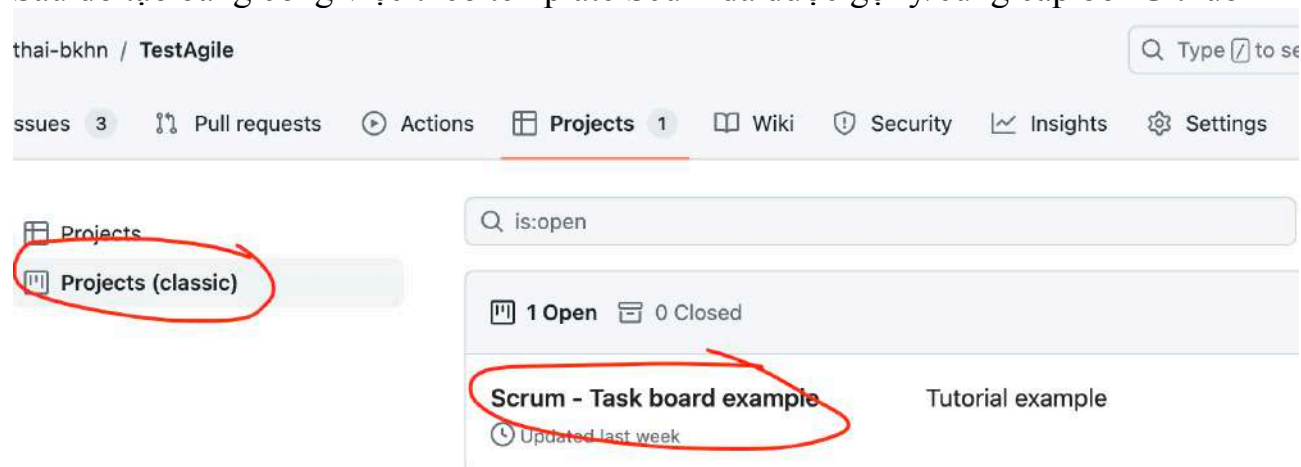
Bước 2: Tạo Project trong Repository

Đầu tiên, vào phần Project như trên hình



Hình 3-1: Vào phần tạo Project trong Github

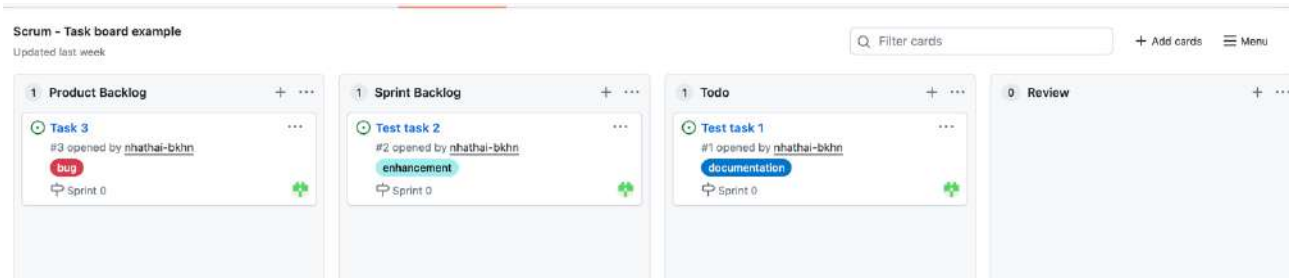
Sau đó tạo bảng công việc theo template Scum đã được gợi ý/cung cấp bởi Github



Hình 3-2: Tạo bảng công việc theo mô hình Scrum

Bước 3: Tạo các danh sách công việc trong Product Backlog và Sprint Backlog

- Sau khi tạo bảng công việc, sẽ được kết quả như hình dưới đây. Bảng công việc này gồm 5 cột: Product Backlog, Sprint Backlog, Todo, Review, Done



Hình 3-3: Tạo bảng công việc theo mô hình Scrum

- Sau khi có bảng công việc, chúng ta có thể thêm các đầu công việc vào từng bảng tương ứng (product backlog, sprint backlog) theo như hướng dẫn ở các phần trước

Các công việc được tạo ra trong bảng có thể kéo-thả (drap & drop) vào từng cột tùy theo trạng thái của công việc

CHƯƠNG 4 QUẢN LÝ DỰ ÁN PHẦN MỀM

Bài 4.1 Khởi tạo dự án (tuyên ngôn dự án)

Hướng dẫn:

Tuyên ngôn dự án (Project charter) là tài liệu quan trọng nhất cần viết khi bắt đầu một dự án. Tài liệu này do nhà tài trợ dự án ban hành, chính thức cho phép tồn tại của dự án và trao cho người quản lý dự án quyền áp dụng các nguồn lực của tổ chức vào các hoạt động của dự án. Tuyên ngôn dự án có thể được gọi với những tên khác như “Điều lệ dự án”, “Tờ trình chủ trương”, “Báo cáo nghiên cứu khả thi”.

Nội dung của tuyên ngôn dự án gồm những thông tin cốt lõi về dự án và sản phẩm phần mềm mà dự án dự định đáp ứng.

1. Thông tin chung				
Tên dự án:	Xây dựng phần mềm quản lý và thu phí ở chung cư Blue Moon			
Nhà tài trợ dự án:	Xxx			
Cơ quan tài trợ:	Công ty ABC			
Mức độ ảnh hưởng của dự án:	Phần mềm nội bộ, thay đổi cách thức quản lý thông tin thu phí và thông tin hộ dân cư ở chung cư Blue Moon			
2. Đội ngũ thực hiện				
	Họ và tên	Nơi công tác	Điện thoại	E-mail
Người quản lý dự án:				
Thành viên dự án:				
3. Các bên liên quan khác (vai trò, trách nhiệm của các cá nhân và đơn vị tham gia chính trong dự án)				
Đại diện khách hàng: ông Nguyễn Văn C, trưởng ban quản trị tòa nhà Blue Moon				
Đại diện người dùng cuối: bà Phạm Thị A, kế toán ban quản trị tòa nhà Blue Moon				
Đại diện bộ phận kinh doanh: ông Lê Văn B, nhân viên kinh doanh Công ty ABC				

4. Tuyên bố phạm vi dự án
<p>Bối cảnh và động cơ <i>Mô tả bối cảnh, động cơ thực hiện dự án</i></p> <p>Hiện tại, Ban quản trị chung cư BlueMoon đang quản lý việc thu phí theo phương thức thủ công, có sử dụng một số công cụ hỗ trợ như Excel nhưng hiệu quả quản lý chưa cao. Ban quản trị có nhu cầu xây dựng một phần mềm quản lý thu các loại phí tại chung cư BlueMoon, cũng như hỗ trợ các công việc quản lý khác tại chung cư được thông suốt.</p>
<p>Mục đích của dự án <i>Mô tả các nhu cầu nghiệp vụ cần giải quyết</i></p> <p>Xây dựng phần mềm hỗ trợ ban quản trị chung cư BlueMoon quản lý thông tin về các loại phí cần đóng góp (phí chung cư, phí dịch vụ, tiền thu hộ, khoản đóng góp) của các hộ gia đình, quản lý thông tin về các hộ gia đình (hộ khẩu) và nhân dân (nhân khẩu) đang sinh sống tại chung cư.</p>
<p>Mục tiêu <i>Mô tả các kết quả mong muốn đạt được theo cách có thể đo lường được</i></p> <p>Phần mềm phải:</p> <ul style="list-style-type: none"> - Quản lý được 100% các loại phí cần thu. - Quản lý được 100% thông tin về các hộ gia đình sinh sống tại chung cư BlueMoon và các biến động nhân khẩu liên quan đến từng căn hộ.
<p>Sản phẩm bàn giao <i>Liệt kê danh sách các đầu mục sản phẩm cần bàn giao</i></p> <p>Cho khách hàng:</p> <ul style="list-style-type: none"> - Bộ cài đặt phần mềm hoàn chỉnh cho các máy tính cá nhân sử dụng các phiên bản khác nhau của hệ điều hành Windows - Chứng nhận Bản quyền phần mềm - Hướng dẫn sử dụng - Hướng dẫn cài đặt <p>Cho chủ đầu tư:</p> <ul style="list-style-type: none"> - Mã nguồn phần mềm - Các tài liệu kỹ thuật: tài liệu đặc tả yêu cầu, tài liệu phân tích thiết kế, tài liệu kiểm thử, tài liệu mô tả cấu trúc mã nguồn, API docs - Các tài liệu dự án
<p>Phạm vi <i>Liệt kê những gì dự án sẽ làm hoặc không làm</i></p> <p>Phạm vi dự án bao gồm các công việc đặc tả yêu cầu, phân tích, thiết kế, lập trình, kiểm thử và triển khai phần mềm; không bao gồm việc nhập liệu thông tin đầu vào cho phần mềm như nhân khẩu, hộ khẩu, các loại phí.</p> <p>Phạm vi sản phẩm: phần mềm gồm 3 module chính:</p> <ul style="list-style-type: none"> - Module quản lý thu phí - Module quản lý người dùng - Module quản lý hộ gia đình
<p>Ngân sách</p> <p>Ngân sách của dự án là 100.000.000 đồng, bao gồm kinh phí xây dựng phần mềm, quản lý dự án và kinh phí dự phòng.</p>
<p>Lịch trình thực hiện <i>Các mốc thời gian chính (bắt đầu, kết thúc) cho các pha trong dự án và các mốc khác</i></p> <p>Dự án được thực hiện trong quý IV 2023.</p> <ul style="list-style-type: none"> - Bàn giao phiên bản thử nghiệm (First user acceptance test): 30/11/2023 - Bàn giao phiên bản chính thức (Second user acceptance test): 15/12/2023 - Thời gian hoàn thành: 30/12/2023

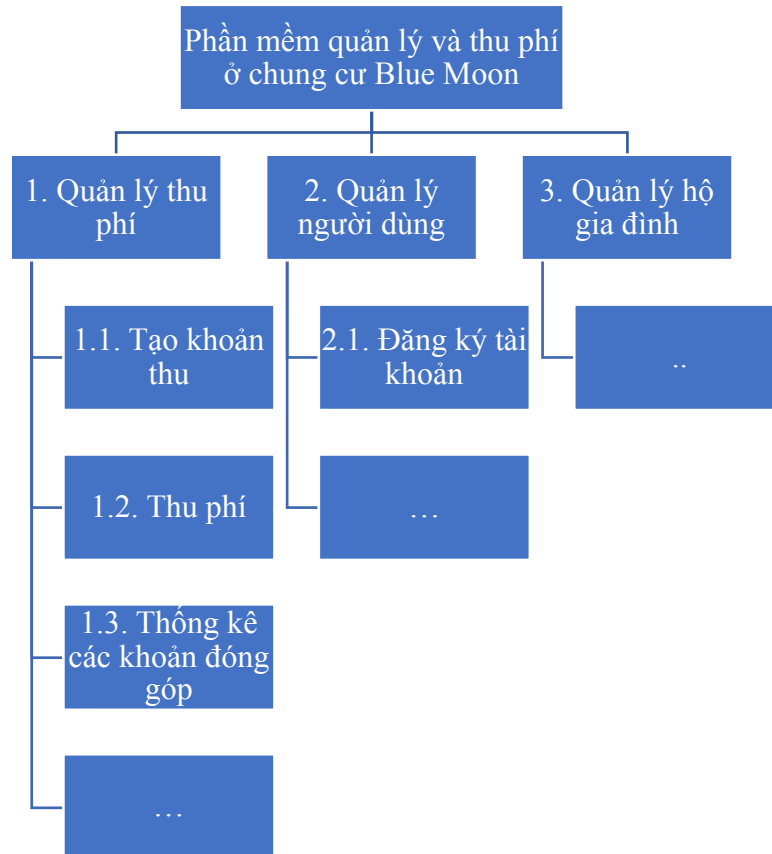
Rủi ro và giả định <i>Liệt kê các trở ngại có thể khiến dự án thất bại</i>			
Rủi ro / Giả định	Xác suất xảy ra	Tác động	
Rủi ro kỹ thuật: ...			
Rủi ro chi phí: ...			
...			
Giả định về nhân lực: ...			
Giả định về máy móc thiết bị: ...			
Giả định về khách hàng: - Người dùng cuối sẽ dùng thử phần mềm phiên bản thử nghiệm, đóng góp ý kiến để hoàn thiện phiên bản chính thức.			
Giả định về phạm vi: - Phạm vi dự án sẽ không thay đổi sau khi các bên liên quan đã ký kết			
Ràng buộc <i>Liệt kê bất kỳ điều kiện nào có thể hạn chế các lựa chọn của nhóm dự án liên quan đến nguồn lực, nhân sự hoặc lịch trình (ví dụ: ngân sách định trước hoặc ngày kết thúc dự án, giới hạn số lượng nhân viên có thể được phân công cho dự án).</i>			
- Dự án sẽ không kéo dài quá 4 tháng. Nếu chậm tiến độ, có thể tính đến phương án bổ sung thêm nhân lực thực hiện nếu không vượt quá ngân sách dự án. - Ngân sách dự án đã dự tính kinh phí dự phòng trong trường hợp có chi phí vượt mức, vì vậy không được phép vượt quá vì bất kỳ lý do gì.			
Các yếu tố phụ thuộc bên ngoài <i>Liệu sự thành công của dự án có phụ thuộc vào sự phối hợp nỗ lực giữa nhóm dự án và các bên liên quan khác không? Các bên liên quan có đồng ý phối hợp không?</i>			
5. Chiến lược truyền thông <i>cách người quản lý dự án sẽ liên lạc với nhà tài trợ dự án, nhóm dự án và các bên liên quan, ví dụ: tần suất báo cáo tiến độ, tần suất các cuộc họp của nhóm dự án, v.v.</i>			
- Gửi báo cáo tiến độ dự án hàng tuần cho nhà tài trợ và đội dự án qua email - Họp nhanh với các thành viên dự án vào đầu giờ sáng hàng ngày - Họp đánh giá kết quả dự án tại mỗi mốc thời gian theo lịch trình thực hiện với khách hàng, nhà tài trợ và đội thực hiện dự án.			
6. Phê duyệt			
	Họ và tên	Chữ ký	Ngày
Nhà tài trợ dự án			
Cơ quan tài trợ dự án			
Người quản lý dự án			
7. Ghi chú			

Bài 4.2 Lập kế hoạch dự án (Work Breakdown Structure, Gantt Chart)

Hướng dẫn:

Với các dự án phần mềm, có 2 cách phổ biến để xây dựng cấu trúc phân cấp công việc là xây dựng cấu trúc phân cấp công việc theo sản phẩm bàn giao, và xây dựng cấu trúc phân cấp công việc theo quy trình.

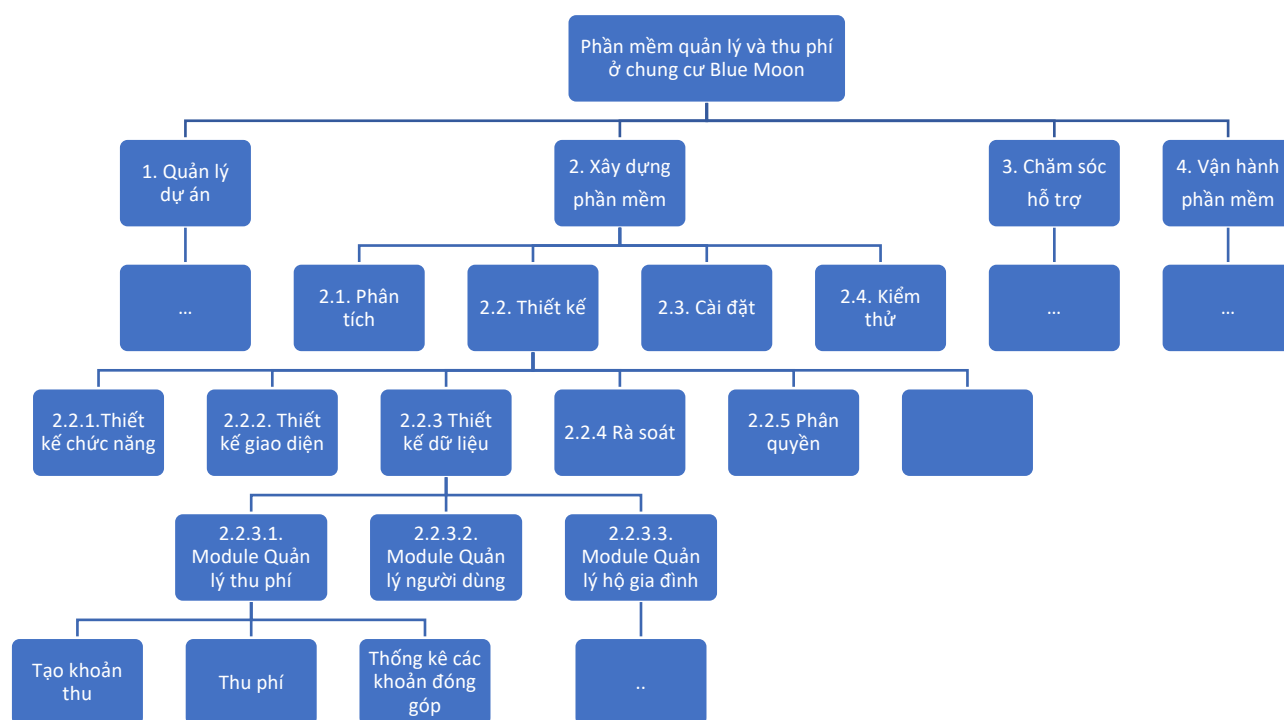
- Để xây dựng cấu trúc phân cấp công việc theo sản phẩm bàn giao, mức 0 chính là phần mềm, mức 1 chính là các module phần mềm, các mức tiếp theo chính là các tính năng phần mềm. Mỗi tính năng sẽ trở thành 1 gói công việc.



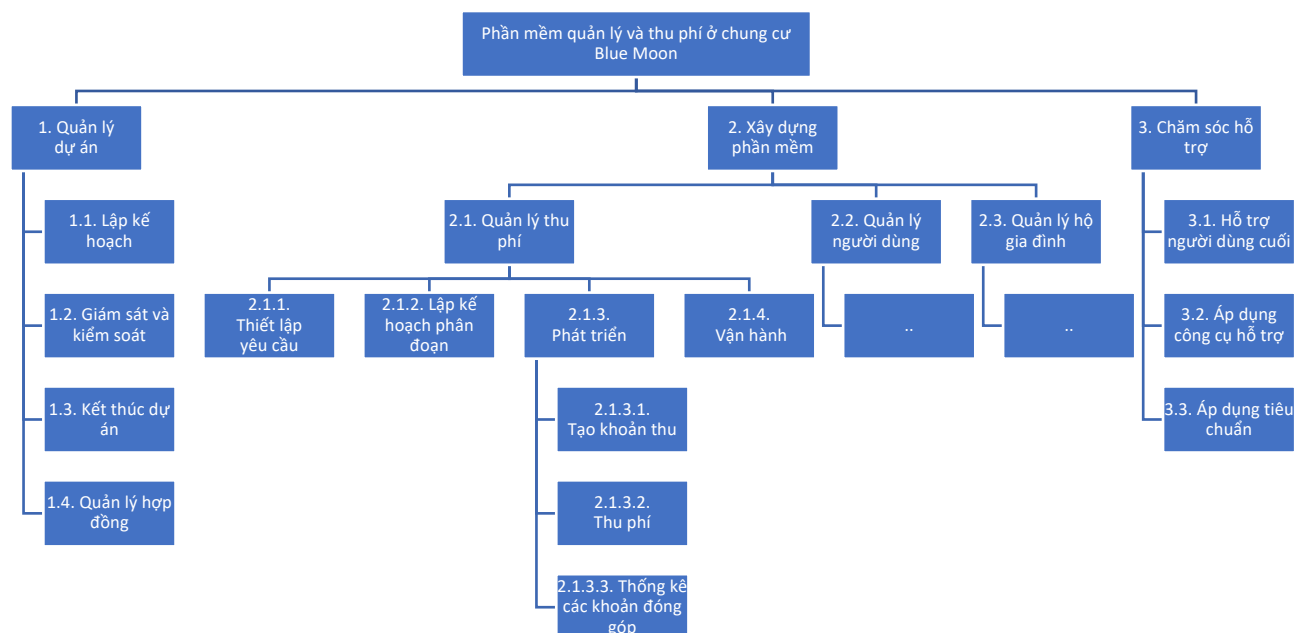
Hình 4-1: Cấu trúc phân cấp công việc theo sản phẩm bàn giao

Trong một số trường hợp, gói công việc là quá lớn để có thể giao cho 1 người thực hiện. Khi đó, ta cần tiếp tục chia nhỏ gói công việc thành các hoạt động cần thiết để tạo lập được sản phẩm của gói công việc đó.

- Để xây dựng cấu trúc phân cấp công việc theo quy trình, mức 0 chính là phần mềm, mức 1 chính là các pha trong phạm vi dự án. Tiếp tục phân chia các pha này thành các công việc cần thực hiện để có thể tạo ra sản phẩm bàn giao tương ứng với từng pha. Ví dụ:



Hình 4-2: Cấu trúc phân cấp công việc theo quy trình Waterfall



Hình 4-3: Cấu trúc phân cấp công việc theo quy trình Scrum

Lưu ý quy tắc 100%: khi hoàn thành hết các gói công việc lớp dưới, gói công việc lớp trên cũng hoàn thành.

Bài 4.3 Lập kế hoạch quản lý rủi ro

Hướng dẫn:

Bảng 4-1: Mô tả rủi ro dự án

STT	Mô tả	Xác suất	Tác động	Mức độ rủi ro cho dự án	Hành động ngăn ngừa rủi ro
1	Nhân sự phải tham gia nhiều dự án cùng lúc hoặc rời dự án	Thấp (1)	Vừa (2)	2 (Minor)	<ul style="list-style-type: none"> - Chia đều các công việc quan trọng cho các thành viên. - Chuẩn bị nhân sự dự phòng. - Chuẩn bị phương án phân chia lại khối lượng công việc cho các thành viên - Đảm bảo mọi nhân sự trong dự án nắm được yêu cầu cụ thể về hiệu suất cá nhân.
2	Các bên liên quan thay đổi yêu cầu hoặc bổ sung yêu cầu mới sau khi dự án bắt đầu	Vừa (2)	Thấp (1)	2 (Minor)	<ul style="list-style-type: none"> - Có hợp đồng, văn bản, sự chấp thuận yêu cầu cụ thể của các bên liên quan - Xây dựng quy trình xử lý thay đổi - Đàm phán lại về thời gian hoàn thành và chi phí phát sinh
3	Các tài liệu kỹ thuật và tài liệu thiết kế kèm theo mã nguồn chưa hoàn thiện	Thấp (1)	Cao (3)	3 (Vừa)	<ul style="list-style-type: none"> - Theo dõi, kiểm tra sản phẩm giao nộp trung gian, xử lý kịp thời các vấn đề.
4	Thiếu sự tương tác, trao đổi đa chiều dẫn đến hiểu lầm, bất đồng quan điểm	Vừa (2)	Vừa (2)	4 (Vừa)	<ul style="list-style-type: none"> - Tổ chức các cuộc họp giữa các bên liên quan để gỡ rối các vấn đề hiện tại, xem xét và tiếp nhận các ý kiến đóng góp của thành viên - Xây dựng và gợi ý về tình huống sử dụng giả định cho khách hàng - Thiết lập thêm kênh trao đổi trực tiếp giữa khách hàng với nhóm phát triển (qua Zalo, điện thoại).
5	Sự chậm trễ trong các giai đoạn dự án trước đó	Thấp (1)	Vừa (2)	2 (Minor)	<ul style="list-style-type: none"> - Sắp xếp công việc theo thứ tự trước sau và thứ tự ưu tiên - Tổ chức hội thảo công bố kế hoạch dự án và cấu trúc phân chia công việc.
6	Hành động pháp lý trì hoãn hoặc tạm dừng dự án	Thấp (1)	Vừa (2)	2 (Minor)	<ul style="list-style-type: none"> - Kiểm tra kỹ các điều khoản hợp đồng trước khi ký - Bảo cáo cấp trên
7	Khách hàng từ chối nghiệm thu sản phẩm	Thấp (1)	Vừa (2)	2 (Minor)	<ul style="list-style-type: none"> - Kiểm tra kỹ các điều khoản về bàn giao sản phẩm trước khi ký hợp đồng - Thiết lập và thỏa thuận với khách hàng về các tiêu chí nghiệm thu sản phẩm một cách định lượng - Kiểm thử với dữ liệu, hệ thống thật nếu có thể
8	Cắt giảm ngân sách bất ngờ	Thấp (1)	Thấp (1)	1 (Trivial)	Có những nguồn ngân sách dự phòng
9	Ước tính thời gian và chi phí quá lạc quan	Vừa (2)	Vừa (2)	4 (Vừa)	Cần phải họp kỹ, tham khảo nhiều bên để đưa ra được ước tính thời gian và chi phí phù hợp

Trong đó, xác suất rủi ro được đánh giá như sau:

Bảng 4-2: Đánh giá xác suất rủi ro dự án

Xác suất	Giải thích
Thấp (1)	Rất hiếm khi xảy ra hoặc được cho là sẽ không xảy ra
Vừa (2)	Đã xảy ra thỉnh thoảng hoặc có lý do hợp lý nhưng không hoàn toàn cơ hội dự kiến xảy ra trong tương lai
Cao (3)	Cho thấy dấu hiệu rủi ro cao, đã xảy ra thường xuyên, có ảnh hưởng rất lớn cơ hội xảy ra trong tương lai

Tác động của rủi ro được đánh giá theo 3 mức như dưới đây:

Bảng 4-3: Đánh giá tác động rủi ro dự án

Tác động	Về phạm vi	Về chất lượng	Về chi phí
Thấp (1)	< 5% quy mô dự án bị ảnh hưởng Hoặc Các khu vực nhỏ trong phạm vi dự án bị ảnh hưởng	Hiệu suất chất lượng giảm < 5% Mục tiêu	< 5% nỗ lực dự án / tăng chi phí
Vừa (2)	5% - 10% quy mô dự án bị ảnh hưởng	Giảm chất lượng yêu cầu sự chấp thuận của nhà tài trợ Hoặc Hiệu suất chất lượng giảm 5%-10% Mục tiêu	Giảm chất lượng yêu cầu sự chấp thuận của nhà tài trợ Hoặc Hiệu suất chất lượng giảm 5%-10% Mục tiêu
Cao (3)	>10% quy mô dự án bị ảnh hưởng	Giảm chất lượng không thể chấp nhận để tài trợ Hoặc Hiệu suất chất lượng giảm >10% Mục tiêu	>10% nỗ lực dự án/tăng chi phí

Sắp xếp các rủi ro theo thứ tự ưu tiên: Sau khi ước lượng xác suất và tác động, các rủi ro được sắp xếp thứ tự ưu tiên theo mức độ ảnh hưởng của chúng đến các mục tiêu của dự án.

Như được mô tả trong bảng ma trận xác suất và tác động bên dưới, sự kết hợp giữa xác suất và tác động dẫn đến rủi ro được xếp thành 5 mức từ 1 – 5, trong đó mức 5 là nghiêm trọng nhất.

Mức 5: Không thể tiếp tục thực hiện dự án.

Mức 4: Tác động đáng kể đến dự án, một số công việc của dự án không thực hiện được

Mức 3: Tác động đến dự án, nhưng dự án vẫn thực hiện được.

Mức 4: Không ảnh hưởng đáng kể đến dự án.

Mức 5: Hầu như không ảnh hưởng hoặc ảnh hưởng không rõ ràng.

	Ảnh hưởng		
Xác suất	Thấp (1)	Vừa (2)	Cao (3)
Thấp (1)	Mức 1	Mức 2	Mức 3
Vừa (2)	Mức 2	Mức 3	Mức 4
Cao (3)	Mức 3	Mức 4	Mức 5

CHƯƠNG 5

QUẢN LÝ CẤU HÌNH PHẦN MỀM

Bài 5.1 Xây dựng cấu trúc thư mục dự án

Hướng dẫn:

1. Tổ chức thư mục dự án

Cấu trúc thư mục và đặt tên tài liệu rõ ràng và nhất quán sẽ giúp việc định vị tệp nhanh chóng dễ dàng hơn, bao gồm các phiên bản mới nhất của tệp. Tổ chức thư mục dự án tốt giúp giảm sự nhầm lẫn và khả năng xảy ra các lỗi có thể gây tổn kém. Đồng thời tạo điều kiện thuận lợi cho việc truy xuất và cập nhật tệp một cách tin cậy từ kho lưu trữ của dự án.

Để đạt được mục tiêu nói trên, nhóm dự án cần phát triển các nguyên tắc và hướng dẫn đơn giản và linh hoạt phù hợp với bối cảnh thực tiễn. Các thành viên tham gia dự án cũng cần được phổ biến về những nguyên tắc này. Ngoài ra việc xây dựng một cấu trúc thư mục mẫu đủ linh hoạt có thể được sử dụng lại cho các dự án tiếp theo.

Sẽ là không thực tế nếu mong đợi các kho lưu trữ của dự án được tổ chức một cách hoàn hảo vì nó bị ảnh hưởng bởi các quy trình quản lý dự án và quy trình phát triển phần mềm phức tạp. Tuy nhiên, việc quản lý tốt các tài liệu dự án có thể đạt được qua các thiết kế chu đáo ngay từ đầu với lưu ý đến việc tổ chức lưu trữ các kết quả cuối cùng khi kết thúc dự án.

Bước 1: *Chọn định dạng chuẩn cho thư mục và tài liệu dự án*

- Sử dụng các mẫu thư mục có sẵn hoặc làm theo hướng dẫn như PMBOK Guide. Có thể tạo định dạng của riêng mình dựa trên nhu cầu của dự án. Điều quan trọng là đảm bảo rằng định dạng dễ đọc, dễ hiểu và dễ cập nhật.

Bước 2: *Xây dựng quy ước đặt tên phù hợp*

- Giúp xác định, sắp xếp và tìm kiếm chúng một cách nhanh chóng và chính xác. Sau khi được thiết lập, hệ thống định danh này sẽ được sử dụng nhất quán cho các dự án trong tương lai.
- Sử dụng tên ngắn gọn, đơn giản, có tính mô tả.

- Khi đánh số tập tin hoặc thư mục, các số 0 ở đầu sẽ giúp các mục được sắp xếp chính xác (ví dụ: 01, 02 hoặc 001, 002 thay vì 1, 2).
- Sử dụng dấu gạch dưới hoặc dấu gạch ngang thay vì dấu cách (ví dụ: file-name.docx; file_name.docx).
- Sử dụng mã ngày gồm sáu hoặc tám chữ số trong tên tệp hoặc thư mục, kết hợp năm, tháng và ngày (ví dụ: 20231022).
- Không sử dụng các ký tự 'đặc biệt' có thể gây khó khăn cho việc tìm kiếm (ví dụ: dấu ngoặc nhọn, dấu sổ chéo, ký hiệu đặc biệt v.v.).
- Mỗi thư mục dự án (project folder) cần được gán một mã định danh duy nhất, có thể là sự kết hợp ký tự với chữ số. Ví dụ như: Số thứ tự cho từng dự án hoặc mã cho mỗi khách hàng.
- Đối với tài liệu dự án, có thể sử dụng tiền tố hoặc hậu tố để biết trạng thái tài liệu, chẳng hạn như bản nháp, bản cuối cùng, đã được phê duyệt hoặc đã lưu trữ (draft, final, approved, or archived).

Bước 3: *Tạo cấu trúc thư mục dự án*

- Cấu trúc thư mục dự án nên bao gồm một thư mục dự án chính và một loạt các thư mục con thích hợp. Thư mục dự án chính phải được gán nhãn bằng mã định danh duy nhất. Các thư mục con có thể được đặt tên theo các giai đoạn dự án điển hình chẳng hạn như bắt đầu, lập kế hoạch, thực hiện, giám sát và kiểm soát cũng như kết thúc.
- Có thể tạo các thư mục cho các mục đích cụ thể, chẳng hạn như liên lạc, quản lý rủi ro hoặc kiểm soát thay đổi.

Bước 4: *Lựa chọn vị trí lưu trữ*

- Giúp truy cập các tài liệu và thư mục một cách dễ dàng và an toàn từ mọi nơi và mọi lúc.
- Sử dụng giải pháp lưu trữ tại chỗ hoặc dựa trên đám mây, chẳng hạn như Google Drive, Dropbox, SharePoint hoặc ổ đĩa mạng, có thể kết hợp sử dụng phần mềm hoặc công cụ quản lý dự án, chẳng hạn như Asana, Trello hoặc Microsoft Project.

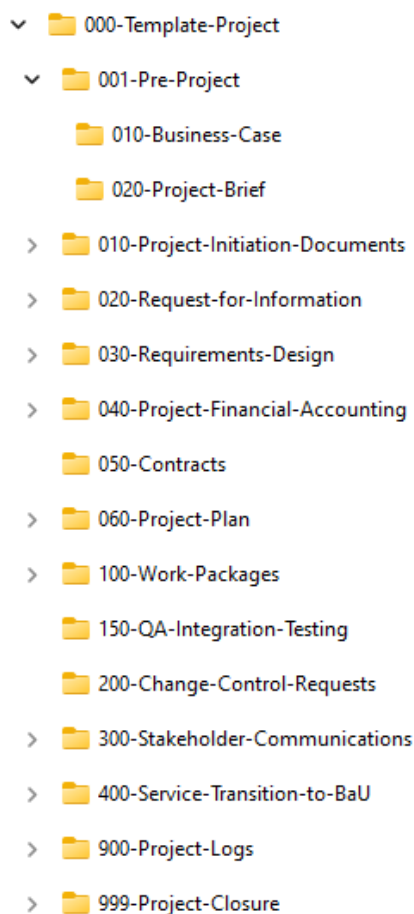
Bước 5: *Triển khai kiểm soát truy cập và kiểm soát phiên bản cho tài liệu dự án*

- Có thể sử dụng mật khẩu, quyền hoặc vai trò để hạn chế những người có thể xem, chỉnh sửa hoặc xóa tài liệu trong thư mục dự án.
- Sử dụng các hệ thống kiểm soát phiên bản để ghi dấu thời gian, nhật ký lịch sử hoặc bản sao lưu để theo dõi ai đã thực hiện thay đổi nào và khi nào.

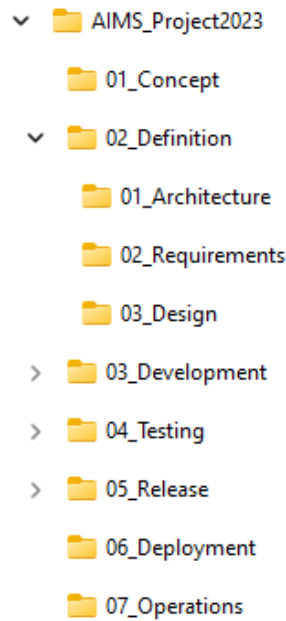
Bước 6: *Xem xét và cập nhật tài liệu dự án thường xuyên*

- Điều này sẽ giữ cho các tài liệu có liên quan, chính xác và cập nhật.
- Sử dụng quy trình phân phối và cập nhật tài liệu, chẳng hạn như yêu cầu thay đổi, thông báo hoặc phản hồi.
- Lưu ý về quá trình kết thúc dự án, cùng với việc lưu trữ hoặc chuyển giao tài liệu sau đó.

Vi dụ tổ chức thư mục dự án:



Hình 5-1: Tổ chức thư mục dự án theo các sản phẩm của dự án



Hình 5-2: Tổ chức thư mục dự án theo các pha trong tiến trình phát triển phần mềm

Lập một bảng chi tiết cho các mục cấu hình gồm tài liệu và cấu trúc thư mục. Chỉ có các mục trong khu vực có kiểm soát được đề cập ở đây. Quy ước đặt tên theo quy ước sẽ được dùng để đặt tên cho tất cả các tài liệu và các tập tin nguồn.

Mục cấu hình (Configuration item)	Tên (Name)	Khu vực làm việc (Working area)	Khu vực xem xét lại (Review area)	Khu vực phát hành (Release area / Baseline)
Project scope document	ScopeDocument.docx			IT3180-GroupX/01-ManagedArea/Scope/
Project plan	ProjectPlan.docx	IT3180-GroupX/04-Users/<UserID>/ProjectDocs/	IT3180-GroupX/05-Reviews/ProjectDocs/	IT3180-GroupX/01-ManagedArea/ProjectDocs/
...				
...				

Hình 5-3: Lập bảng quản lý các mục cấu hình

2. Tổ chức thư mục mã nguồn phần mềm

Việc tổ chức mã nguồn có vai trò rất quan trọng để duy trì và cập nhật phần mềm theo thời gian. Để đảm bảo khả năng bảo trì tối ưu, điều quan trọng là áp dụng cách tiếp cận có cấu trúc để tổ chức các tệp mã nguồn. Thuật ngữ “cấu trúc” ở đây có nghĩa là cách thức chúng ta tổ chức các tệp mã nguồn và sắp xếp chúng vào các thư

mục một cách logic với các phân phụ thuộc rõ ràng. Khi tìm kiếm câu trả lời cho các câu hỏi dưới đây, nhà phát triển sẽ định hình sản phẩm hoàn thiện trông như thế nào:

- Những chức năng của sản phẩm nên gắn với mô-đun nào?
- Các tệp mã nguồn nào có thể được nhóm lại với nhau hoặc cần phải tách biệt?
- Luồng dữ liệu đi qua dự án như thế nào?
- ...

Dưới đây là một số hướng dẫn về tổ chức thư mục mã nguồn phần mềm:

2.1 Thư mục gốc

Thư mục gốc sẽ bao gồm các tệp tin liên quan đến toàn bộ mã nguồn dự án, đây là điểm khởi đầu đối với hầu hết các nhà phát triển và được hiển thị theo mặc định khi truy nhập vào dự án, ví dụ như các kho lưu trữ mã nguồn trên GitHub. Không nên đặt quá nhiều thứ vào thư mục gốc vì nó khiến việc điều hướng trở nên khó khăn. Một số tệp tin và thư mục phổ biến có thể được đặt trong thư mục gốc:

- **README (Readme.md)**: tệp mô tả các thông tin cơ bản về phần mềm, các hướng dẫn về cách thực thi phần mềm, bao gồm các tham chiếu thư viện, các khóa cần thiết lập, cách tạo cơ sở dữ liệu, v.v. Có thể bao gồm liên kết đến bản demo trực tiếp (nếu có phiên bản được triển khai).
- **CHANGELOG**: liệt kê những tính năng mới, đã sửa đổi hoặc đã ngừng trên mỗi phiên bản hoặc bản sửa đổi - thường theo thứ tự thời gian đảo ngược để thuận tiện theo dõi (những thay đổi mới nhất được hiển thị trên đầu tệp)
- **LICENSE**: một tệp chứa nội dung của giấy phép sao chép và sử dụng, bao gồm một số thông tin bản quyền bổ sung (chẳng hạn như giấy phép của bên thứ ba).
- **.gitignore**: tệp tin của hệ thống kiểm soát phiên bản Git, cho biết những tệp nào không được đồng bộ hóa với kho lưu trữ.
- Thư mục **src**: chứa mã nguồn của toàn bộ ứng dụng
- Thư mục **build / bin / dist**: chứa các tệp được biên dịch từ mã nguồn

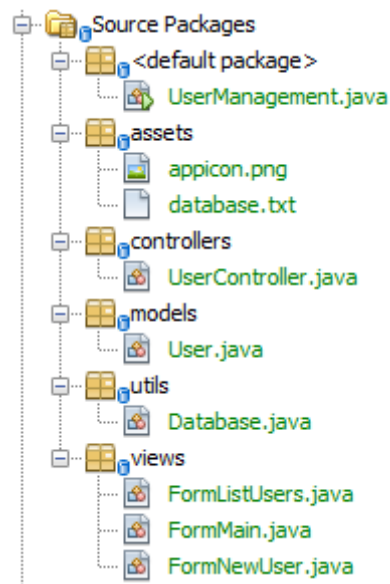
- Thư mục **docs**: chứa các tệp tin tài liệu cho phần mềm
- Thư mục **test**: chứa các dữ liệu và tệp script cho kiểm thử
- Thư mục **config**: cấu hình cho ứng dụng, ví dụ như tệp môi trường (.env). Thư mục này cũng có thể nằm trong từng phần (máy khách, máy chủ, v.v.) sẽ tùy thuộc vào từng ứng dụng.
- Thư mục **scripts / tools**: lưu trữ các công cụ và tập lệnh script (để đặt lại cơ sở dữ liệu, tạo báo cáo, v.v.). Ví dụ nếu dự án sử dụng docker thì có thể chưa tệp được sử dụng để khởi động ứng dụng: Dockerfile / docker-compose.yml
- Các thư mục khác: data, libs,...

2.2 Thư mục src

Thư mục **src** sẽ chứa các mô-đun phần mềm. Mọi thứ trong đó đều tác động trực tiếp đến cách nhà phát triển viết mã nguồn và cách ứng dụng hoạt động. Các lựa chọn tổ chức thư mục src phổ biến:

- Tổ chức các tệp mã nguồn theo mô-đun tính năng / chức năng. Mỗi mô-đun phản ánh các thực thể miền nghiệp vụ mà nó liên quan đến.
- Nhóm các tệp mã nguồn theo mục đích của tệp, ví dụ các thư mục riêng cho hình ảnh, video, css,...
- Cấu trúc theo yêu cầu của framework hoặc thư viện, ví dụ các thư mục models, views, controllers,...

Ví dụ một số thư mục con trong thư mục src



Hình 5-4: Ví dụ tổ chức thư mục mã nguồn phần mềm

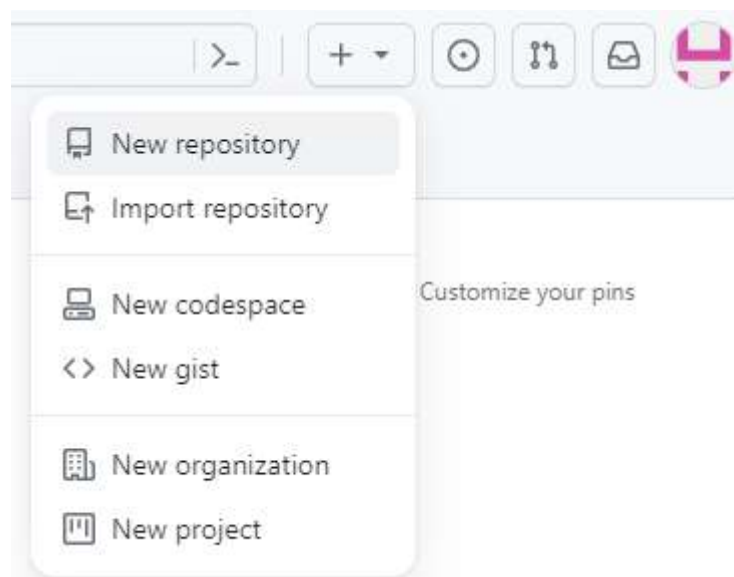
Bài 5.2 Tạo lập kho lưu trữ dự án

Hướng dẫn:

1. Tạo mới kho lưu trữ dự án trên Github

Bước 1: Đăng nhập vào Github, nếu chưa có tài khoản thì tạo một tài khoản miễn phí

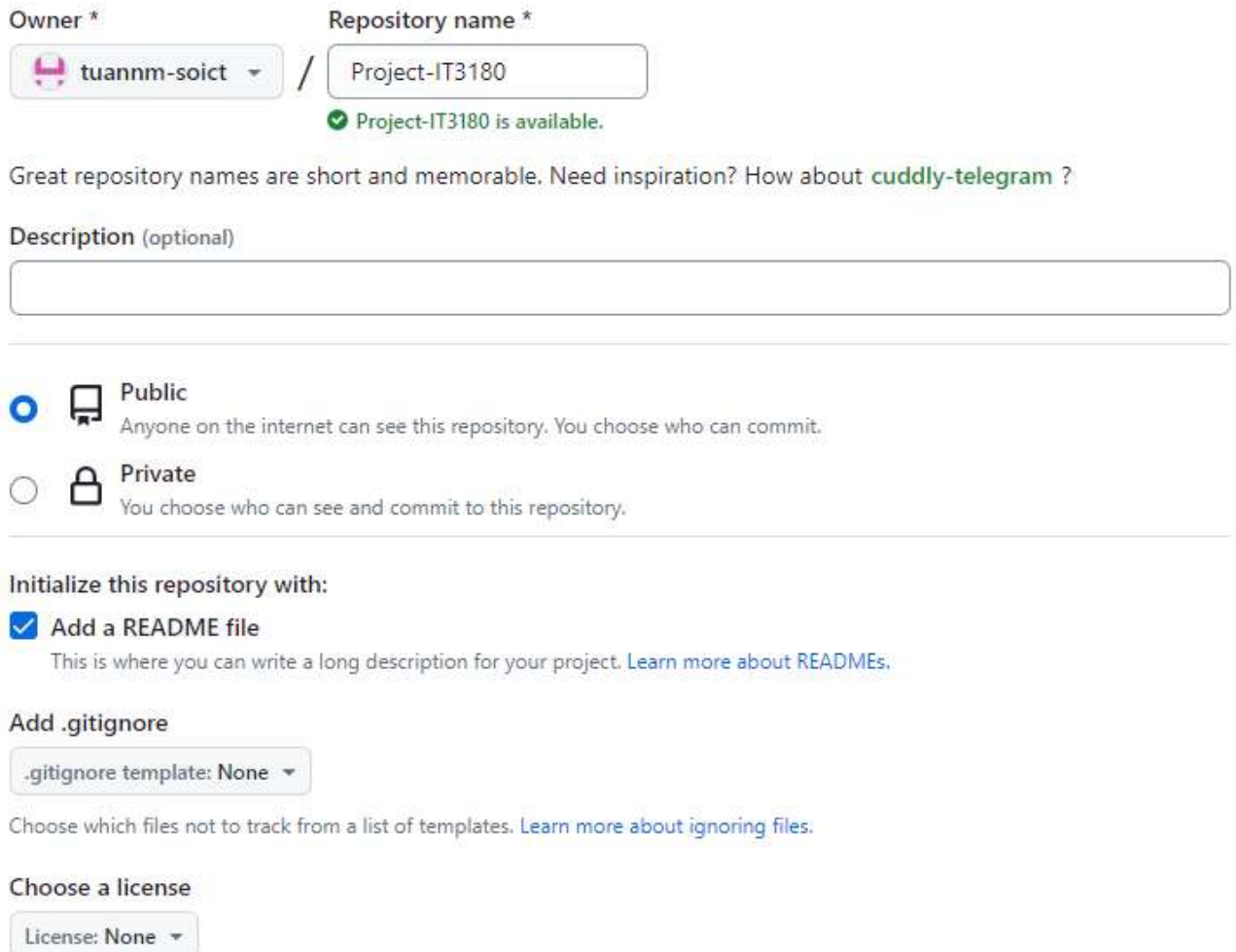
Bước 2: Nhấn vào nút dấu + ở góc trên bên phải và chọn mục New repository trong danh sách để tạo một kho lưu trữ (repository) mới




Hình 5-5: Tạo kho lưu trữ mới trên Github

Bước 3: Nhập các thông tin về kho lưu trữ, bao gồm:

- Tên kho lưu trữ
- Mô tả
- Lựa chọn kho lưu trữ là Public / Private
- Một số tệp tin khởi tạo theo mẫu: README, License, .gitignore




Owner * Repository name *


 tuannm-soict / Project-IT3180

✔ Project-IT3180 is available.

Great repository names are short and memorable. Need inspiration? How about [cuddly-telegram](#) ?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

Hình 5-6: Thông tin kho lưu trữ trên Github

Bước 4: Nhấn nút Create repository, một kho lưu trữ trống mới được tạo thành công.

Địa chỉ sau khi tạo thành công trong hướng dẫn này là:

<https://github.com/tuannm-soict/Project-IT3180>

2. Tạo mới kho lưu trữ dự án cục bộ

Tạo kho lưu trữ dự án cục bộ trên máy của nhà phát triển có thể thực hiện theo 2 cách:

Cách 1: Tạo bản sao từ kho lưu trữ từ xa trên Github, sử dụng lệnh git clone

```
git clone https://github.com/tuannm-soict/Project-IT3180
```

Cách 2: Tạo kho lưu trữ dự án cục bộ rồi sau đó kết nối với kho lưu trữ từ xa trên Github

```
git init
```

```
git add "temp.txt"
```

```
git commit -m "Tao moi file";
```

```
git remote add origin https://github.com/tuannm-soict/Project-IT3180
```

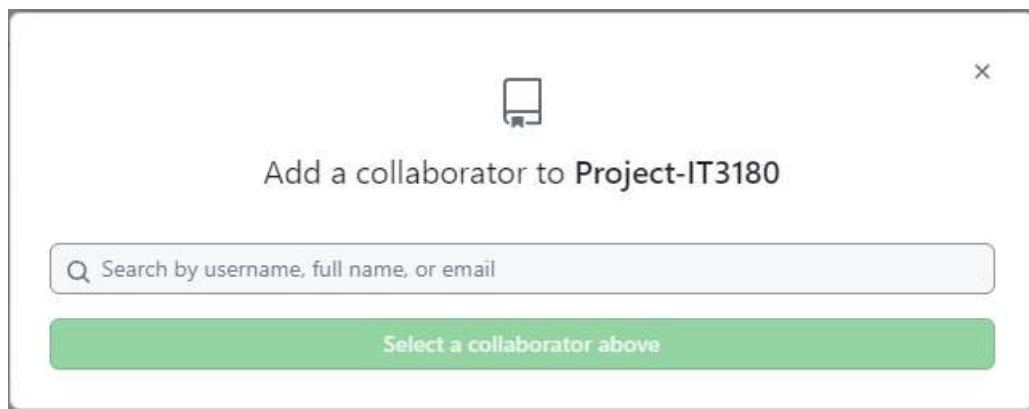
Tuỳ theo yêu cầu có thể sẽ cần phải nhập username và password đăng nhập của tài khoản Github để đưa dữ liệu lên.

3. Thêm thành viên vào kho lưu trữ Github

Bước 1: Truy cập vào Github, tại màn hình kho lưu trữ dự án → truy cập tab Setting

Bước 2: Chọn mục Collaborators bên phải → nhấn nút Add people

Bước 3: Nhập vào email hoặc username Github của thành viên



Hình 5-7: Thêm thành viên vào kho lưu trữ trên Github

Bài 5.3 Thao tác với kho lưu trữ dự án

Hướng dẫn:

Luồng công việc khi thao tác với Git và Github:

- **Bước 1:** Chọn một bug/feature để giải quyết
- **Bước 2:** *Checkout* nhánh phù hợp
- **Bước 3:** *Pull* các bổ sung mới nhất từ kho lưu trữ chính
- **Bước 4:** Tạo một *branch* cho bug/feature yêu cầu
- **Bước 5:** Thực hiện viết code / lập trình & *commit*
- **Bước 6:** *Push* các thay đổi đến các kho lưu trữ từ xa (remote repository)
- **Bước 7:** Tạo một *pull request* đến nhánh chính

Pull Request là một tính năng của hệ thống quản lý mã nguồn (source control management) như Git, GitHub, hay GitLab. Nó là một cách để nhà phát triển có thể đề xuất thay đổi vào mã nguồn của một dự án đã được lưu trữ trên hệ thống quản lý mã nguồn đó. Pull request được quan tâm sau khi nhà phát triển đã hoàn tất việc viết mã và sẵn sàng chuyển tiếp sang bước xét duyệt mã nguồn.

- Nếu cần thay đổi hay chỉnh sửa mã nguồn, hãy lặp lại quy trình

Thực hiện các thao tác cơ bản:

- Thay đổi nội dung mã nguồn dưới local repo, lệnh sử dụng: `git status`, `git add`, `git commit`
- Đưa sự thay đổi lên remote repo, lệnh sử dụng: `git push`, `git fetch`, `git pull`
- Thay đổi mã nguồn trong cùng một nội dung file dưới local repo và trên remote repo sau đó giải quyết vấn đề xung đột mã nguồn. Câu lệnh `git merge` có chức năng chính là hợp nhất hai nhánh và tự động xử lý xung đột. Tuy nhiên, có những xung đột mà không thể tự xử lý được, nó sẽ đánh dấu và dừng quá trình merge. Những conflict này bắt buộc phải xử lý thủ công. Cuối cùng là thực hiện commit lại code sau khi đã xử lý xung đột, các bước như sau:
 - Chạy lệnh `git status` để kiểm tra lại lần nữa những thay đổi
 - Chạy lệnh `git add [file name]` để thêm những file thay đổi vào stage

- Thực hiện commit các thay đổi lên local repo bằng lệnh : `git commit -m "nội dung message"`
- Cuối cùng là push code bằng lệnh `git push`
- Khôi phục một file với phiên bản cũ hơn. Những file đã được commit vào Git luôn có thể phục hồi lại. Một trong những cách phục hồi phổ biến thường dùng khi bạn commit quá sớm/vội và có thể quên thêm vào đó một số tập tin hoặc là thông điệp commit không như ý muốn. Nếu như bạn muốn thực hiện lại commit đó, bạn có thể chạy lệnh commit với tham số `--amend`
`$ git commit --amend`

CHƯƠNG 6

KỸ NGHỆ YÊU CẦU PHẦN MỀM

Bài 6.1 Đặc tả chung về dự án và xác định các tác nhân chính.

Dự án có thể được đặc tả ngắn gọn như sau:

- Thông tin về nhân khẩu của mỗi hộ gia đình được lưu trong sổ hộ khẩu với một mã định danh duy nhất. Sổ hộ khẩu gia đình sẽ lưu trữ các thông tin như: số hộ khẩu, họ tên chủ hộ, số nhà, đường phố (thôn ấp), phường (xã), quận (huyện)
- Sổ hộ khẩu thêm các chi tiết cho các nhân khẩu trong hộ như họ và tên, bí danh, ngày tháng năm sinh, nơi sinh, nguyên quán, dân tộc, nghề nghiệp, nơi làm việc, số CMND hoặc số CCCD, ngày cấp, nơi cấp, các thông tin lưu trú trước khi chuyển đến
- Các hoạt động biến đổi nhân khẩu: Thêm nhân khẩu mới, thay đổi nhân khẩu, đăng ký thường trú, đăng ký tạm trú
- Hàng năm khu chung cư thực hiện thu một số khoản phí và đóng góp của các hộ gia đình, công việc này do cán bộ kế toán phụ trách. Khoản phí vệ sinh là bắt buộc với tất cả các hộ gia đình, mỗi năm thu 1 lần với định mức 6.000VNĐ/ 1 tháng/ 1 nhân khẩu.
- Cán bộ kế toán sẽ lập danh sách các hộ gia đình và số nhân khẩu tương ứng, sau đó đến từng nhà thu phí và ghi nhận số tiền nộp.
- Đối với các khoản đóng góp thì không quy định số tiền mà phụ thuộc vào từng hộ, các khoản đóng góp này được thu theo từng đợt của các cuộc vận động như: “Ủng hộ ngày thương binh- liệt sỹ 27/07”, “Ủng hộ ngày tết thiếu nhi”, “Ủng hộ vì người nghèo”, ...
- Cán bộ kế toán cũng cần thống kê tổng số tiền đã thu trong mỗi đợt, tổng số hộ đã nộp và có thể xem chi tiết mỗi hộ đã nộp những khoản tiền nào.

Hướng dẫn xác định tác nhân và nghiệp vụ:

Hệ thống có 2 tác nhân:

- Tổ trưởng/Tổ phó: Người sử dụng hệ thống với chức năng quản lý hộ khẩu, nhân khẩu và cũng là người quản lý, phân quyền cho các tài khoản trong hệ thống
- Kế toán: Người sử dụng hệ thống liên quan đến quản lý thu phí, quản lý các khoản đóng góp của các hộ dân

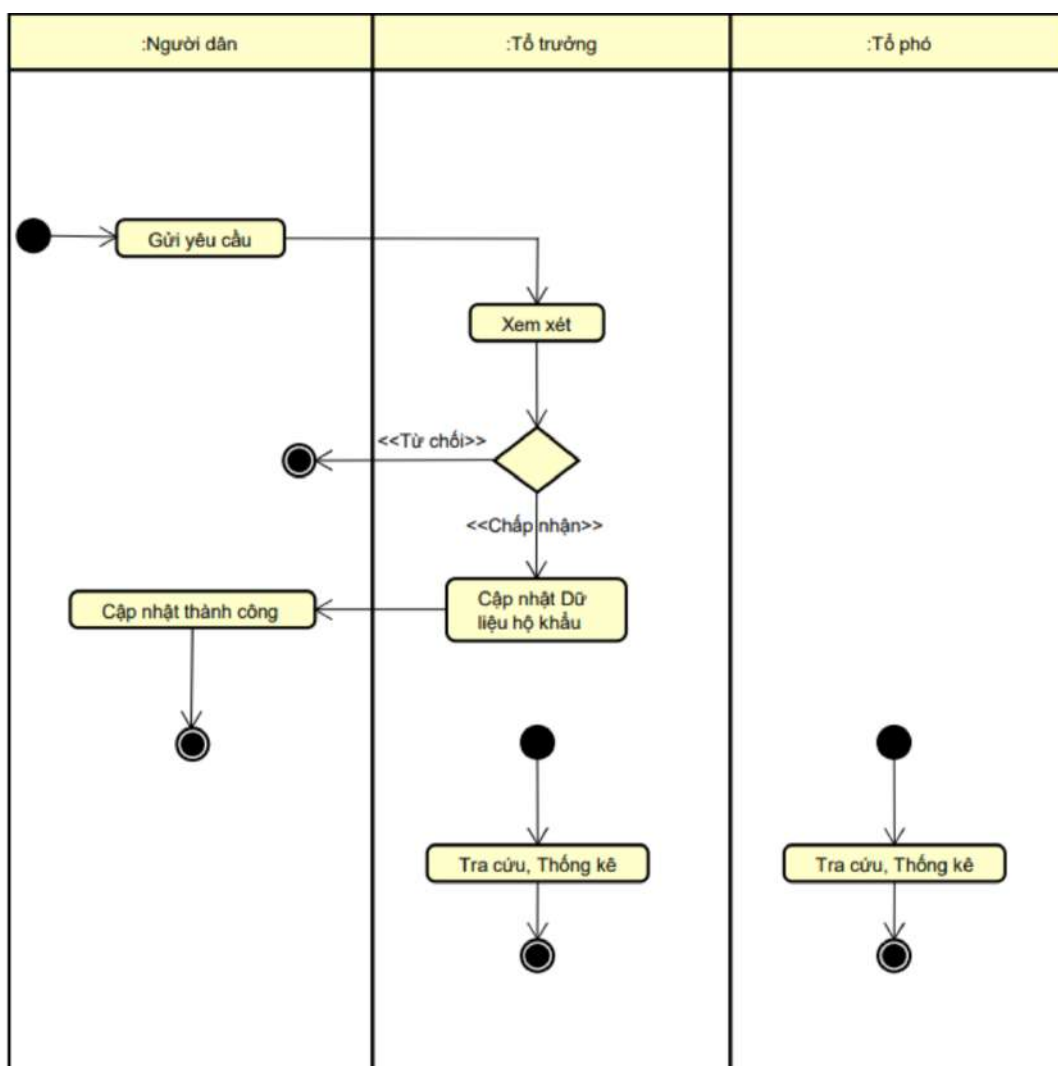
Hệ thống có 2 luồng nghiệp vụ chính

- Quản lý thông tin hộ khẩu và nhân khẩu
 - Quản lý biến đổi nhân khẩu
 - Thống kê
 - Truy vấn
- Quản lý thu phí và đóng góp
 - Lập danh sách các khoản thu
 - Thống kê
 - Truy vấn

Bài 6.2 Sử dụng biểu đồ hoạt động mô hình hoá nghiệp vụ và phân rã chức năng của hệ thống

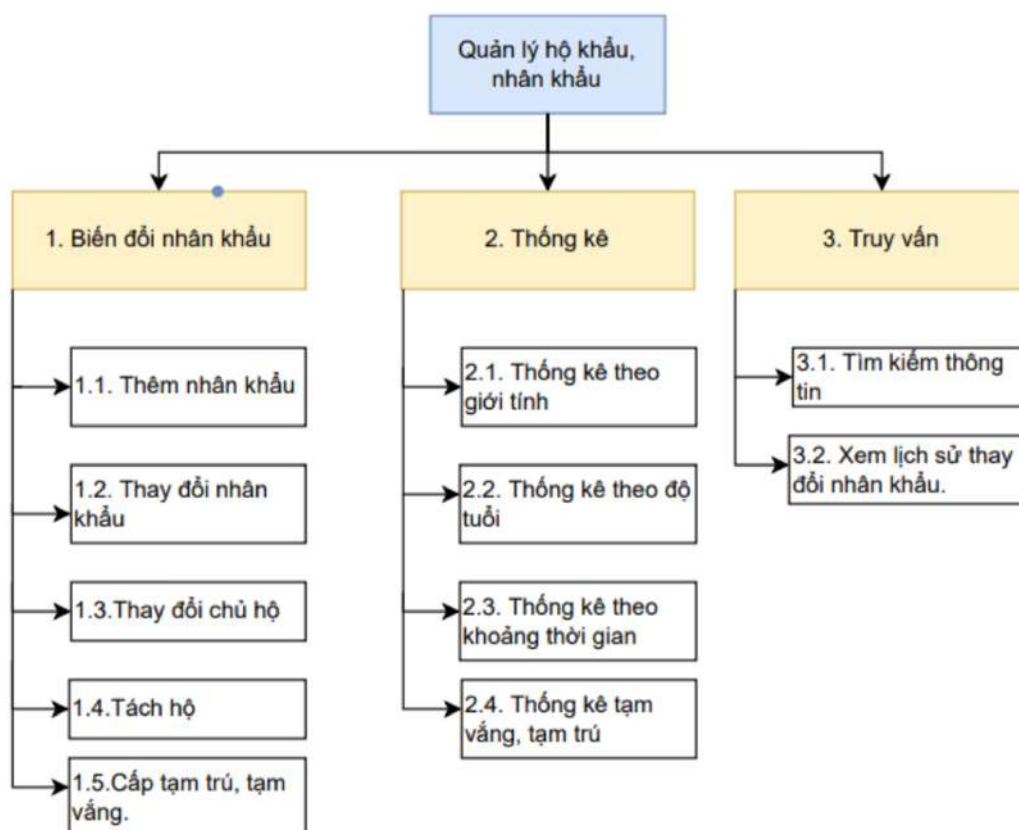
Hướng dẫn: Xác định các đối tượng tham gia trong nghiệp vụ và phân làn rồi xây dựng luồng hành động

Nghiệp vụ quản lý hộ khẩu và nhân khẩu



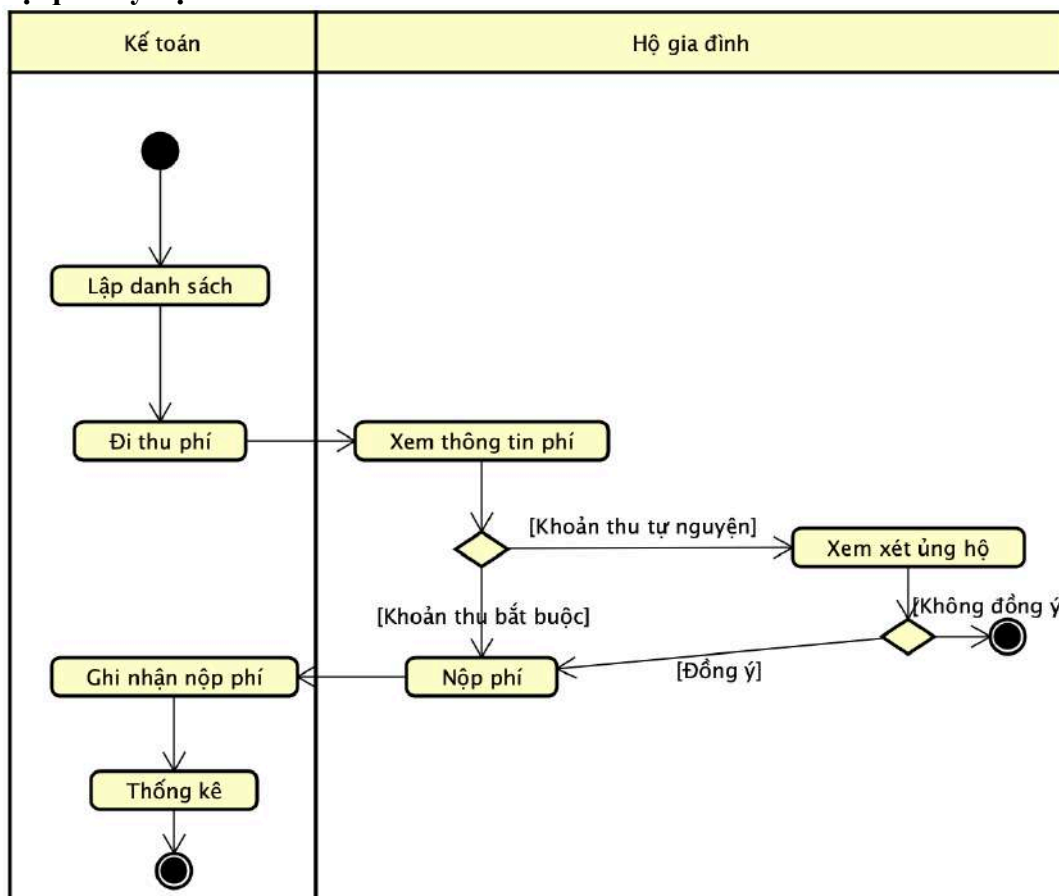
Hình 6-1: Nghiệp vụ quản lý nhân khẩu và hộ khẩu

Từ nghiệp vụ nói trên chúng ta gom nhóm các chức năng và xây dựng sơ đồ phân rã chức năng cho nghiệp vụ quản lý nhân khẩu và hộ khẩu như trong hình dưới đây.



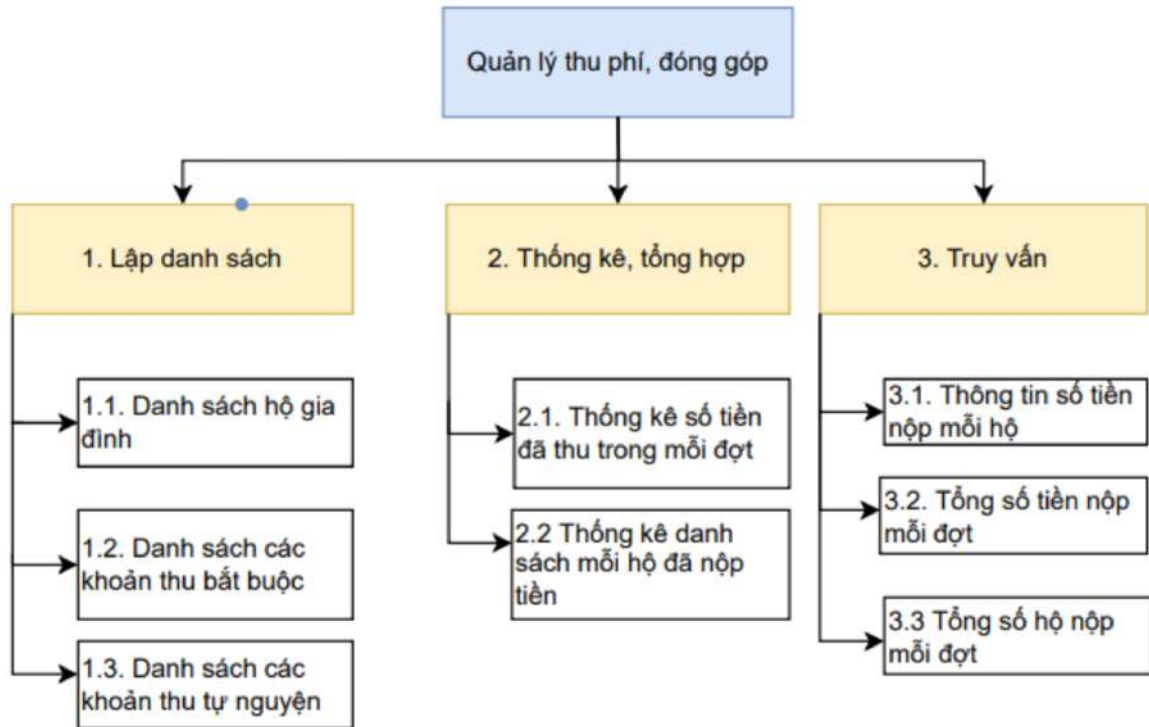
Hình 6-2: Phân rã chức năng nghiệp vụ quản lý nhân khẩu và hộ khẩu

Nghiệp vụ quản lý hộ khẩu và nhân khẩu



Hình 6-3: Nghiệp vụ quản lý thu phí và đóng góp

Phân rã các chức năng cho nghiệp vụ quản lý thu phí và đóng góp như trong sơ đồ phân rã chức năng dưới đây.



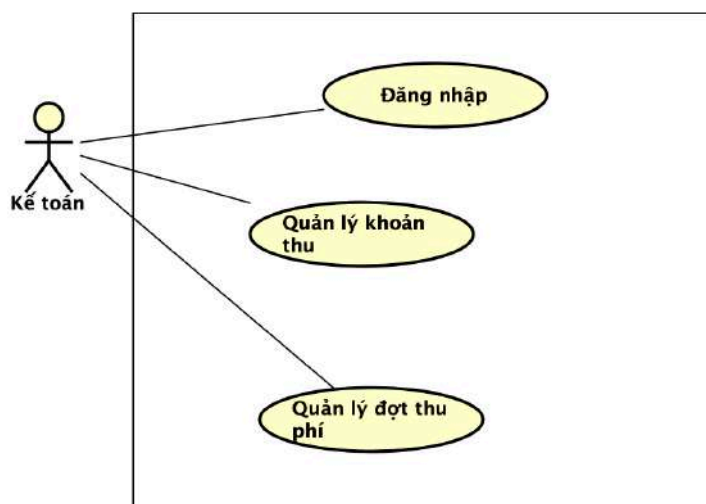
Hình 6-4: Phân rã chức năng nghiệp vụ quản lý thu phí và đóng góp

Bài 6.3 Xây dựng use case tổng quan và các các use case phân rã cho nghiệp vụ quản lý thu phí và đóng góp.

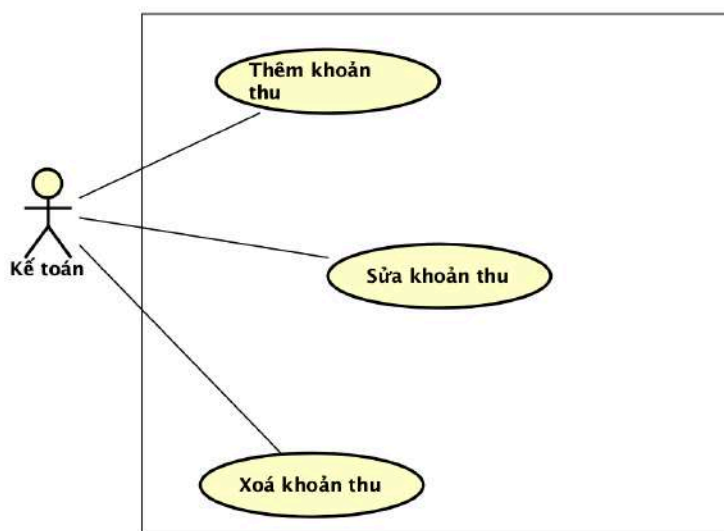
Kế toán có thể thực hiện các chức năng như sau:

- Đăng nhập
- Quản lý các khoản thu: Tạo/Sửa/Xoá khoản thu
- Quản lý đợt thu phí: Thêm thông tin nộp khoản thu các hộ gia đình, Thống kê, Truy vấn

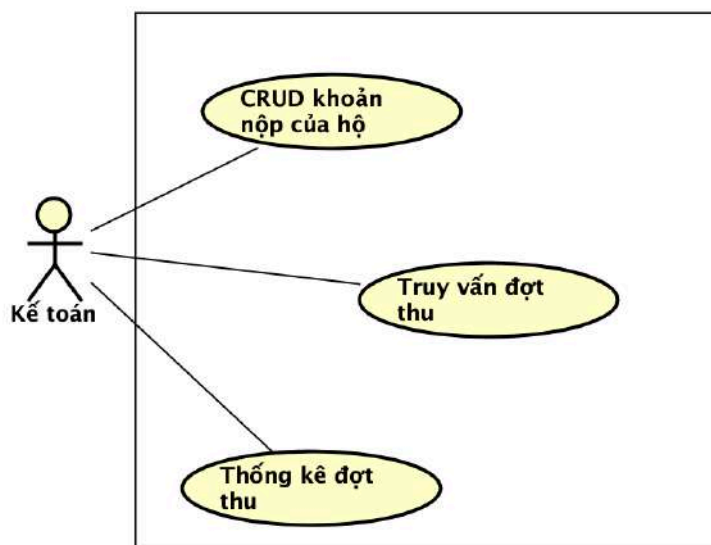
Hướng dẫn: Từ các đặc tả trên vẽ ra biểu đồ usecase tổng quan



Hình 6-5: Usecase tổng quan module quản lý thu phí
Thực hiện phân rã từng usecase: UC Quản lý khoản thu và UC Quản lý đợt thu phí



Hình 6-6: Biểu đồ UC quản lý khoản thu



Hình 6-7: Biểu đồ Usecase quản lý đợt thu phí

Bài 6.4 Xây dựng đặc tả chi tiết Usecase: Tạo khoản thu mới

Hướng dẫn: Cấu trúc cơ bản của bảng đặc tả chi tiết usecase:

Mã Use case	Tên Use case		
Tác nhân chính			
Tác nhân phụ			
Tiền điều kiện			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.		
	2.		
	3.		
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
Hậu điều kiện	Không		

Hình 6-8: Cấu trúc đặc tả chi tiết usecase

Chức năng tạo khoản thu mới có các bước thực hiện như sau:

1. Kế toán đăng nhập vào hệ thống và chọn chức năng tạo khoản thu mới
2. Hệ thống hiển thị màn hình Tạo khoản thu
3. Kế toán chọn loại hình khoản thu: Bắt buộc hoặc Đóng góp
4. Kế toán chọn khoản thu bắt buộc: Tiền vệ sinh
5. Kế toán truy vấn thông tin nhân khẩu của các hộ gia đình trong tổ dân phố
6. Hệ thống trả về danh sách thông tin nhân khẩu các hộ gia đình
7. Kế toán chọn cập nhật khoản thu bắt buộc cho các hộ gia đình
8. Hệ thống tính toán số tiền cần nộp cho từng hộ
9. Kế toán chọn hoàn thành tạo khoản thu
10. Hệ thống lưu khoản thu mới cho từng hộ gia đình.

Các luồng sự kiện thay thế xảy ra ở các bước trung gian:

- 4a. Kế toán chọn hình thức khoản thu Đóng góp

5a. Hệ thống hiển thị màn hình nhập thông tin khoản thu đóng góp

6a. Kế toán nhập thông tin: Tên khoản thu – Số tiền tối thiểu (mặc định: 0đ)

7a. Kế toán chọn hoàn thành nhập thông tin khoản thu

8a. Hệ thống hiển thị thông tin khoản thu mới

9a. Tiếp tục bước 5 trên luồng sự kiện chính

Xây dựng bảng đặc tả chi tiết UC Tạo khoản thu như sau:

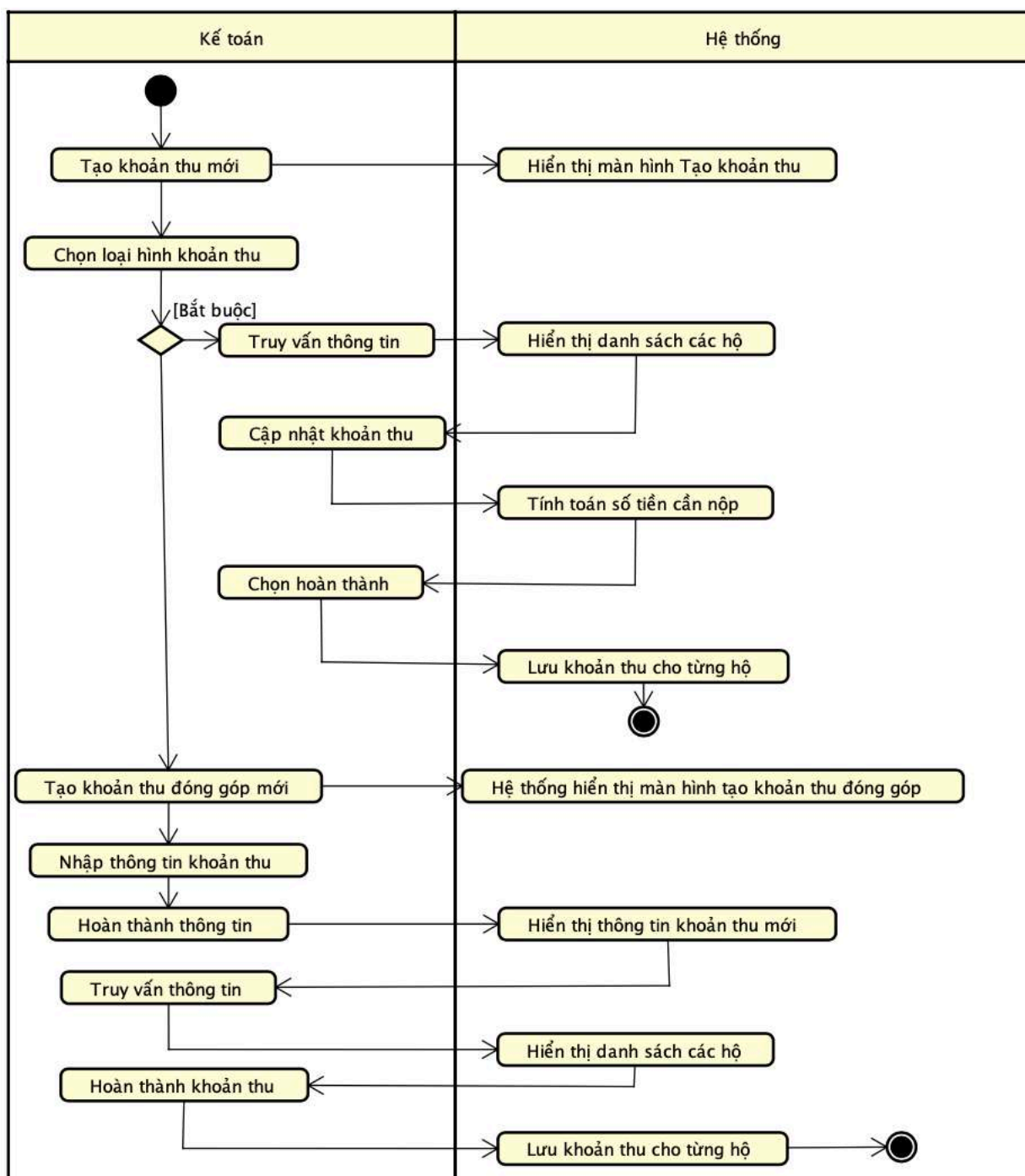
Bảng 6-1: Đặc tả chi tiết UC Tạo khoản thu

Mã Use case	UC002	Tên Use case	Tạo khoản thu
Tác nhân chính	Kế toán		
Tác nhân phụ	Không có		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Kế toán	chọn chức năng tạo khoản thu mới
	2.	Hệ thống	hiển thị màn hình tạo khoản thu
	3.	Kế toán	chọn loại hình khoản thu: Bắt buộc hoặc Đóng góp
	4.	Kế toán	chọn khoản thu bắt buộc là: Tiền vệ sinh
	5.	Kế toán	Truy vấn thông tin nhân khẩu các hộ gia đình trong tổ dân phố
	6.	Hệ thống	trả về danh sách các hộ gia đình
	7.	Kế toán	chọn cập nhật khoản thu cho các hộ gia đình
	8.	Hệ thống	Tính toán số tiền cần nộp cho từng hộ
	9.	Kế toán	chọn hoàn thành tạo khoản thu
	10.	Hệ thống	Lưu khoản thu mới cho từng hộ gia đình
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a.	Kế toán	Chọn hình thức khoản thu đóng góp
	5a.	Hệ thống	Hệ thống hiển thị màn hình nhập thông tin khoản thu đóng góp
	6a.	Kế toán	Nhập thông tin: Tên khoản thu – Số tiền tối thiểu (mặc định: 0đ)
	7a.	Kế toán	Chọn hoàn thành nhập thông tin khoản thu
	8a.	Hệ thống	Hiển thị thông tin khoản thu mới
	9a.	Kế toán	Truy vấn thông tin nhân khẩu các hộ gia đình trong tổ dân phố
	10a.	Hệ thống	trả về danh sách các hộ gia đình
	11a.	Kế toán	chọn hoàn thành tạo khoản thu
	12a.	Hệ thống	Lưu khoản thu mới cho từng hộ gia đình
Hậu điều kiện	Khoản thu mới được cập nhật cho mỗi hộ gia đình trong tổ dân phố		

Bài 6.5 Vẽ biểu đồ hoạt động trực quan hoá luồng hành động chi tiết của UC Tạo khoản thu.

Hướng dẫn:

- *Swimlanes:*
 - *Kế toán / Hệ thống*
- *Activities:*
 - *Xác định các hoạt động ứng với các bước tương ứng trong đặc tả chi tiết UC*
 - *Bước 1 – bước 10*
 - *Bước 4a – 12a*
- *Decision nodes:*
 - *Xác định các điểm quyết định tương ứng với các bước có điều kiện cần kiểm chứng*
- *Bước 3: Khoản thu là bắt buộc thì theo luồng sự kiện chính. Khoản thu là đóng góp thì theo luồng sự kiện thay thế*



Hình 6-9: Biểu đồ hoạt động UC Tạo khoản thu

CHƯƠNG 7

THIẾT KẾ PHẦN MỀM

Bài 7.1 Xây dựng mô hình lớp phân tích cho ca sử dụng "Thêm khoản thu" trong project "Phần mềm quản lý thu phí ở chung cư"

Hướng dẫn:

Qua các bài học trước chúng ta đã thấy được mối quan hệ gắn bó giữa hai giai đoạn phân tích và thiết kế. Đầu vào của thiết kế chính là các kết quả của quá trình phân tích yêu cầu, trong đó các yêu cầu phần mềm được biểu diễn qua các mô hình bao gồm các thành phần dựa trên kịch bản, các thành phần hành vi và các thành phần dựa trên lớp. Nhiệm vụ của nhà phát triển là phân tích sự tương tác của các lớp phân tích và xác định các phần tử của mô hình thiết kế để hình thành các lớp thiết kế chi tiết.

Trong đó các yếu tố của mô hình thiết kế bao gồm:

- Các yếu tố thiết kế dữ liệu: mô hình dữ liệu được thể hiện qua các mức trừu tượng khác nhau
- Các yếu tố thiết kế kiến trúc
- Các yếu tố thiết kế giao diện
- Các yếu tố thiết kế mức thành phần: Mô tả đầy đủ chi tiết bên trong của từng thành phần phần mềm
- Các yếu tố thiết kế mức triển khai: cách phân bổ chức năng phần mềm và các hệ thống con trong môi trường máy tính vật lý sẽ hỗ trợ thực thi phần mềm

Các bạn cần tự hoàn thiện trước bước phân tích yêu cầu phần mềm (Requirement Analysis) trước buổi học. Kết quả của bước phân tích yêu cầu phần mềm là tài liệu SRS (Software Requirement Specification) bao gồm các nội dung: biểu đồ use case tổng quan, biểu đồ use case phân rã nếu có, đặc tả các use case nghiệp vụ, từ điển thuật ngữ (glossary), đặc tả phụ trợ.

Dựa trên nội dung đặc tả của ca sử dụng "Thêm khoản thu" đã thực hiện trong các bài tập trước, trong usecase này tác nhân tham gia thực hiện là Kế toán, luồng nghiệp vụ của usecase khi thực hiện thành công sẽ thêm thông tin khoản thu mới vào cơ sở dữ liệu.

Các bước xây dựng mô hình lớp phân tích cho ca sử dụng "Thêm khoản thu" như sau:

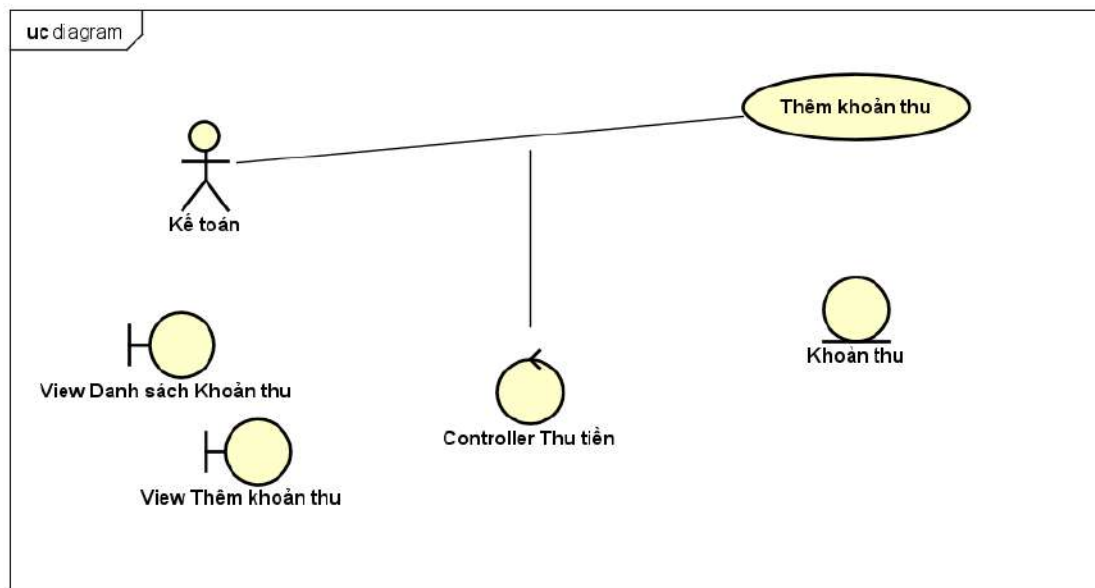
Bước 1. Phân tích lớp

Tìm các lớp (class) từ các hành vi trong use case.

Tìm lớp boundary: Lớp giao diện người dùng (user interface), trong ca sử dụng này chúng ta có Form danh sách khoản thu và Form thêm khoản thu

Tìm lớp entity: Khoản thu

Tìm lớp control: Control quản lý khoản thu



Hình 7-1: Phân rã usecase "Thêm khoản thu"

Kết quả thu được sau khi phân tích lớp được thể hiện trong hình 7.1. Đây là các lớp ban đầu được xác định từ ba khía cạnh của hệ thống có thể sẽ thay đổi:

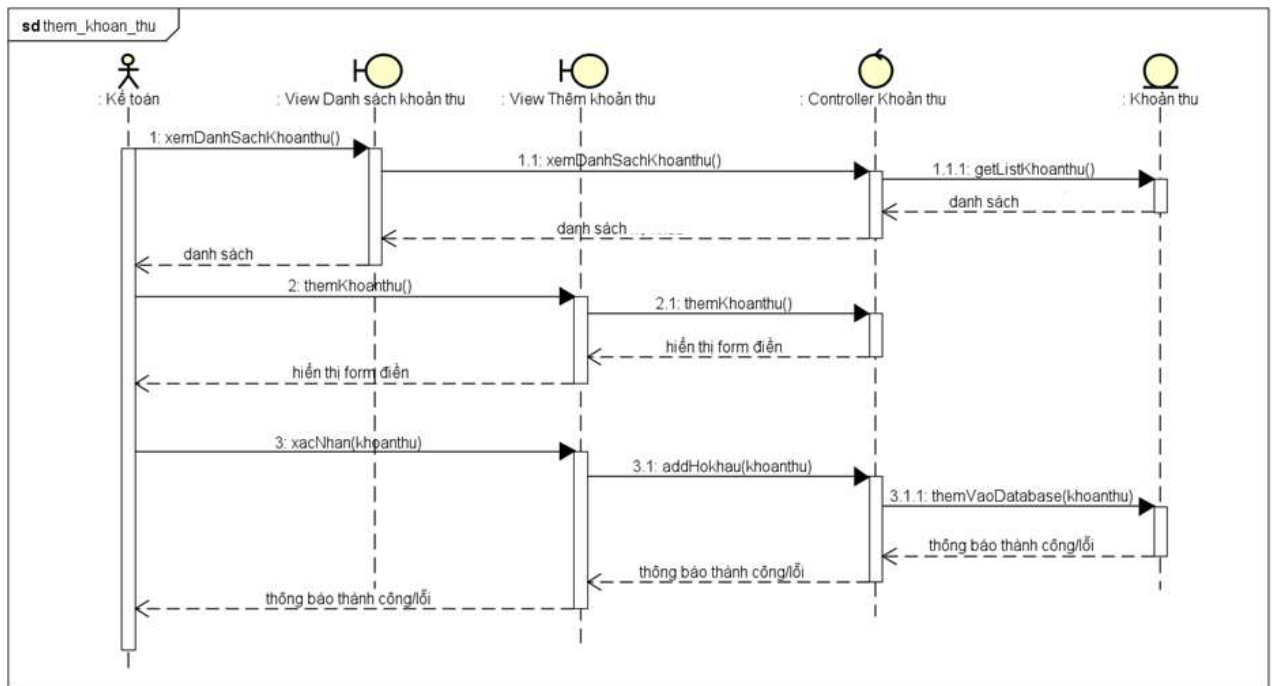
- Ranh giới giữa hệ thống và các tác nhân của nó
- Thông tin hệ thống sử dụng
- Logic điều khiển của hệ thống

Các lớp phân tích đại diện cho một mô hình khái niệm ban đầu cho “những thứ trong hệ thống có trách nhiệm và hành vi”. Các lớp phân tích xử lý các yêu cầu chức năng chủ yếu. Chúng mô hình hóa các đối tượng từ miền vấn đề. Các lớp phân tích có thể được sử dụng để đại diện cho "các đối tượng mà chúng ta muốn hệ thống hỗ trợ" mà không cần đưa ra quyết định về mức độ hỗ trợ của chúng bao nhiêu với phần cứng và bao nhiêu với phần mềm.

Bước 2. Phân phối hành vi trong use case tới các lớp

Phân bổ trách nhiệm tới các lớp và mô hình hóa mối quan hệ giữa các lớp bằng cách sử dụng biểu đồ tương tác (interaction diagram). Chúng ta có thể sử dụng biểu đồ trình tự (sequence diagram) **hoặc/và** biểu đồ giao tiếp (communication diagram).

Để xác định các lớp phân tích chịu trách nhiệm về hành vi được yêu cầu thực hiện bằng cách xem qua luồng sự kiện của ca sử dụng. Ở bước trước, chúng ta đã phác thảo một số lớp. Bây giờ là lúc để xem chính xác vị trí chúng được áp dụng trong luồng sự kiện ca sử dụng. Ngoài các lớp phân tích đã xác định, sơ đồ tương tác sẽ hiển thị các tương tác của hệ thống với các tác nhân của nó. Các tương tác nên bắt đầu với một tác nhân, vì một tác nhân luôn gọi ca sử dụng. Hình 7.2 minh họa Biểu đồ trình tự cho usecase "Thêm khoản thu".



Hình 7-2: Biểu đồ trình tự cho usecase "Thêm khoản thu"

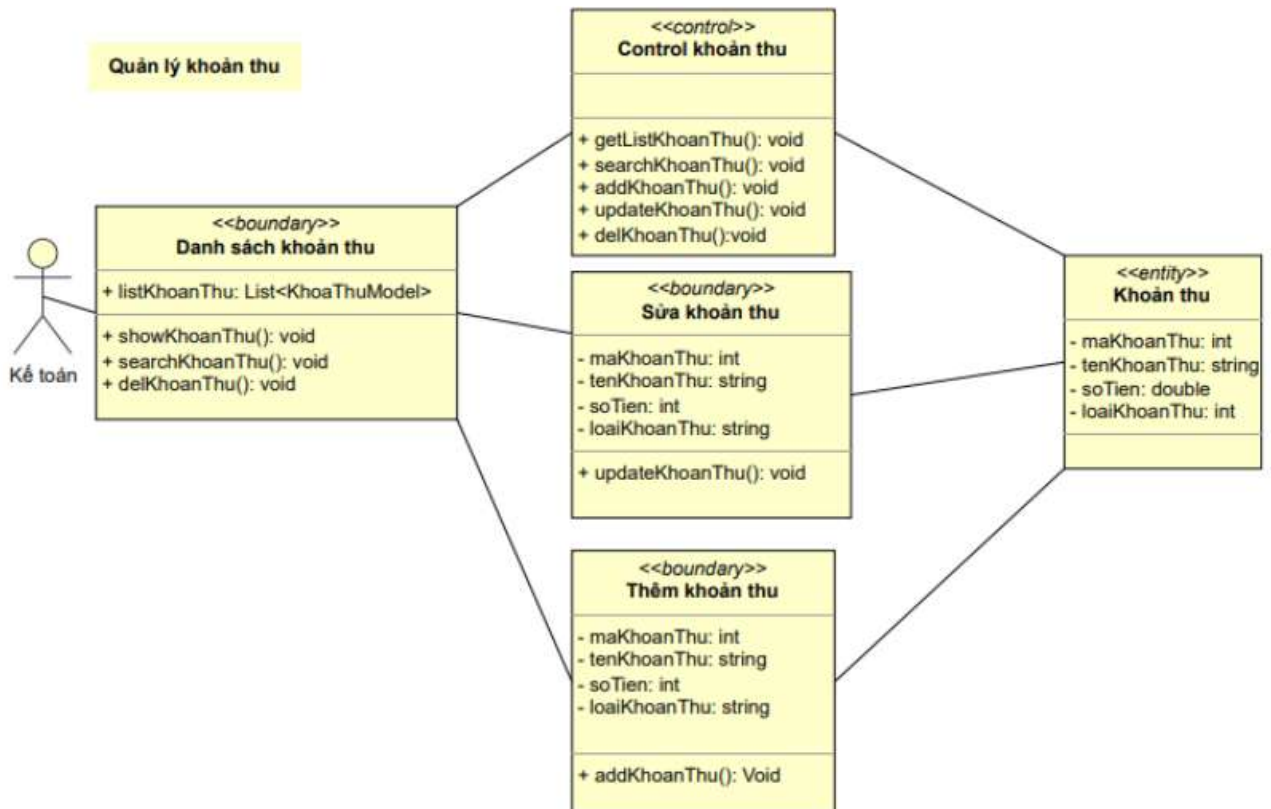
Bước 3. Xây dựng sơ đồ lớp phân tích

Để tìm mối quan hệ giữa các lớp phân tích, hãy bắt đầu nghiên cứu các liên kết trong sơ đồ Giao tiếp. Liên kết giữa các lớp chỉ ra rằng các đối tượng của hai lớp cần giao tiếp với nhau để thực hiện ca sử dụng.

Trước khi công việc thiết kế có thể được thực hiện, các lớp phân tích cần phải được lọc để đảm bảo rằng số lượng khái niệm mới tối thiểu đã được tạo ra. Một lớp có thể

tham gia vào nhiều ca sử dụng. Do đó, điều quan trọng là phải kiểm tra từng lớp để đảm bảo tính nhất quán trên toàn bộ hệ thống. Hợp nhất các lớp xác định những hành vi tương tự hoặc đại diện cho cùng một thực thể.

Kết quả chúng ta thu được sơ đồ lớp phân tích cho usecase tổng hợp "Quản lý khoản thu" bao gồm các lớp thực hiện usecase "Thêm khoản thu", "Sửa khoản thu" và "Danh sách khoản thu" trong hình 7.3.



Hình 7-3: Sơ đồ lớp phân tích cho usecase tổng hợp "Quản lý khoản thu"

Bài 7.2 Xây dựng mô hình lớp phân tích cho ca sử dụng "Quản lý thu phí" trong project "Phần mềm quản lý thu phí ở chung cư"

Hướng dẫn:

Bước 1. Phân tích lớp

Tìm các lớp (class) phân tích từ các hành vi trong ca sử dụng "Quản lý thu phí".

Tìm lớp boundary: Lớp giao diện người dùng (user interface), trong ca sử dụng này chúng ta có Form danh sách nộp phí

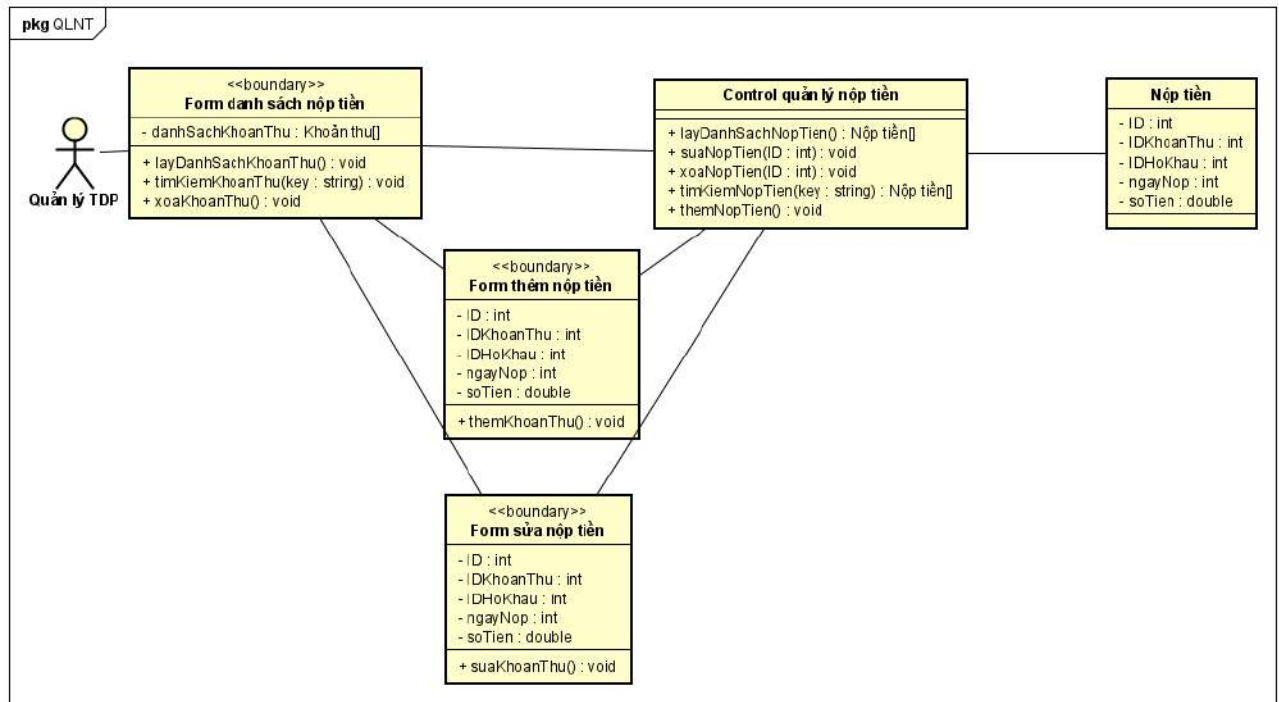
Tìm lớp entity: Khoản thu, Hộ khẩu, Nộp phí

Tìm lớp control: Control thu phí

Bước 2. Phân phối hành vi trong ca sử dụng tới các lớp

Phân bổ trách nhiệm tới các lớp và mô hình hóa mối quan hệ giữa các lớp thông qua phân tích luồng nghiệp vụ của ca sử dụng "Quản lý thu phí".

Bước 3. Xây dựng sơ đồ lớp phân tích



Hình 7-4: Sơ đồ lớp phân tích cho usecase tổng hợp "Quản lý thu phí"

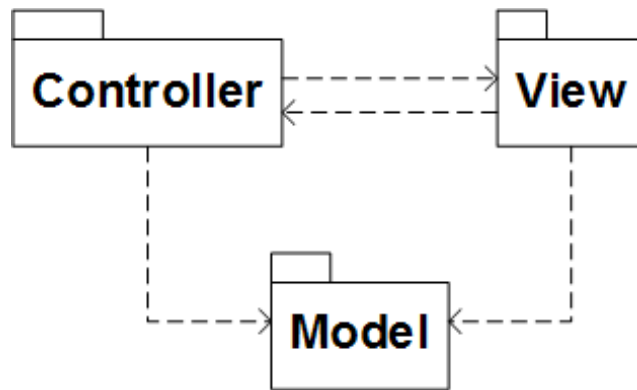
Bài 7.3 Thiết kế kiến trúc phần mềm

Hướng dẫn:

Khởi đầu của pha thiết kế là hoạt động xác định các yếu tố thiết kế. Các lớp phân tích được xác định trong Phân tích ca sử dụng thu được ở bước trước sẽ được tinh chỉnh thành các yếu tố thiết kế (ví dụ: các lớp hoặc hệ thống con). Các lớp phân tích chủ yếu xử lý các yêu cầu chức năng và các đối tượng mô hình từ miền "yêu cầu"; các phần tử thiết kế sẽ xử lý gồm cả các yêu cầu phi chức năng và các đối tượng mô hình từ miền "giải pháp". Trong đó, mô hình kiến trúc được bắt nguồn từ:

- (1) Thông tin về miền ứng dụng cho phần mềm sẽ được xây dựng;
- (2) Các thành phần của mô hình yêu cầu cụ thể như trường hợp sử dụng hoặc lớp phân tích, mối quan hệ và sự cộng tác của chúng đối với vấn đề hiện tại;
- (3) Sự sẵn có của các phong cách kiến trúc.

Trong bài tập này chúng ta xây dựng thiết kế theo kiến trúc MVC.

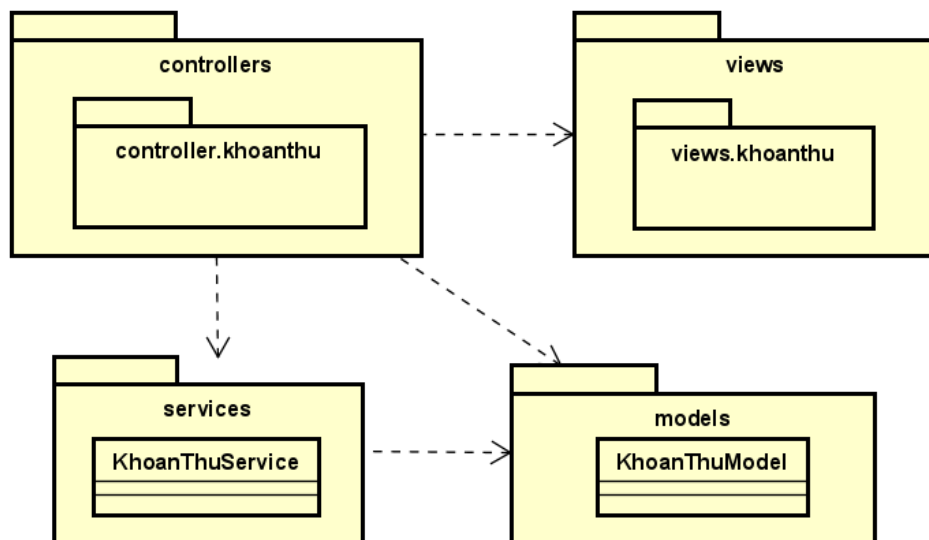


Hình 7-5: Mẫu kiến trúc MVC

Đặc điểm chính của kiến trúc MVC là cấu trúc liên kết "bộ ba" hoặc "tam giác":

- Khi có một sự kiện đầu vào, input view (đầu vào) sẽ liên hệ với bộ điều khiển.
- Bộ điều khiển thực thi đúng giao thức (thuật toán) để cập nhật mô hình.
- Các output view có liên quan được báo hiệu gián tiếp thông qua lệnh để truy vấn mô hình về kết quả được hiển thị.

Khi xác định các lớp, bạn nên nhóm chúng thành các gói, cho các mục đích quản lý cấu hình và tổ chức. Thông qua phân chia hay nhóm thành các gói, mô hình thiết kế có thể được cấu trúc thành các đơn vị nhỏ hơn để dễ hiểu hơn.



Hình 7-6: Biểu đồ gói cho usecase “Thêm khoản thu”

Theo các thành phần của mô hình kiến trúc MVC, chúng ta tổ chức các gói cho ca sử dụng "Thêm khoản thu" như hình 7.6, bao gồm:

- *Package models*: chứa các lớp mô hình hóa dữ liệu nghiệp vụ, trong usecase này là class KhoanThuModel
- *Package views*: chứa các package nhỏ hơn phân tách theo usecase. Package views.khoanthu gồm các file .fxml định nghĩa các giao diện người dùng
- *Package controllers*: chứa các package theo usecase với các lớp điều khiển để thực hiện luồng xử lý logic của usecase
- *Package services*: chứa các lớp xử lý các truy vấn dữ liệu với cơ sở dữ liệu

Bài 7.4 Thiết kế dữ liệu cho "Phần mềm quản lý thu phí ở chung cư"

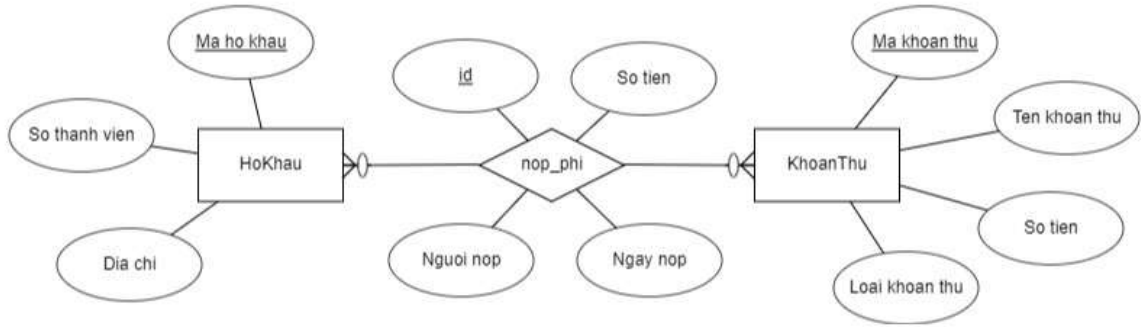
Hướng dẫn:

Bước 1. Mô hình dữ liệu mức khái niệm: biểu diễn bằng sơ đồ thực thể - quan hệ (ER diagram)

- Xác định các tập thực thể dữ liệu:
 - users: lưu trữ tài khoản đăng nhập
 - nhankhau: lưu trữ thông tin nhân khẩu của hệ thống
 - hokhau: lưu trữ thông tin về hộ khẩu
 - tamtrutamvang: lưu trữ thông tin tạm trú tạm vắng
 - khoanthu: lưu trữ thông tin các khoản thu
- Xác định các thuộc tính mô tả cho thực thể dữ liệu:
 - user: username, password, vai tro
 - nhankhau: id, hoten, ngaysinh, gioitinh, dantoc, tongiao, cccd, ngaycap, noicap, nghenghiiep, ghichu
 - hokhau: sohokhau, sonha, duong, phuong, quan, ngaylamhokhau
 - tamtrutamvang: id, trangthai (tạm trú hay tạm vắng), diachitamtrutamvang, thoigian, noidungdenghi
 - khoanthu: id, ngaytao, thoihan, tenkhoanthu, batbuoc, ghichu

- Xác định mối quan hệ liên kết giữa các thực thể dữ liệu
 - Quan hệ: Nhân khẩu là chủ hộ của Hộ khẩu:
 - Bội số quan hệ: một nhân khẩu chỉ có thể là chủ hộ của tối đa một hộ khẩu, và một hộ khẩu bắt buộc phải có một chủ hộ duy nhất. Hộ khẩu không được phép không có chủ hộ. Quan hệ 1-1.
 - Thuộc tính bổ sung: Không
 - Quan hệ: Hộ khẩu chứa Nhân khẩu
 - Bội số quan hệ: một Hộ khẩu có thể chứa nhiều nhân khẩu khác nhau nhưng bắt buộc phải chứa ít nhất một nhân khẩu, một nhân khẩu chỉ có thể hoặc không thuộc hộ khẩu nào hoặc nằm trong một hộ khẩu duy nhất. Quan hệ 1-n.
 - Thuộc tính bổ sung:
 - ngaythemnhankhau: lưu trữ ngày thêm nhân khẩu vào hộ khẩu
 - quanhevoichuho: lưu trữ quan hệ của nhân khẩu với chủ hộ khẩu
 - Quan hệ: Nhân khẩu thay đổi lịch sử Hộ khẩu
 - Giải thích: Lưu trữ thông tin về toàn bộ hoạt động thêm nhân khẩu vào hộ khẩu và xóa nhân khẩu khỏi hộ khẩu để tạo nên lịch sử thay đổi nhân khẩu của hộ khẩu.
 - Bội số quan hệ: Một nhân khẩu có thể thay đổi lịch sử (vào-ra hộ khẩu) của nhiều hộ khẩu khác nhau (tại những thời điểm khác nhau) và ngược lại, một hộ khẩu cũng có thể bị thay đổi lịch sử bởi nhiều nhân khẩu (không giới hạn số lượng nhân khẩu tối đa cho một hộ khẩu). Quan hệ n-n.
 - Thuộc tính bổ sung:
 - id: khóa chính, id của mỗi phiên thay đổi lịch sử

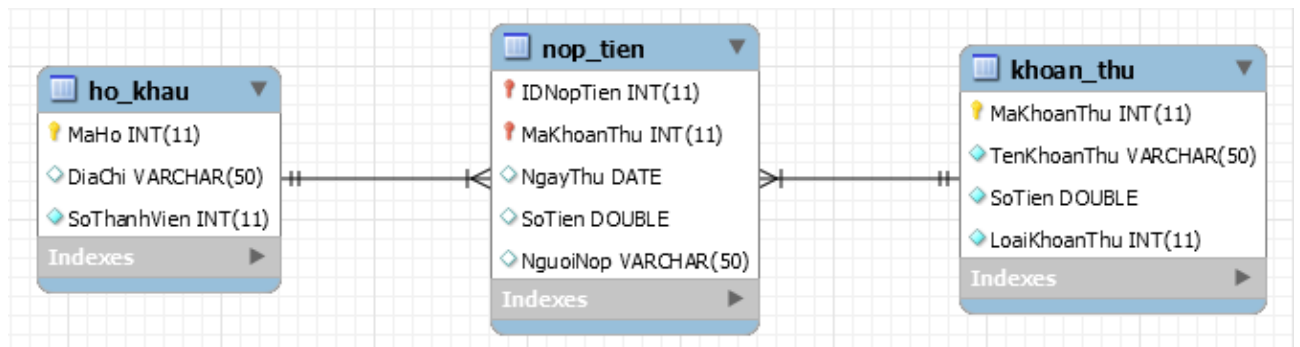
- loaithaydoi: loại thay đổi: 1 - thêm vào hộ khẩu, 2 - xóa ra hộ khẩu
- thoigian: thời gian thực hiện hành động thay đổi lịch sử
- Quan hệ: Nhân khẩu tham gia Tạm trú tạm vắng
 - Bội số quan hệ: Một nhân khẩu có thể tham gia nhiều hoạt động tạm trú, tạm vắng tại những thời điểm khác nhau. Một hoạt động tạm trú tạm vắng xác định duy nhất một nhân khẩu thực hiện hoạt động ấy. Quan hệ 1-n.
 - Thuộc tính bổ sung: Không
- Quan hệ: Nhân khẩu nộp phí cho Khoản thu
 - Giải thích: lưu trữ thông tin nộp tiền của từng hộ khẩu cho từng khoản thu.
 - Bội số quan hệ: Một hộ khẩu nộp tiền cho nhiều khoản thu và mỗi khoản thu bao gồm nhiều hộ nộp tiền. Quan hệ n-n.
 - Thuộc tính bổ sung:
 - id: Khóa chính, định danh cho mỗi phiên mà một hộ khẩu nộp tiền cho một khoản thu cụ thể.
 - nguoinop: tên người đại diện nhân khẩu nộp tiền
 - sotien: số tiền mà hộ khẩu đã nộp cho khoản thu
 - ngaynop: thời gian hộ khẩu nộp tiền cho khoản thu
- Xây dựng sơ đồ thực thể - quan hệ, hình 7.7 minh họa quan hệ giữa hộ khẩu và khoản thu



Hình 7-7: Sơ đồ thực thể - quan hệ

Bước 2. Mô hình dữ liệu mức logic: mô hình dữ liệu quan hệ

CSDL biểu diễn dưới dạng bảng (table), mỗi bảng gồm gồm các hàng (row / tuple) và các cột (column / field). Các thao tác thực hiện trên các hàng của bảng. Sự kết nối giữa các bảng được mô tả logic bằng các giá trị được lưu trữ trong một trường chung. Hình 7.8 minh họa lược đồ các bảng dữ liệu tương ứng với sơ đồ thực thể - quan hệ đã xây dựng ở bước trước.



Hình 7-8: Lược đồ các bảng dữ liệu

Bước 3. Xây dựng cơ sở dữ liệu và đặc tả cho các bảng dữ liệu

Đặc tả dữ liệu cho bảng ho_khai:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
<i>MaHo</i>	int		Khóa chính	Số nguyên dương	
<i>SoThanhVien</i>	int			Số nguyên dương	
<i>DiaChi</i>	varchar(200)	200 ký tự		Văn bản	

Bảng 7.1. Đặc tả dữ liệu bảng ho_khai

Đặc tả dữ liệu cho bảng khoan_thu:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
<i>MaKhoanThu</i>	int		Khóa chính		
<i>TenKhoanThu</i>	varchar(100)	100 ký tự		Văn bản	
<i>SoTien</i>	double			Số thực dương	
<i>LoaiKhoanThu</i>	int			0 hoặc 1	0 là bắt buộc 1 là tự nguyện

Bảng 7.2. Đặc tả dữ liệu bảng khoan_thu

Đặc tả dữ liệu cho bảng nop_tien:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
<i>IDNopTien</i>	int		-Khóa chính -Khóa tham chiếu từ bảng nhan_khau	Số nguyên dương	
<i>MaKhoanThu</i>	int		-Khóa chính -Khóa tham chiếu từ bảng khoan_thu	Số nguyên dương	
<i>NgayThu</i>	date				
<i>SoTien</i>	double				
<i>NguoiNop</i>	varchar(100)	100 ký tự			

Bảng 7.3. Đặc tả dữ liệu bảng nop_tien

Bài 7.5 Thiết kế chi tiết các lớp

Hướng dẫn:

1. Xác định các phương thức cho mỗi lớp

Bước đầu tiên trong xây dựng biểu đồ lớp chi tiết là xác định phương thức cho các lớp. Dựa trên các biểu đồ tương tác vừa xây dựng, quá trình xác định các phương thức được thực hiện theo các nguyên tắc sau:

- Xem xét các thông điệp trong các biểu đồ tương tác để xác định hành động tương ứng với thông điệp đó thuộc trách nhiệm của lớp nào.
- Các phương thức nào cần thiết để chuyển đổi các trạng thái trong biểu đồ trạng thái của một lớp.
- Xác định xem với mỗi lớp, lớp đó cần các hàm tạo và hàm hủy hay không.

Sau khi đã xác định đầy đủ các phương thức, công việc tiếp theo là phải xác định chi tiết giá trị trả về và các tham số liên quan với mỗi phương thức.

2. Xác định mối quan hệ giữa các lớp

Trong bước này, người phát triển hệ thống phải xác định đầy đủ mối quan hệ giữa các lớp và các vấn đề liên quan đến các mối quan hệ đó. Những công việc cụ thể phải thực hiện trong bước này là:

- Xác định cụ thể dạng của quan hệ giữa các lớp
- Xác định số lượng trong mỗi mối quan hệ

3. Xác định thuộc tính cho các lớp

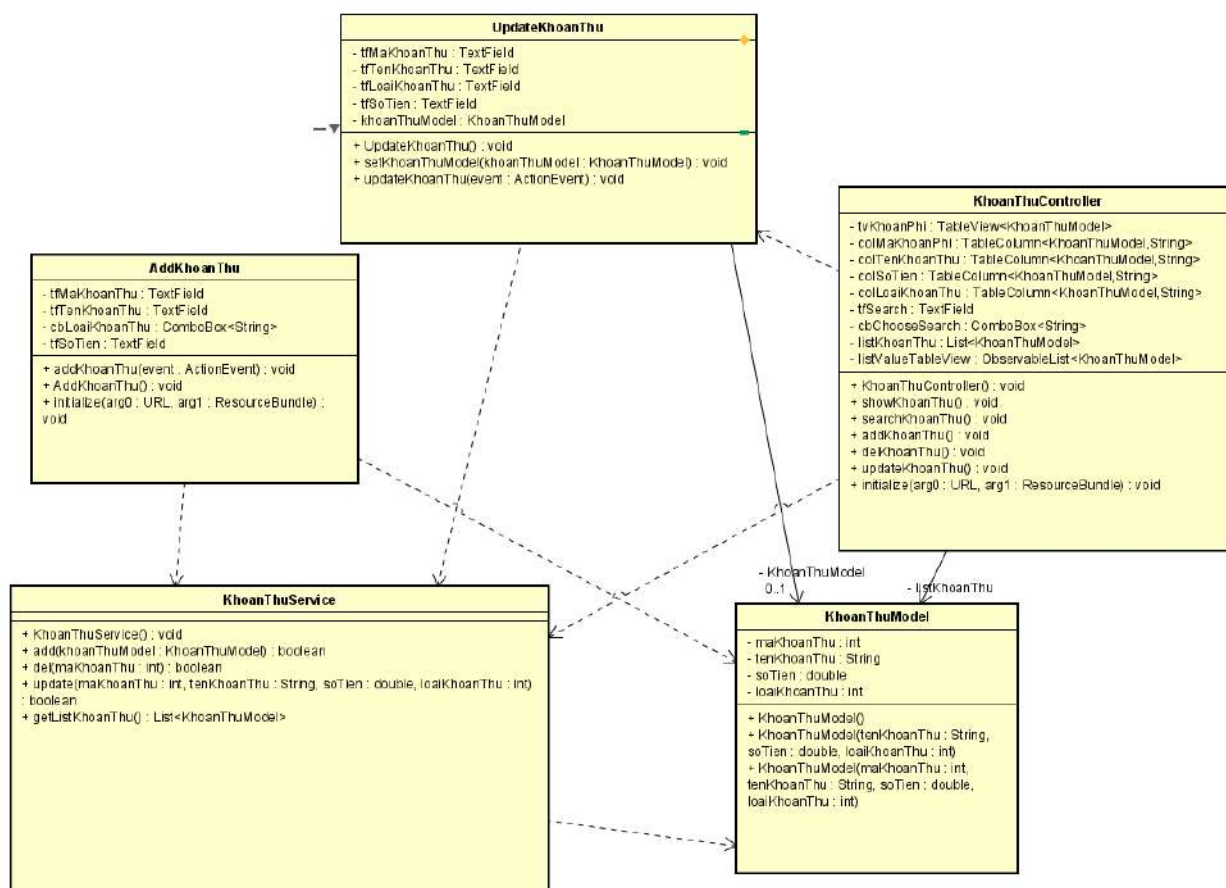
Bổ sung các thuộc tính cho các lớp, có thể tìm thấy qua phân tích trạng thái của đối tượng và quan hệ giữa các lớp.

4. Hoàn chỉnh sơ đồ lớp chi tiết

Trong bước này, người thiết kế phải thực hiện các công việc sau:

- Bổ sung các lớp còn thiếu.
 - Các lớp trung gian: giữa hai lớp có quan hệ m...n (nhiều thể hiện của lớp này tương ứng với nhiều thể hiện của lớp kia), người ta thường sinh ra thêm một lớp trung gian để chuyển quan hệ đó thành 2 quan hệ dạng 1..n.
 - Các lớp trừu tượng: trong một số trường hợp, một số lớp có thể có các thuộc tính chung hoặc phương thức chung. Khi đó, để tiện cho cài đặt, người thiết kế có thể bổ sung thêm các lớp trừu tượng
- Hiệu chỉnh mô tả thuộc tính và phương thức
- Kiểm thử tính đúng đắn của biểu đồ lớp, kiểm tra và xác định lỗi trong biểu đồ lớp: lỗi cú pháp và lỗi về ngữ nghĩa

Hình 7.9 minh họa sơ đồ lớp chi tiết cho ca sử dụng “Quản lý khoản thu”



Hình 7-9: Sơ đồ lớp chi tiết cho ca sử dụng “Quản lý khoản thu”

Bài 7.6 Thiết kế màn hình

Mẫu thử giao diện là một sản phẩm cho người dùng hình dung giao diện của ứng dụng sẽ như thế nào sau khi cài đặt xong, và góp ý để đội phát triển chỉnh sửa trước khi lập trình. Mẫu thử này càng đầy đủ, càng giống sản phẩm thật thì được coi là mẫu thử có độ tin cậy cao, ngược lại thì gọi là mẫu thử độ tin cậy thấp. Thiết kế màn hình là một trong các công việc thiết kế mẫu thử giao diện.

Dưới đây, chúng tôi hướng dẫn thiết kế màn hình bằng cách sử dụng công cụ Figma. Các bạn truy cập link <https://www.figma.com/> và đăng ký tài khoản theo hướng dẫn. Nếu các bạn có tài khoản email @xxx.edu.vn, các bạn sẽ có khả năng được sử dụng phiên bản Figma Pro theo diện Education plan: <https://www.figma.com/education/apply>

Việc thiết kế màn hình thực hiện theo trình tự sau:

- Thiết kế bố cục
- Thiết kế bản đơn sắc

- Thiết kế bản phôi màu.

Thiết kế bố cục

Đầu tiên chúng ta lựa chọn thiết kế bố cục. Mục đích ở đây là định hình bố cục trang màn hình theo kinh nghiệm cá nhân hoặc tham khảo các tỉ lệ thiết kế như đã đề cập trong bài đọc.

Các bạn cũng có thể sử dụng các template thiết kế miễn phí có sẵn, được cung cấp dưới dạng các thư viện của các công cụ hỗ trợ thiết kế như Figma.

Giao diện dành cho ứng dụng trên máy tính bàn (desktop application) có một số đặc trưng khác với giao diện web và giao diện cho thiết bị di động. Bố cục của ứng dụng trên máy tính bàn gắn liền với khái niệm màn hình và cửa sổ.

- Màn hình chỉ kích thước màn hình của người dùng, tương ứng với diện tích tối đa chứa thông tin mà ứng dụng có thể giao tiếp với người dùng tại một thời điểm. Nếu màn hình ứng dụng được thiết kế lớn hơn kích thước màn hình của người dùng thì người dùng phải thực hiện thêm các thao tác điều hướng thì mới có thể xem hết thông tin trên giao diện. Chính vì lý do đó, màn hình của ứng dụng desktop thường là hình chữ nhật có kích thước bằng hoặc nhỏ hơn kích thước màn hình người dùng.

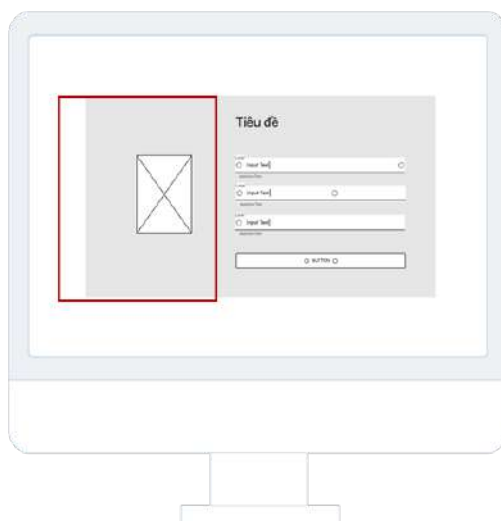
- Cửa sổ là một vùng trên màn hình, thường có hình chữ nhật, được xác định bởi một đường viền cho phép xem cụ thể một vùng trên màn hình máy tính hoặc thể hiện hộp thoại giữa người và máy tính. Cửa sổ có thể được di chuyển và hiển thị độc lập trên màn hình.

Một cửa sổ có thể có kích cỡ nhỏ, vừa đủ chứa một thông báo ngắn hoặc một nội dung một chiều đơn lẻ. Một cửa sổ cũng có thể có kích cỡ lớn, chiếm phần lớn hoặc toàn bộ không gian hiển thị có sẵn.

Như vậy, một màn hình có thể chứa một, hai hoặc nhiều cửa sổ trong ranh giới của nó. Vì các nội dung cần hiển thị của ứng dụng dài ngắn khác nhau, nên chúng ta chia thành 3 mẫu bố cục chính.

*Bố cục 1: cửa sổ 1120*645*

Mẫu bố cục thứ nhất sử dụng cửa sổ có kích thước 1120* 645, dùng cho giao diện của các chức năng đăng nhập, thêm khoản thu, thu tiền.

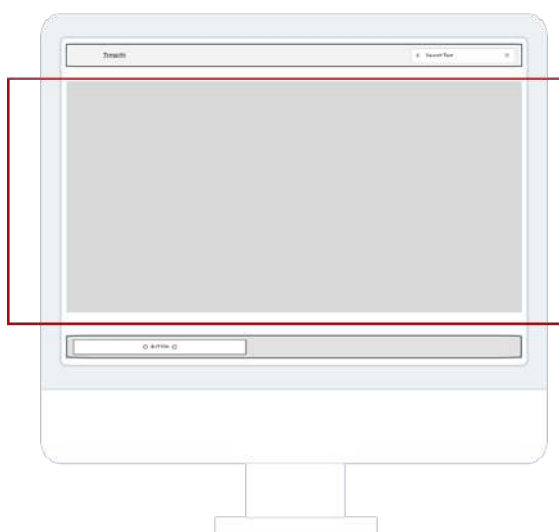


Hình 7-10: Bố cục 1 - Khung cửa sổ 1120 * 645

Kích thước này tương ứng với một hình chữ nhật có 1 cạnh là 1, 1 cạnh là căn bậc hai của hai (1:1.414), giống kích thước của 1 khổ giấy tiêu chuẩn.

*Bố cục 2: cửa sổ 1440 * 1024, đầu – thân – đuôi*

Mẫu bố cục thứ 2 sử dụng cửa sổ toàn màn hình, dùng cho trang chính, chức năng báo cáo thống kê. Trong bài này, chúng ta lựa chọn kích thước của một khung màn hình desktop điển hình: 1440 * 1024. Theo mẫu này, màn hình được chia làm 3 phần: phần đầu, phần thân và phần đuôi. Phần đầu và phần đuôi là vùng tương tác (vùng thông tin vào). Phần thân là vùng kết xuất (vùng thông tin ra).

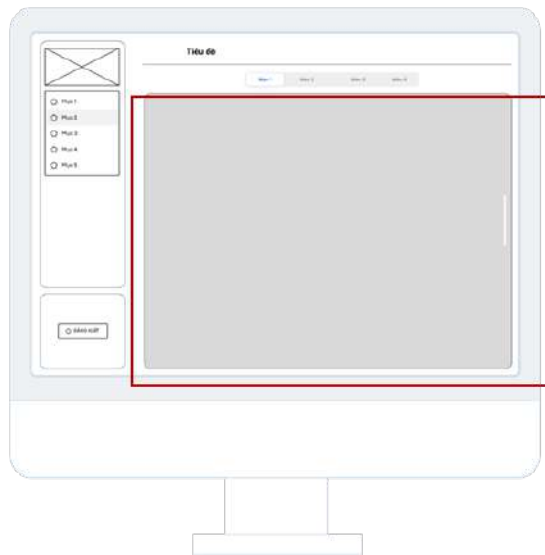


Hình 7-11: Bố cục 2 - Khung cửa sổ 1440 * 1024

*Bố cục 3: cửa sổ 1440 * 1024, trái – đầu – thân*

Để khách hàng có thêm lựa chọn, chúng ta thiết kế thêm 1 bố cục nữa sử dụng cửa sổ toàn màn hình. Theo mẫu này, màn hình cũng được chia thành 3 phần: phần bên trái, phần đầu và phần thân. Phần bên trái và phần đầu là vùng tương tác. Phần chính là phần kết xuất.

Lưu ý, lưu ý là mỗi mẫu thử chỉ dùng 1 lựa chọn bố cục cửa sổ toàn màn hình, nhất quán cho toàn bộ ứng dụng. Trong mỗi bố cục, các vùng kết xuất (vùng thông tin ra) và vùng tương tác (vùng thông tin vào) phải rõ ranh giới. Ví dụ, vùng khoanh đỏ là vùng kết xuất.



Hình 7-12: Bố cục 3 - khung cửa sổ 1440 * 1024

Thiết kế bản đơn sắc

Mục tiêu thiết kế bản đơn sắc là để kiểm tra tính hợp lý của bố cục khi người dùng tương tác với ứng dụng theo yêu cầu nghiệp vụ.

Màn hình đăng nhập

Các màn hình được thiết kế tương ứng với 2 yêu cầu nghiệp vụ khác nhau. Với mỗi màn hình, cần lập bảng đặc tả giao diện với cấu trúc như sau:

Bảng 7-1: Đặc tả màn hình giao diện đăng nhập - trường hợp người dùng có thể tự đăng ký tài khoản sử dụng phần mềm

Phần tử giao diện	Nội dung	Loại	Thuộc tính	Ghi chú
Tiêu đề màn hình	Nội dung một chiều, chứa tên màn hình	Text label	Hiển thị Kiểu chữ, kích thước chữ	
Ảnh	Nội dung một chiều, chứa ảnh đại diện / logo ứng dụng	Image	Hiển thị Kích thước ảnh	
Ô nhập 'Email'	Nội dung tương tác, cho phép người dùng nhập Email	Text label + text input	Hiển thị Kiểu chữ, kích thước chữ	Phải nhập email
Ô nhập 'Password'	Nội dung tương tác, cho phép người dùng nhập password	Text label + text input	Hiển thị Kiểu chữ, kích thước chữ	Phải nhập password
Nút 'Đăng nhập'	Nội dung tương tác, khi nhấn sẽ mở màn hình trang chủ nếu đăng nhập thành công	Button	Sự kiện click Hiển thị Kiểu chữ, kích thước chữ	
Liên kết 'Đăng ký'	Nội dung tương tác, khi nhấn sẽ mở màn hình đăng ký tài khoản	Link	Sự kiện click Hiển thị Kiểu chữ, kích thước chữ	
Liên kết 'Quên mật khẩu'	Nội dung tương tác, khi nhấn sẽ mở màn hình quên mật khẩu	Link	Sự kiện click Hiển thị Kiểu chữ, kích thước chữ	



Hình 7-13: Màn hình đăng nhập theo vai trò người dùng đăng ký

Trong Hình 7-13, người dùng có thể tự đăng ký tài khoản sử dụng phần mềm như trong đặc tả ở Bảng 7-1.

Bảng 7-2: Đặc tả giao diện màn hình đăng nhập với vai trò biết trước

Phần tử giao diện	Nội dung	Loại	Thuộc tính	Ghi chú
Tiêu đề màn hình	Nội dung một chiều, chứa tên màn hình	Text label	Hiển thị Kiểu chữ, kích thước chữ	
Ảnh	Nội dung một chiều, chứa ảnh đại diện / logo ứng dụng	Image	Hiển thị Kích thước ảnh	
Ô nhập 'Email'	Nội dung tương tác, cho phép người dùng nhập Email	Text label + text input	Hiển thị Kiểu chữ, kích thước chữ	Phải nhập email
Ô nhập 'Password'	Nội dung tương tác, cho phép người dùng nhập password	Text label + text input	Hiển thị Kiểu chữ, kích thước chữ	Phải nhập password
Nút 'Đăng nhập'	Nội dung tương tác, khi nhấn sẽ mở màn hình trang chủ nếu đăng nhập thành công	Button	Sự kiện click Hiển thị Kiểu chữ, kích thước chữ	
Ô nhập 'Vai trò'	Nội dung tương tác, cho phép người dùng lựa chọn vai trò đăng nhập hệ thống	Dropdown box	Sự kiện click Hiển thị	Nếu không chọn vai

Phần tử giao diện	Nội dung	Loại	Thuộc tính	Ghi chú
			Kiểu chữ, kích thước chữ	trò, ngầm định là vai nhân viên quản lý



Hình 7-14: Đăng nhập theo vai trò thiết lập sẵn

Trong Hình 7-14, người dùng chỉ có thể đăng nhập với vai trò xác lập sẵn, theo đặc tả trong Bảng 7-2.

Màn hình trang chính

Người học tự thực hiện đặc tả giao diện màn hình trang chính tương tự như trên.

Hình 7-16 minh họa màn hình trang chính theo bố cục đầu – thân – đuôi.

Theo bố cục 2, các nhóm thông tin cần biểu thị bao gồm:

- Menu giới thiệu các chức năng của ứng dụng, gồm 5 chức năng chính: Trang chủ, căn hộ, cư dân, khoản thu, thống kê, cài đặt.
- Một số thông tin tổng hợp cho người dùng, hiển thị dưới dạng số, như thông tin về căn hộ, cư dân, khoản thu, có dẫn link đến màn hình chi tiết. Các khoản thu mới cập nhật, biến động dân cư.
- Lịch sử thu phí hiển thị dưới dạng biểu đồ thanh.
- Cuối cùng là các nút tương tác nhanh như tạo khoản thu, thu phí, xuất báo cáo.



Hình 7-16: Màn hình trang chính theo bố cục 2



Hình 7-15: Màn hình trang chính theo bố cục 3

Hình 7-15 minh họa màn hình trang chính theo bố cục đầu – trái – thân.

- Menu chức năng ở bên trái, phía dưới là nút đăng xuất
- Góc trên bên trái có thể bố trí ảnh đại diện của người dùng hoặc logo ứng dụng.
- Trang chủ tổng hợp thông tin theo ngày, tuần, và tháng.
- Có vị trí chờ để hiển thị thông tin tổng hợp về căn hộ và khoản thu

- Có biểu đồ lịch sử thu phí và lịch sử biến động dân cư
- Các nút tương tác nhanh nằm trong phần khoản thu và biến động dân cư.



Hình 7-17: Màn hình tạo khoản thu theo bố cục 1

Có thể thấy, các nội dung hiển thị trên 2 màn hình nói trên được kết xuất khác nhau. Ví dụ: xét chức năng thêm khoản thu. Theo bố cục 2 ta tìm thấy chức năng này ở phần đuôi màn hình (vùng tương tác). Theo bố cục 3, chức năng này ở phần menu bên trái, ngoài ra còn có link truy cập nhanh trong khung tổng hợp thông tin về khoản thu.

Màn hình tạo khoản thu

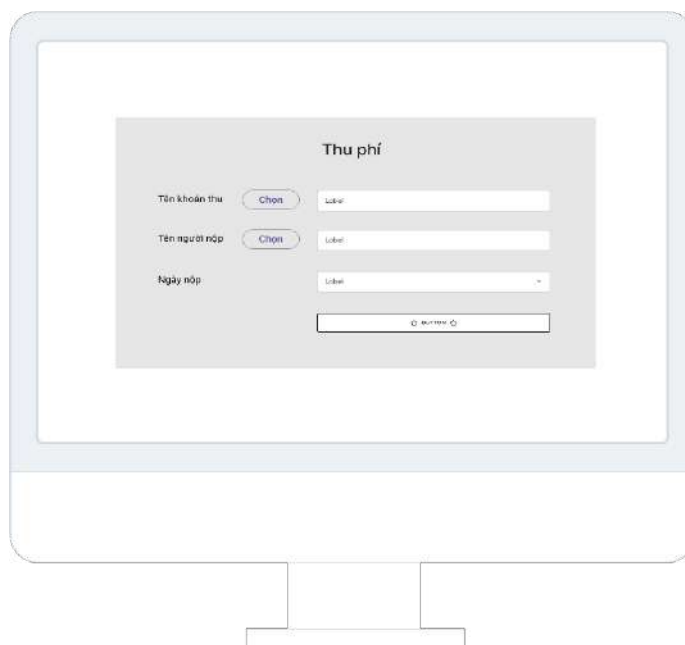
Màn hình tạo khoản thu theo bố cục 1, với 4 nội dung.

Màn hình thu phí

Màn hình thu phí chỉ cần 3 nội dung, cho phép người quản lý nhập thông tin tại thời điểm thực hiện thu tiền.

Màn hình thống kê

Với màn hình thống kê các khoản đóng góp, thông tin được hiển thị dưới dạng bảng. Vẫn tôn trọng bố cục như cũ.



Hình 7-19: Màn hình thu phí theo bố cục 1



Hình 7-18: Màn hình thống kê theo bố cục 2

Với bố cục 2, thông tin về trạng thái đã thanh toán và chưa thanh toán đặt trong cột trạng thái.

Với bố cục 3, tách riêng thông tin đã thanh toán và chưa thanh toán thành 2 nhóm.



Hình 7-20: Màn hình thống kê theo bố cục 3

Thiết kế bản phối màu

Với bản phối màu, đầu tiên chúng ta phải viết được bản hướng dẫn phong cách thiết kế giao diện. Đây là một bộ tiêu chuẩn để thiết kế nội dung và kết xuất nội dung, đảm bảo các phần tử giao diện được thiết kế nhất quán trong toàn bộ ứng dụng. Bản này quy ước thông tin chi tiết về kiểu chữ, dải màu, biểu tượng, các phần tử điều hướng lưới và bố cục như thanh trượt, các phần tử hỗ trợ tương tác như cảnh báo, thông báo lỗi, và mục đích sử dụng cho các phần tử giao diện đã liệt kê trong đặc tả giao diện người dùng.

Nên cung cấp cho người dùng tối thiểu 3 lựa chọn phối màu.

Bảng 7-3: Một số thông tin hướng dẫn thiết kế

Kiểu 1

Kiểu chữ: Inter, Balsamiq Sans

Dải màu: CAD6E2, FF7BAC, D9D9D9, 1B59F8

Biểu tượng: outlined

Bố cục và lưới: Bố cục 1 + 2

Phần tử giao diện: dropdown menu, item, vector, label, leading icon, image, text, line, table, head, frame, row, bounds, leg, body, screen.

Kiểu 2

Kiểu chữ: raleway

Dải màu: 6F6AF8, 7874F9, CFCFEF, F2F2FD

Biểu tượng: outlined

Bố cục và lưới: bố cục 1 + 3

Phần tử giao diện: dropdown menu, item, vector, label, leading icon, image, text, line, table, head, frame, row, bounds, leg, body, screen.

Kiểu 3

Kiểu chữ: raleway

Dải màu: 516089, 586995, 7874F9, E9B880, F1E4D2

Biểu tượng: drop shadow

Bố cục và lưới: Bố cục 1 + 2

Phần tử giao diện: dropdown menu, item, vector, label, leading icon, image, text, line, table, head, frame, row, bounds, leg, body, screen.

Màn hình đăng nhập



Hình 7-21: Các lựa chọn phối màu của màn hình đăng nhập

Màn hình trang chính



Hình 7-22: Các lựa chọn phối màu của màn hình trang chính

Màn hình tạo khoản thu



Hình 7-23: Các lựa chọn phối màu của màn hình tạo khoản thu

Màn hình thu phí



Hình 7-24: Các lựa chọn phối màu cho màn hình thu phí

Màn hình thống kê



Hình 7-25: Các lựa chọn phối màu của màn hình thống kê

Thiết kế bản phối màu trên Figma hay công cụ tương tự sẽ cho ra mẫu thử giao diện độ tin cậy cao.

Bài 7.7 Thiết kế điều hướng

Thiết kế điều hướng là quá trình tạo ra mẫu thử cho phép người dùng hiểu và thử tương tác với ứng dụng. Tương tự như với mẫu thử giao diện, ta có mẫu thử tương tác độ tin cậy cao nếu người dùng có thể thử thực hiện tương tác như sau này sử dụng trong thực tế, và độ tin cậy thấp nếu chỉ cung cấp manh mối về việc thực hiện tương tác sau này. Có 2 nội dung thiết kế điều hướng là điều hướng giữa các màn hình và điều hướng trong từng màn hình.

Các sự kiện điều hướng sẽ được gán cho từng phần tử giao diện cụ thể.

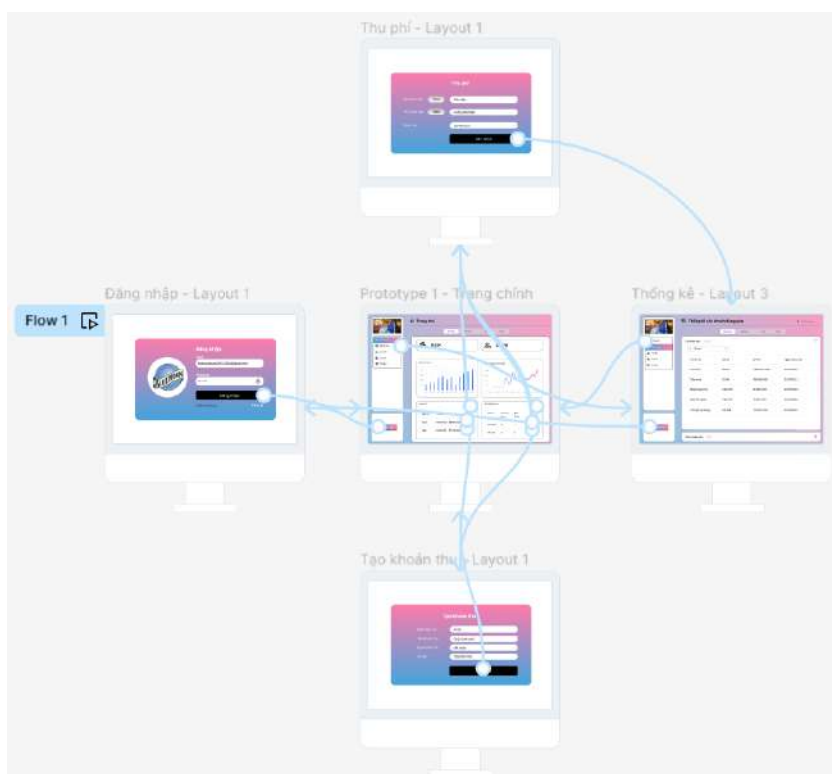
Dưới đây là minh họa điều hướng giữa các màn hình liên quan đến các chức năng đăng nhập, tạo khoản thu, thu phí, thông kê khoản thu theo 3 cách phối màu khác nhau.

Chi tiết về cách phối màu theo mẫu thứ nhất xem tại:

<https://www.figma.com/file/fhgXOgKRW7d04OxFE8iiX8/Prj07---Layout-01?type=design&node-id=0-1&mode=design&t=2J7rVDsnM0Fqdp3a-0>



Hình 7-26: Luồng duyệt tin theo cách phối màu thứ nhất



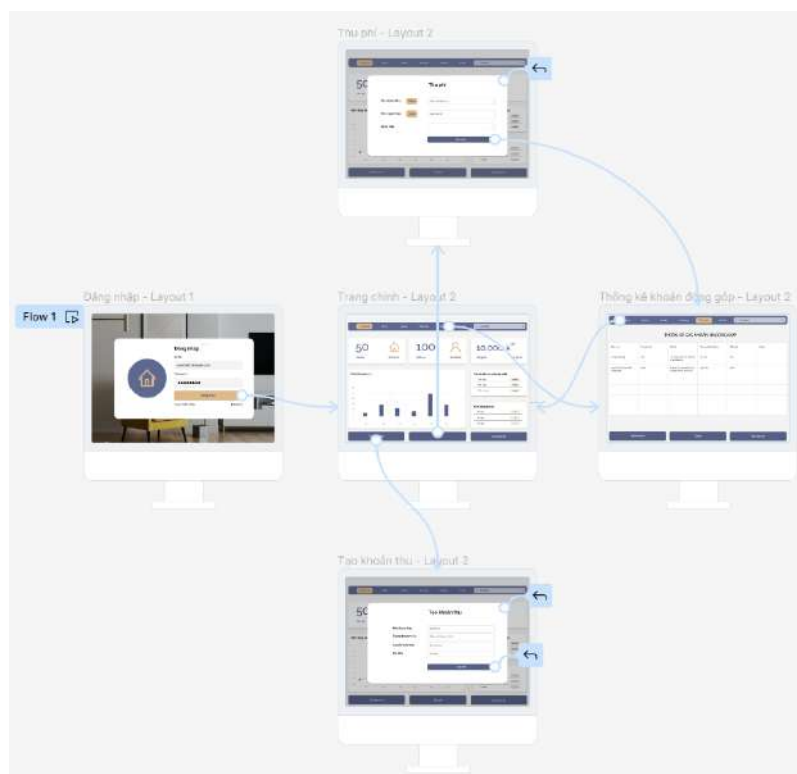
Hình 7-27: Luồng duyệt tin theo cách phối màu thứ hai

Chi tiết về cách phối màu theo mẫu thứ 2 xem tại:

<https://www.figma.com/proto/gm8lcdRDraVgXCUt14CKOo/Prj07---Layout-03---Color?node-id=1-2&starting-point-node-id=1%3A2>

Chi tiết về thiết kế theo mẫu phối màu số 3 xem tại:

<https://www.figma.com/file/nIHHUIkQboQaxI22HjA6wE/Prj07---Layout-04?type=design&node-id=0-1&mode=design&t=mKE68kCtGcoFPYTI-0>



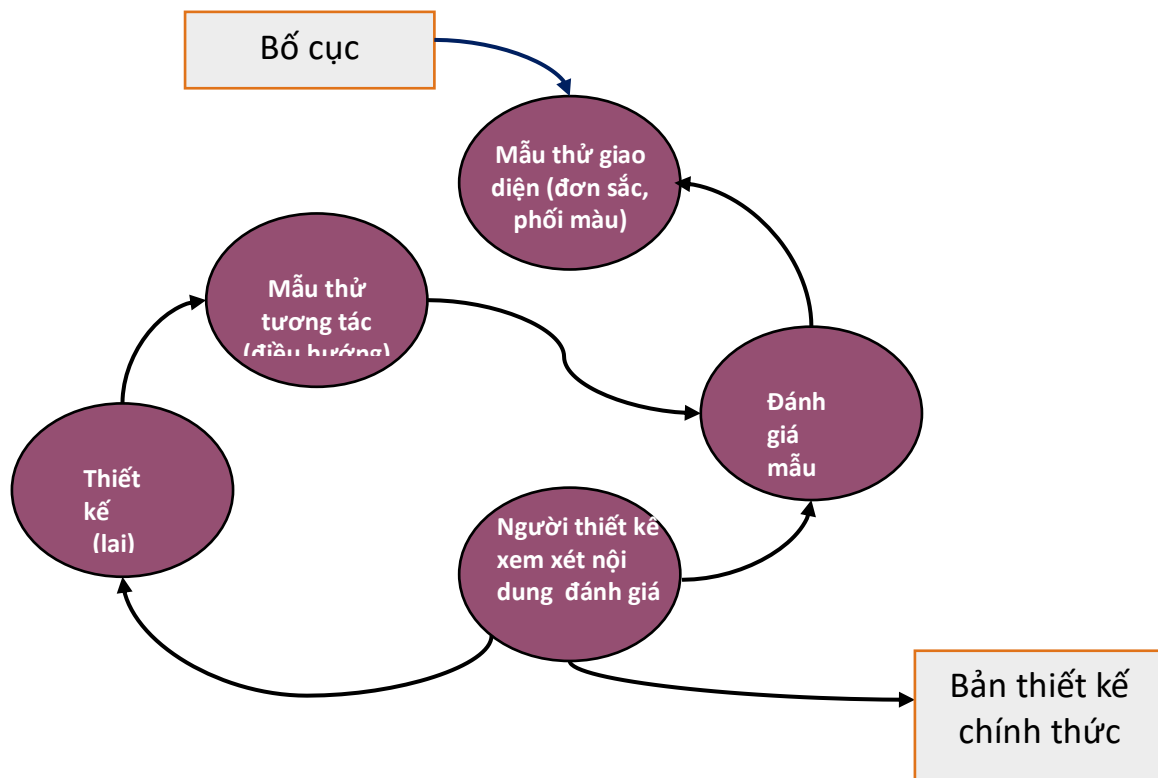
Hình 7-28: luồng duyệt tin theo cách phối màu thứ ba

Bài 7.8 Đánh giá mẫu thử

Việc đánh giá mẫu thử được thực hiện cùng với người dùng, theo quy trình minh họa ở Hình 7-29.

Các kỹ thuật đánh giá có thể sử dụng bao gồm:

- Thảo luận trực tiếp
- Câu hỏi điều tra để lấy phản hồi của người dùng.
- Quay video về việc sử dụng hệ thống rồi sau đó đánh giá nội dung.
- Cài các đoạn mã thu thập thông tin về các tiện ích được sử dụng và lỗi của người dùng.
- Sử dụng phần mềm có chức năng thu thập phản hồi trực tuyến của người dùng.



Hình 7-29: Các bước thực hiện đánh giá mẫu thử

Để đánh giá bản đơn sắc, phối màu, ta có thể đưa ra 2-3 mẫu thiết kế, dán lên tường, mời mọi người cùng xem và thảo luận:

- Đáp ứng yêu cầu khách hàng?
- Thông tin, chức năng dễ tìm?
- Bố cục gắn kết, thẩm mỹ?
- Thông tin quan trọng có dễ tìm với màn hình thực không?
- Giao diện có dễ đọc, dễ thực hiện với người dùng không?
- Giao diện có thể hiện ra tính cách riêng không?

Với mẫu thử độ tin cậy cao, ta có thể bước đầu đánh giá tính dùng được bằng cách trả lời các câu hỏi sau đây:

Bảng 7-4: Các thuộc tính về tính dùng được của mẫu thử

Thuộc tính	Mô tả
Khả năng học	Người dùng mới cần bao lâu để có thể hoạt động hiệu quả với hệ thống?
Tốc độ vận hành	Tốc độ phản ứng của hệ thống có đáp ứng tốt công việc của người dùng?

Chịu lỗi	Mức độ dung thứ lỗi của hệ thống đối với lỗi người dùng.
Khả năng khôi phục	Khả năng hệ thống khôi phục từ lỗi của người dùng.
Tương thích	Hệ thống gắn bó chặt chẽ với một kiểu làm việc đến đâu?

Như vậy, thông qua trao đổi với người dùng, chúng ta có thể phát hiện những vấn đề còn tồn tại trong mẫu thử và chỉnh sửa trước khi lập trình.

Các bạn có thể kiểm tra các phiên bản phối màu nói trên, xác định các vấn đề còn tồn tại và tự chỉnh sửa thêm.

CHƯƠNG 8 XÂY DỰNG PHẦN MỀM

Bài 8.1 Thiết kế biểu đồ lớp chi tiết cho project “Phần mềm quản lý thu phí ở chung cư”

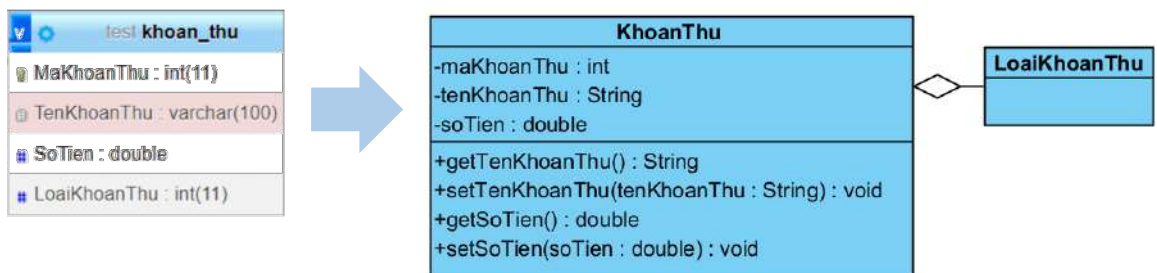
Hướng dẫn:

Dựa vào thiết kế kiến trúc hệ thống, tách các chức năng của mô hình vào các gói tương ứng. *Ví dụ:* KhoanThu, KhoanThuService, KhoanThuController...

Một số lưu ý khi thiết kế biểu đồ lớp chi tiết:

1. Đối với thiết kế các lớp model

- Lớp model là **trừu tượng hóa** của đối tượng trong thực tế
 - Ví dụ: Lớp *KhoanThu* không có các phương thức như *getConnection()*, *updateKhoanThuDb()* hay *showKhoanThuInfo()*
- Các phương thức của lớp cũng là các hành vi **trừu tượng**
 - Ví dụ: Không dùng *println(“Nhap so tien:”);* bên trong *setSoTien()*
- Lớp là kiểu của **một** đối tượng
 - Ví dụ: Lớp *KhoanThu* không có các phương thức như *addKhoanThu()* hay *showAllKhoanThu()*...
- Lớp model không nhất thiết phải giống hoàn toàn dữ liệu trong db
 - Ví dụ: Loại khoản thu



- *Các ví dụ khác:* Danh sách hộ phải nộp; Cán bộ phụ trách; Khoản thu bắt buộc hay tự nguyện...

2. Đối với thiết kế các lớp view

- Thường do các IDE hoặc các công cụ thiết kế giao diện sinh ra

- Hạn chế tối đa các xử lý (tính toán, kết nối CSDL,...) trên giao diện.
(Trừ trường hợp không ảnh hưởng đến dữ liệu)

3. Đối với thiết kế các lớp controller

- Một model có thể có nhiều controller (*xem lại khái niệm về móc nối và kết dính*)
- Ví dụ: Xây dựng các lớp Service hay Adapter

Bài 8.2 Hoàn thiện viết mã nguồn cho project “Phần mềm quản lý thu phí ở chung cư”, sau đó tiến hành xác định các code smell và tái cấu trúc lại mã nguồn (refactor)

Hướng dẫn:

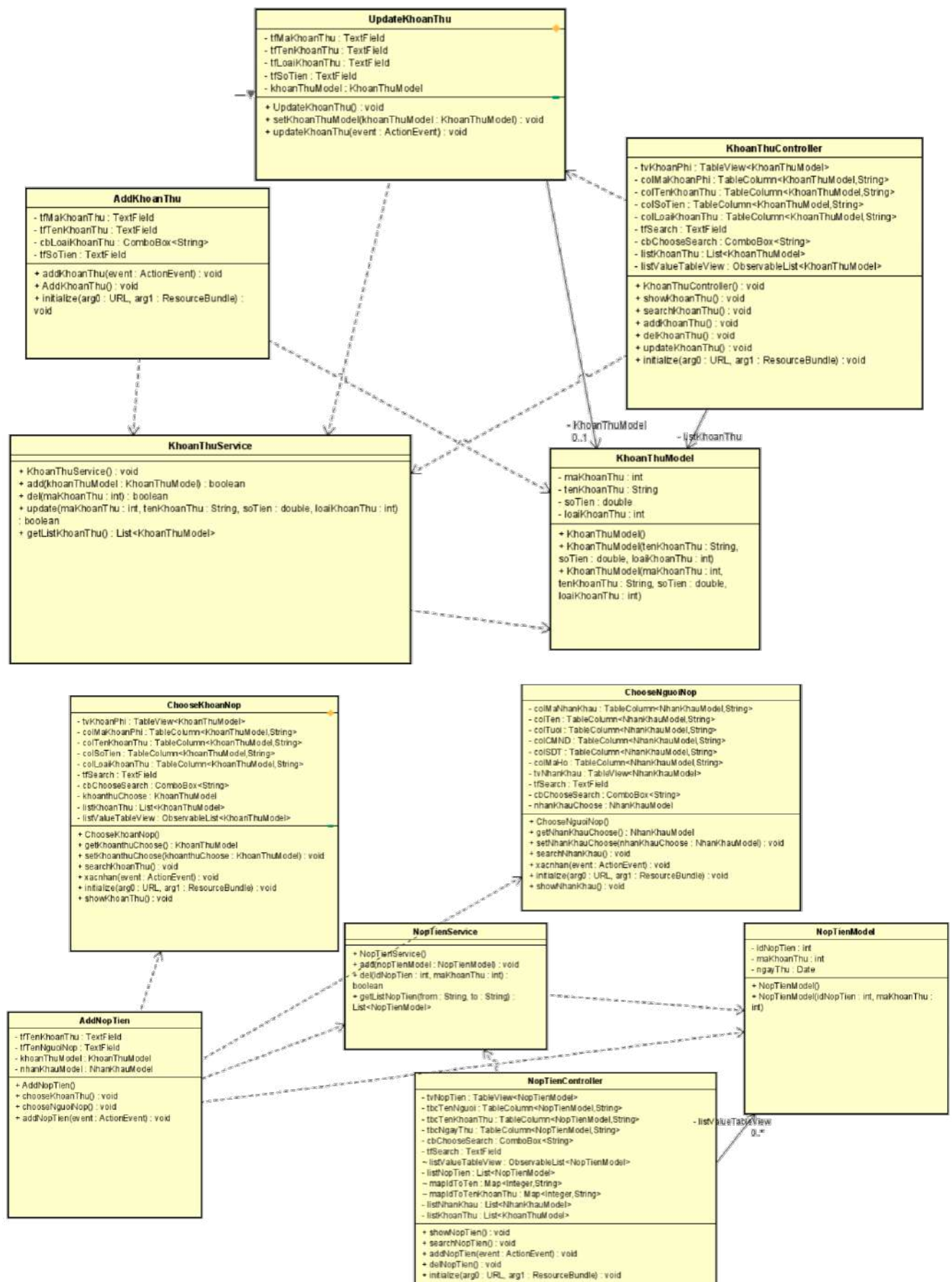
Người học cần thực hiện những việc sau trước tiến hành tái cấu trúc mã nguồn

- Viết mã nguồn cài đặt các chức năng của hệ thống quản lý
- Xây dựng các test case để đảm bảo các chức năng của hệ thống không bị thay đổi sau quá trình refactor

Khi tiến hành refactor, trước hết, cần xác định các “code smell” thông dụng do đó là những nơi dễ có mã nguồn xấu cần phải tái cấu trúc lại.

Xem xét ví dụ thiết kế trong Hình 8.1. Các code smell có thể có đối với thiết kế này

- *Shotgun Surgery*: Ví dụ khi các có thay đổi đối với khoản thu (chẳng hạn như thêm một thuộc tính), phải thay đổi ở cả Model, Controller và Service.
- *Duplicate Code*: Thiết kế “copy-pasta” nên nhiều khả năng các đoạn mã bị trùng lặp.
- *Data Clump*: Các dữ liệu (vd. mã khoản thu, khoản thu, loại khoản thu, số tiền) được truyền qua lại trong mã nguồn mà không được gói lại thành một đối tượng riêng biệt.
- ...



Hình 8-1: Thiết kế biểu đồ lớp cho các chức năng Khoản thu và Nợ tiền

Bài 8.3. Cài đặt mã nguồn cho project “Phần mềm quản lý thu phí ở chung cư” tuân thủ theo nguyên tắc viết mã (coding convention). Người học có thể tham khảo nguyên tắc viết mã của Java.

Hướng dẫn:

Nên xây dựng một bộ tài liệu hướng dẫn để thống nhất giữa các thành viên cùng nhóm hoặc để tham khảo trong quá trình phát triển phần mềm sau này (Đặc biệt nếu các nguyên tắc viết mã không tương đồng với nguyên tắc viết mã chuẩn của Java hoặc bổ sung thêm các nguyên tắc khác). Ngoài ra, người học cũng nên tuân theo nguyên tắc viết chú thích Javadocs do nguyên tắc này là nguyên tắc viết chú thích tiêu chuẩn, được tích hợp trong nhiều IDE và cũng là công cụ để tự động sinh tài liệu mã nguồn.

Đối với nguyên tắc viết mã Java

a) *Cách lề (Indentation)*

- Đơn vị cách lề: 4 khoảng trắng. Đối với tab là 8 khoảng trắng
- Dòng: Không nên quá 80 ký tự (documentation: không quá 70 ký tự). Trong trường hợp dòng quá 80 ký tự: Xuống dòng sau dấu phẩy, hoặc trước toán tử
- Dòng mới căn thẳng với bắt đầu của biểu thức
- Trường hợp dễ gây hiểu nhầm, có thể cách lề 8 khoảng trắng

Ví dụ

```
update(int maKhoanThu, String tenKhoanThu,
       double soTien, int loaiKhoanThu) {
    KhoanThuService().update(maKhoanThu, tenKhoanThu, soTien,
                              getLoaiKhoanThu(param01, param02, param 03
                                                param04, param05));

    String query = "UPDATE khoan_thu SET "
                  + "TenKhoanThu =" + "'" + tenKhoanThu + "'"
                  + "SoTien = " + soTien + ","
                  + "LoaiKhoanThu =" + loaiKhoanThu
                  + " WHERE MaKhoanThu = " + maKhoanThu;
```

b) *Khai báo và khởi tạo*

- Mỗi khai báo một dòng để tiện cho việc chú thích

Ví dụ

```
private int    maKhoanThu;    // Mã khoản thu
private String tenKhoanThu;  // Tên khoản thu
private int    soTien;       // Số tiền; Đơn vị: đồng
private int    loaiKhoanThu; // 1: Tự nguyện; 0: bắt buộc
```

- Khai báo ngay sau khi bắt đầu block mới (sau dấu "{")
- Nên khởi tạo ngay sau khi khai báo
- Đối với các block
 - Giữa tên phương thức và dấu ngoặc đơn "(" không có khoảng trắng
 - Mở ngoặc nhọn "{" cùng dòng với khai báo
 - Các phương thức cách nhau một dòng trống

c) Các biểu thức

- Không dùng ngoặc đơn "("..." khi return
- Luôn dùng ngoặc nhọn "{...}" với lệnh if
- Đối với cấu trúc if... else... else if...

```
if (condition) {
    statements;
} else if (condition) {
    statements;
} else {
    statements;
}
```

- Đối với cấu trúc switch

```
switch (condition) {
case ABC:
    statements;
    /* falls through */
case DEF:
    statements;
    break;
case XYZ:
    statements;
    break;
default:
    statements;
    break;
}
```

d) Khoảng trắng

- Cách 1 khoảng trắng sau từ khóa và trước ngoặc đơn

- `for (expr1; expr2; expr3)`
 - Cách 1 khoảng trắng sau dấu phẩy trong danh sách tham số
 - Cách 1 khoảng trắng trước và sau toán tử (trừ toán tử “.” và “++”, “--”)
 - Cách 1 khoảng trắng sau ép kiểu
- ```
myFunc((int) (cp + 5), ((int) (i + 3)) + 1);
```

e) Nguyên tắc đặt tên

- Tên lớp, giao diện: là danh từ, viết hoa chữ cái đầu tiên
- ```
class KhoanThuModel;
```
- Tên phương thức: là động từ, viết thường chữ cái đầu tiên
- ```
getMaKhoanThu();
```
- Tên biến: viết thường chữ cái đầu tiên

```
int maKhoanThu;
String tenKhoanThu;
```

- Hằng số: viết hoa tất cả các chữ cái, thay khoảng trắng bằng ký tự \_
- ```
final String ID_PREFIX = “KT_”;
```

Đối với chú thích mã nguồn

- Chú thích kết thúc dòng (End-Of-Line comment): Không nên dùng cho nhiều dòng (trừ trường hợp chú thích code)

```
if (foo > 1) {
    // Do a double-flip.
    ...
} else
    return false;          // Explain why
//if (bar > 1) {
//    // Do a triple-flip.
//    ...
//}
//else
//    return false;
```

- Chú thích nối tiếp (trailing comment): cùng dòng với đoạn mã cần chú thích, cách một khoảng đủ xa. Nếu có nhiều chú thích cuối dòng trong cùng một đoạn thì căn thẳng lề

```
if (a == 2) {  
    return TRUE;           /* special case */  
} else {  
    return isprime(a);     /* works only for odd a */  
}
```

- Chú thích 1 dòng: căn thẳng hàng với đoạn mã tiếp theo

Đối với chú thích Javadocs: viết chú thích dạng block được viết bằng HTML. Phải được viết ở trước khai báo lớp, trường, phương thức khởi tạo hoặc phương thức.

Chú thích Javadocs bao gồm hai phần

1. Phần mô tả
2. Phần tag. Ví dụ: @param, @see

Ví dụ Hình 8.2 minh họa một đoạn chú thích được viết bằng Javadocs

Chú thích được viết bằng Javadocs có thể dễ dàng sinh ra các tài liệu từ mã nguồn.

Để thực hiện việc này, có thể sử dụng các tiện ích trong IDE, hoặc chạy Javadocs từ dòng lệnh

```
javadoc -d C:\test com.test
```

Kết quả của chú thích tài liệu có thể được xem ở Hình 8.3.

```

/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute <a href="#{@link}">{@link URL}</a>.
 * The name argument is a specifier that is relative to the url argument.
 * <p>
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param url an absolute URL giving the base location of the image
 * @param name the location of the image, relative to the url argument
 * @return the image at the specified URL
 * @see Image
 */
public Image getImage(URL url, String name) {
    ...
}

```

Hình 8-2: Minh họa một đoạn chú thích bằng Javadocs

getImage

```
public Image getImage(URL url,
                     String name)
```

Returns an Image object that can then be painted on the screen. The `url` argument must specify an absolute URL. The `name` argument is a specifier that is relative to the `url` argument.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

Parameters:

`url` - an absolute URL giving the base location of the image.

`name` - the location of the image, relative to the `url` argument.

Returns:

the image at the specified URL.

See Also:

Image

Hình 8-3: Tài liệu được sinh ra từ chú thích tài liệu Javadocs

Bài 8.4. Sử dụng Null Object để thay thế đoạn mã nguồn sau

```
Customer c = findCustomer(...);
...
    if (customer == null) {
        name = "occupant";
    }
    else {
        name = customer.getName();
    }
if (customer == null) {
...

```

Hướng dẫn:

Có rất nhiều mã nguồn kiểm tra giá trị null. Do đó, có thể thay thế giá trị null bằng một đối tượng NullCustomer

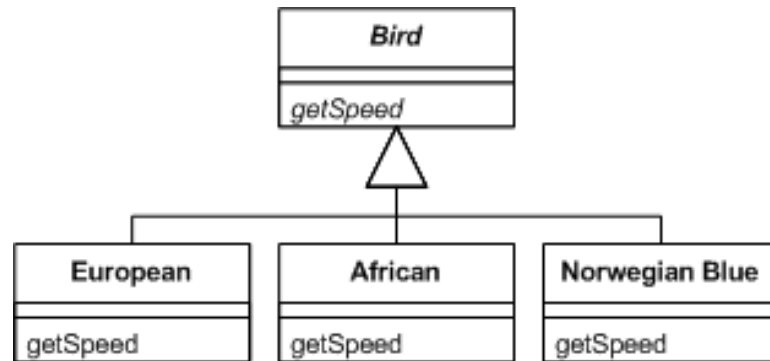
```
public class NullCustomer extends Customer {
    public String getName() {
        return "occupant";
    }
}
...
Customer c = findCustomer()
name = c.getName()
```

Bài 8.5. Tái cấu trúc đoạn mã nguồn sau bằng cách thay thế các kiểm thử điều kiện bằng đa hình.

```
double getSpeed() {
    switch (_type) {
        case EUROPEAN:
            return getBaseSpeed();
        case AFRICAN:
            return getBaseSpeed() -
                getLoadFactor()*numberOfCoconuts;
        case NORWEGIAN_BLUE:
            return (_isNailed) ? 0 :
                getBaseSpeed(_voltage);
    }
    throw new RuntimeException ("Should be unreachable");
}
```

Hướng dẫn:

Di chuyển từng nhánh của điều kiện sang một phương thức ghi đè trong mỗi lớp con. Thay đổi phương thức gốc thành phương thức trừu tượng.

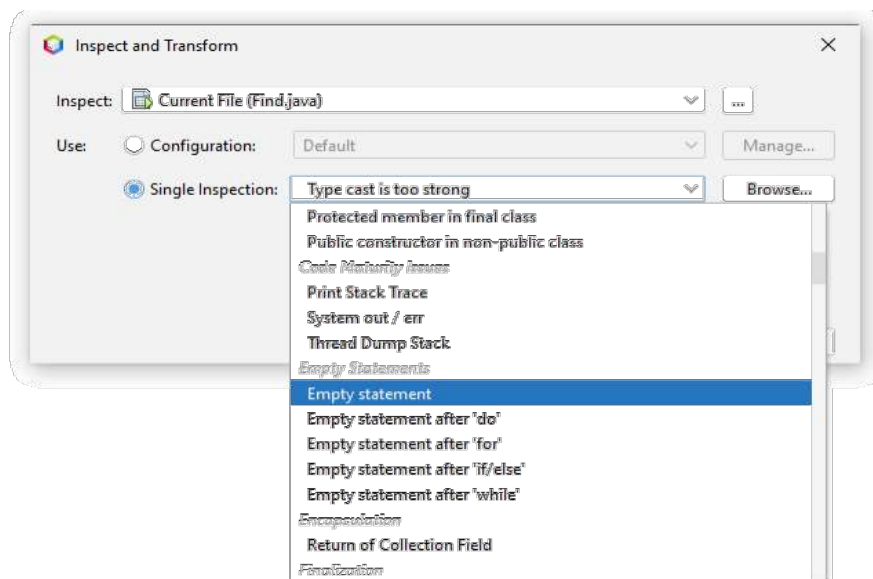


Hình 8-4: Biểu đồ lớp sau khi refactor

Bài 8.6. Tìm hiểu các công cụ hỗ trợ refactor của IDE người học đang sử dụng

Hướng dẫn:

Ví dụ trong NetBeans: Nhấp phải vào một file bất kỳ, sau đó chọn *Refactor* | *Inspect and Transform*. Xem Hình 8.5



Hình 8-5 : Bộ công cụ Inspect and Transform trong NetBeans

CHƯƠNG 9

ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

Luồng nghiệp vụ cần xử lý:

1. Đăng nhập
2. Tạo khoản thu
3. Thu phí
4. Thống kê các khoản đóng góp

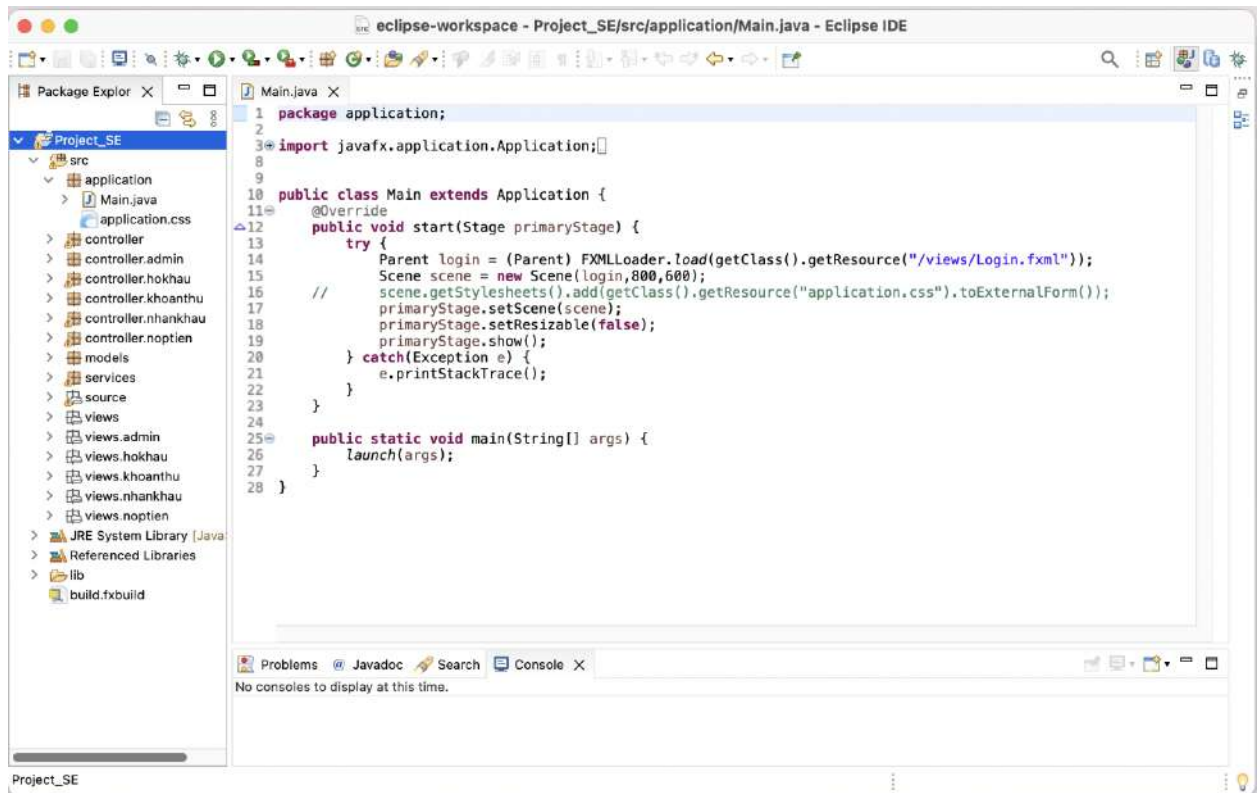
Trong bài học này chúng ta cùng thực hiện thao tác kiểm thử đối với chương trình quản lý thu phí và đóng góp của các hộ dân trong khu chung cư. Luồng nghiệp vụ xử lý chính của chương trình mà ta cần quan tâm được thực hiện theo các chức năng sau:

1. Đăng nhập
2. Tạo khoản thu
3. Nộp tiền với khoản thu đó
4. Thực hiện việc thống kê báo biểu

Trước khi thực hiện phần này chúng ta cũng cần nhắc lại một số điểm cần lưu ý như sau trong các kĩ thuật kiểm thử thì việc thực hiện kiểm thử hộp trắng phụ thuộc vào yếu tố cấu trúc của chương trình, tức là cách thực hiện viết code như thế nào và việc kiểm thử hộp đen tùy thuộc vào mô tả đầu vào và đầu ra của chương trình.

Trong phần bài tập này, chúng ta giả thiết rằng không có một yêu cầu cụ thể nào về chi tiết về dữ liệu và cũng không có một mô tả hoặc yêu cầu đầy đủ về mặt chức năng nào. Do đó, việc viết code tùy thuộc hoàn toàn vào người lập trình. Phần minh họa kiểm thử trong nội dung bài học này nhằm mục đích cung cấp cách thực hiện kiểm thử đối với một số nội dung cụ thể. Sau đây là phần thực hiện cho bài tập.

Phần chương trình đối với bài toán thu phí, đóng góp được xây dựng bằng ngôn ngữ Java, theo mô hình MVC, sử dụng JavaFX để phục vụ cho việc thiết kế giao diện và lập trình. Hình bên dưới thể hiện cấu trúc của chương trình trên bộ công cụ lập trình Eclipse.

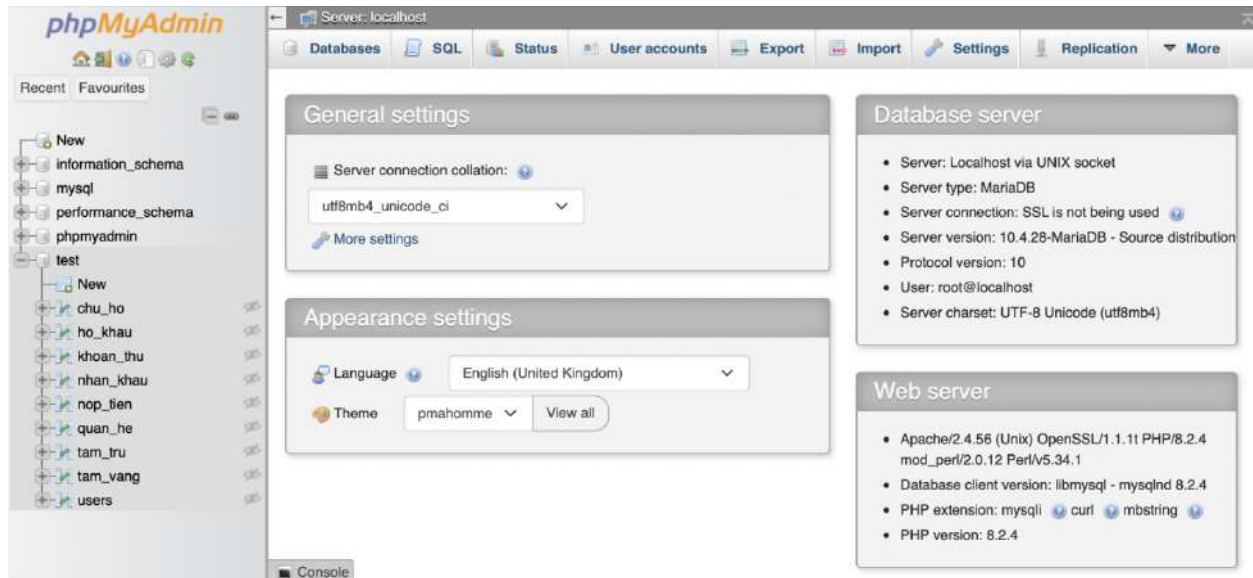


Hình 9-1: Cấu trúc của chương trình

Theo đó, nhiệm vụ của các package là:

- Phần **application** sử dụng cho phần khởi động chương trình
- Phần **controller** thực hiện phần xử lý nghiệp vụ
- Phần **models** phục vụ cho phần dữ liệu
- Phần **views** dành cho phần giao tiếp

Phần dữ liệu được thiết kế và cài đặt trên hệ quản trị CSDL mySQL. Hình bên dưới mô tả là phần cơ sở dữ liệu đã được thiết kế trên hệ quản trị CSDL này.



Hình 9-2: Cơ sở dữ liệu trên phpMyAdmin

Về mặt yêu cầu tổng thể của bài tập thì phần nghiệp vụ sẽ thực hiện một số thao tác liên quan đến quản lý thông tin về nhân khẩu và hộ khẩu. Trong đó, quản lý thu phí, đóng góp là một chức năng của hệ thống này. Các dữ liệu được thiết kế và bố trí trong các bảng **users**, **chu_ho**, **ho_khau**, **khoan_thu**, **nhan_khau**, **nop_tien**, **quan_he**, **tam_tru**, **tam_vang**.

Theo luồng nghiệp vụ được mô tả ở trên và trong phạm vi hướng dẫn về mặt chức năng của phần bài tập này, chúng ta chủ yếu hướng đến phần quản lý thu phí và đóng góp theo luồng xử lý nghiệp vụ đã được đề cập. Do đó, về mặt thông tin và dữ liệu, chúng ta chỉ cần quan tâm đến các bảng liên quan như sau: **users**, **chu_ho**, **ho_khau**, **khoan_thu**, **nhan_khau**, **nop_tien** còn một số thông tin quản lý khác sẽ không được đề cập đến.

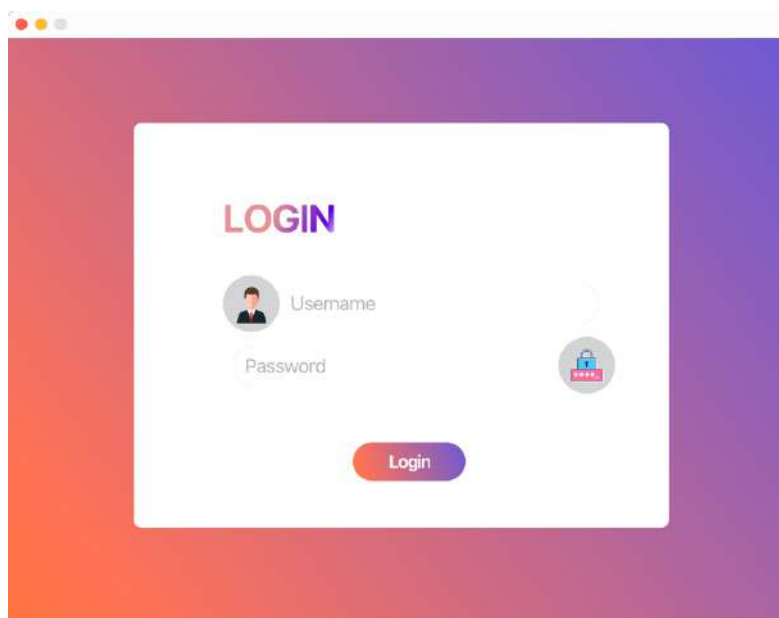
Đối với phần kiểm thử, chúng ta chỉ xem xét phần xử lý nghiệp vụ tương ứng theo luồng đã đề cập ở phần đầu, và như vậy theo đó chúng ta chỉ quan tâm đến tầng điều khiển ở mô hình MVC (tương ứng với các lớp trong package controller). Theo đó các chức năng kiểm thử bao gồm.

1. Bài tập 1. Kiểm thử chức năng đăng nhập
2. Bài tập 2. Kiểm thử chức năng tạo khoản thu
3. Bài tập 3. Kiểm thử chức năng nộp tiền cho khoản thu

Bài 9.1 Kiểm thử chức năng đăng nhập

Mô tả chức năng

Giao diện của chức năng được thể hiện trong hình dưới.



Hình 9-3: Màn hình chức năng login

Kiểm thử hộp trắng chức năng đăng nhập

Chức năng này được xử lý trong class LoginController và phương thức thực hiện là Login. Đoạn code tương ứng cho chức năng này như sau:

```

43 public void Login(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
44     String name = tfUsername.getText();
45     String pass = tfPassword.getText();
46     Connection connection = MySqlConnection.getMysqlConnection();
47     String query = "select * from users";
48     Statement st = connection.createStatement();
49     ResultSet set = st.executeQuery(query);
50     while(set.next()) {
51         if(name.equals((String)set.getObject("username")) && pass.equals((String)set.getObject("pa:
52             isLogin=true;
53             usersModel= new UsersModel((int)set.getObject("id"), (String)set.getObject("name"), (S
54                 , (String)set.getObject("passwd"), (int)set.getObject("role"));
55                 ROLE = (int) set.getObject("role");
56                 break;
57         }
58     }
59
60     // check username and password
61     if(!isLogin || (adModeBtn.selectedProperty().get() && ROLE!=1 )) {
62         Alert alert = new Alert(AlertType.WARNING, "Tài khoản hoặc mật khẩu không chính xác!", But
63         alert.setHeaderText(null);
64         alert.showAndWait();
65         return;
66     }
67     Stage stage = (Stage)((Node) event.getSource()).getScene().getWindow();
68     Parent home;
69     if(adModeBtn.selectedProperty().get()) {
70         home = FXMLLoader.load(getClass().getResource("/views/Admin.fxml"));
71     } else {
72         home = FXMLLoader.load(getClass().getResource("/views/Home3.fxml"));
73     }
74     stage.setScene(new Scene(home));
75     stage.centerOnScreen();
76     stage.setResizable(false);
77     stage.show();
78 }

```

Theo nội dung đoạn mã thực hiện chức năng đăng nhập thì chương trình sẽ thực hiện qua các bước như sau:

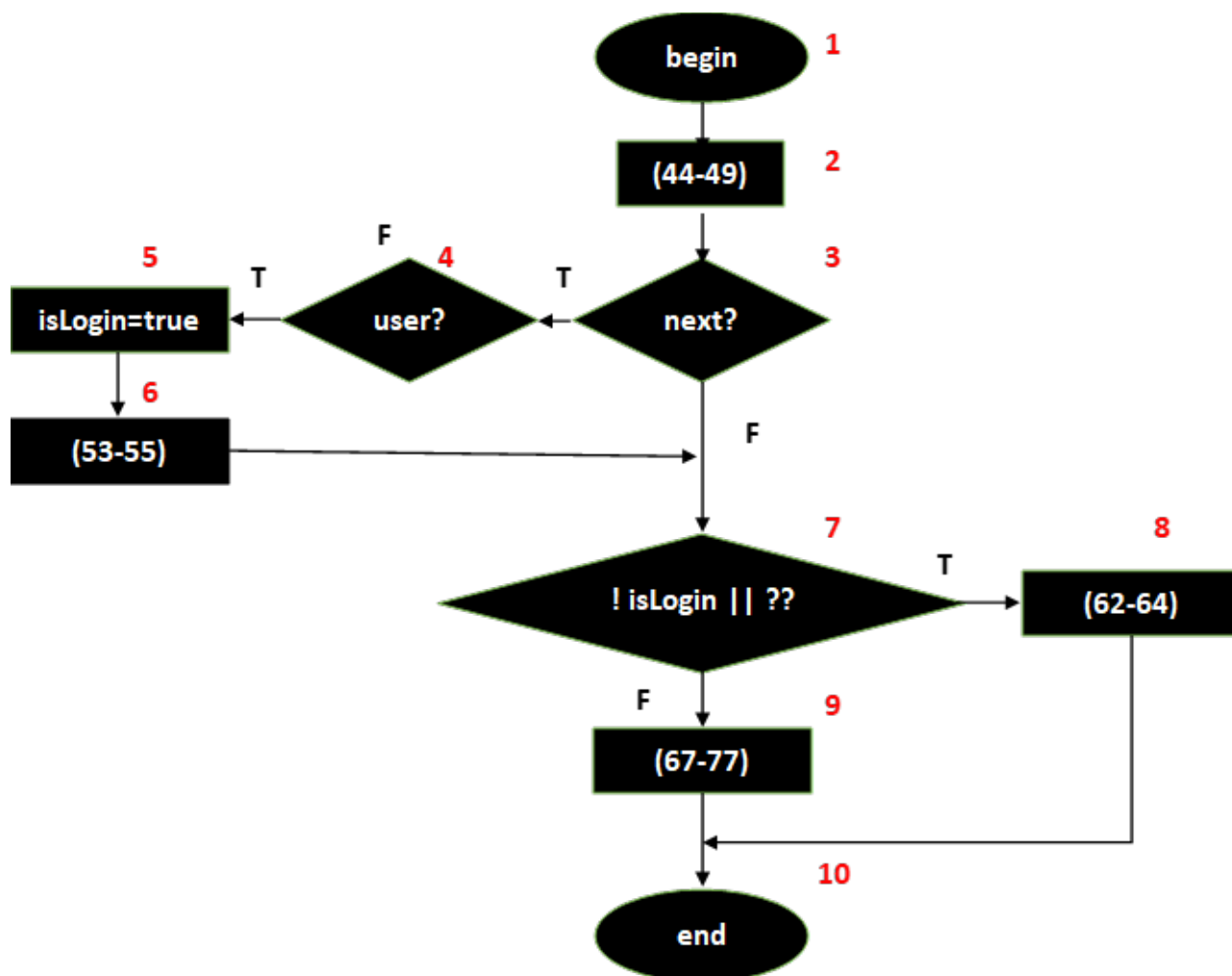
1. Lấy username & password từ giao diện đăng nhập của người dùng.
2. Thực hiện kết nối đến cơ sở dữ liệu và lấy thông tin các tài khoản của người dùng
3. Duyệt qua tất cả các tài khoản, nếu thấy có thông tin tài khoản trùng với thông tin đăng nhập thì hiển thị giao tiếp tương ứng với quyền của người dùng đang đăng nhập. Trong trường hợp ngược lại thì hiển thị thông báo "Tài khoản hoặc mật khẩu không chính xác".

Trong đó, thao tác kết với cơ sở dữ liệu được thực hiện qua class MySqlConnection như sau:

```
public class MySqlConnection {
    public static Connection getMySqlConnection() throws SQLException, ClassNotFoundException {
        String hostName = "localhost";
        String dbName = "test";
        String userName = "root";
        String password = "";
        return getMySqlConnection(hostName, dbName, userName, password);
    }

    public static Connection getMySqlConnection(String hostName, String dbName, String userName, String password)
        throws SQLException, ClassNotFoundException{
        //Class.forName("com.mysql.jdbc.Driver");
        String connectionUrl = "jdbc:mysql://" + hostName + ":3306/" + dbName + "?useUnicode=true&characterEncod
        String connectionUrl = "jdbc:mysql://" + hostName + ":3306/" + dbName ;
        Connection conn = DriverManager.getConnection(connectionUrl, userName, password);
        return conn;
    }
}
```

Nếu ta xét về cấu trúc và chức năng của chương trình với giả thiết là việc kết nối với dữ liệu thành công và việc lấy thông tin từ giao diện (username và password) là không gặp vấn đề nào thì ta thấy rằng các bước thực hiện ở trên ngoài vòng lặp duyệt qua các tài khoản của người dùng thì các câu lệnh được thực hiện một cách tuần tự. Đối với kỹ thuật kiểm thử hộp trắng, chúng ta xét phương pháp thiết kế các test case theo tiêu chí phủ mọi đường dẫn. Ở đây chúng ta cần vẽ sơ đồ luồng điều khiển cho đoạn chương trình trên. Kết quả được thể hiện ở trong hình.



Theo đó, các đường dẫn đi qua các node trên đồ thị có thể là:

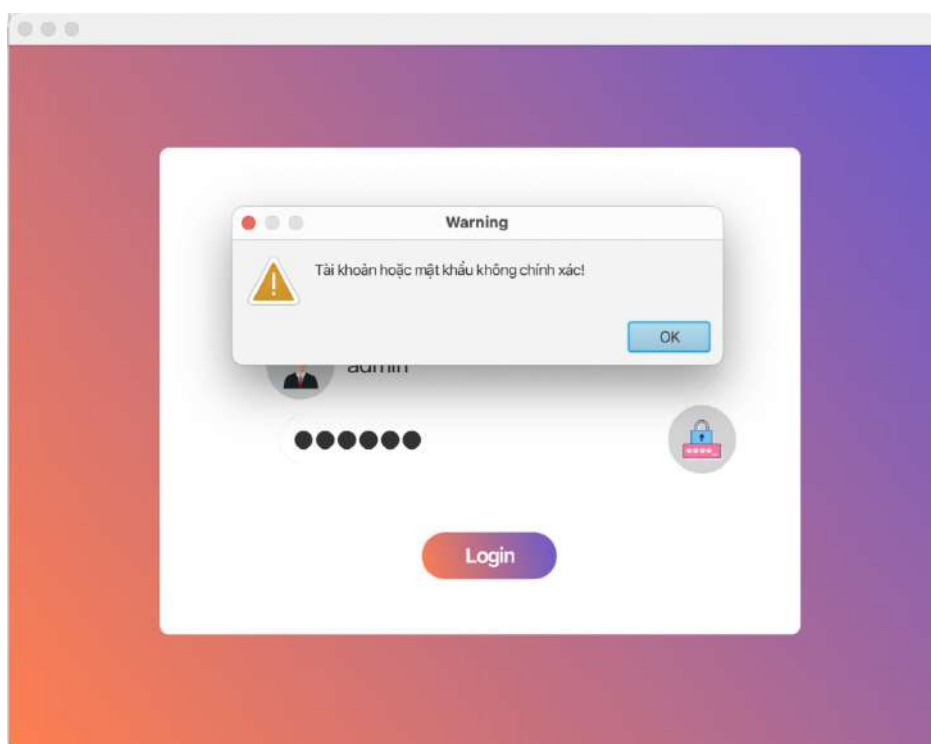
- (1) 1, 2, (3, 4)⁺, 5, 6, 7, 9, 10
- (2) 1, 2, (3, 4)⁺, 5, 6, 7, 8, 10
- (3) 1, 2, (3, 4)⁺, 7, 9, 10
- (4) 1, 2, (3, 4)⁺, 3, 7, 8, 10
- (5) 1, 2, 3, 7, 9, 10
- (6) 1, 2, 3, 7, 8, 10

Nhận xét: ở đây ta có tổng cộng 6 đường dẫn. Tuy nhiên, trong số 6 đường dẫn này thì có 2 đường dẫn (2) và (5) không thực thi được, lý do là biến **isLogin** không thể cùng nhận cả hai giá trị true và false (biến này được khởi tạo với giá trị ban đầu là false).

Trong cơ sở dữ liệu chúng ta đã có tài khoản như sau: username: **admin**, password: **123456789**. Theo đó, loại hai trường hợp (2) và (5) ở trên, chúng ta có thể thiết lập nên 4 test case tương ứng như sau:

Bảng 9-1: Các test case

TC	username	password	expected outcome	Kết quả
1	admin	123456789	chuyển sang form làm việc của admin	passed
2	admin	123456	TB "Tài khoản hoặc mật khẩu không chính xác"	passed
3	adminn	123456789	TB "Tài khoản hoặc mật khẩu không chính xác"	passed
4	adminn	12345678	TB "Tài khoản hoặc mật khẩu không chính xác"	passed



Hình 9-4: TC2 cho chức năng login

Kiểm thử hộp đen chức năng đăng nhập

Phân tích chức năng đăng nhập với kiểm thử hộp đen, chúng ta thấy

- Đầu vào (input) gồm 2 thông tin: username, password
- Đầu ra (output) có 2 khả năng như sau
 1. (TH1) TB: username hoặc password không chính xác;
 2. (TH2) chuyển sang Form làm việc

Ta lập bảng kết quả như sau:

Bảng 9-2: Các kết quả có thể

Username	true	true	false	false
Password	true	false	true	false
kết quả	(TH2) Form	(TH1) TB	(TH1) TB	(TH1) TB

Bảng giá trị này có thể được rút gọn thành bảng sau

Bảng 9-3: Bảng rút gọn các kết quả có thể

		Rule 1	Rule 2	Rule 3
điều kiện	Đúng username	false	-	true

	Đúng password	-	false	true
kết quả		"Tài khoản hoặc mật khẩu không chính xác"	"Tài khoản hoặc mật khẩu không chính xác"	Chuyển sang màn hình làm việc

Như vậy ta có thể thực hiện kiểm thử về mặt nghiệp vụ của chức năng này với 3 test case như sau:

1. Nhập đúng username & password, kết quả là hệ thống sẽ hiển thị form làm việc.
2. Nhập đúng username nhưng password sai, kết quả nhận được là thông báo "Tài khoản hoặc mật khẩu không chính xác".
3. Nhập sai username (muốn sử dụng theo dự kiến) nhưng nhập đúng password, kết quả nhận được là thông báo "Tài khoản hoặc mật khẩu không chính xác".

Ta thiết kế và thực hiện các ca kiểm thử trên và nhận được kết quả như sau:

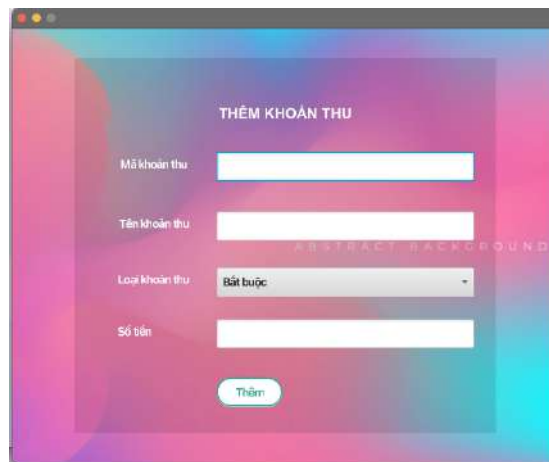
Bảng 9-4: Kết quả thực hiện các test case

TC	username	password	expected outcome	Kết quả
1	admin	123456789	chuyển sang form làm việc	passed
2	admin	12345678	TB "Tài khoản hoặc mật khẩu không chính xác"	passed
3	adminn	123456789	TB "Tài khoản hoặc mật khẩu không chính xác"	passed

Bài 9.2 Kiểm thử chức năng tạo khoản thu

Mô tả chức năng

- Đối với khoản thu, chúng ta có 4 thông tin cần được ghi nhận, đó là: mã khoản thu, tên khoản thu, loại khoản thu, và số tiền cần đóng.
- Với loại khoản thu ta có 2 hình thức đó là tự nguyện hoặc bắt buộc. Thông tin này được lựa chọn từ combobox trên giao diện của chức năng.



Hình 9-5: Giao diện thêm khoản thu

Các yêu cầu về mặt thông tin dành cho thêm khoản thu như sau:

- maKhoanThu là một dãy số từ 1 đến 11 chữ số
- tenKhoanThu là một chuỗi từ 1 đến 50 kí tự
- sotien là số và nhỏ hơn 11 chữ số

Mã nguồn cho chức năng này được chỉ ra ở bên dưới, chức năng được tiến hành thông qua hàm addKhoanThu trong class AddKhoanThu ở trong package controller.khoanthu như sau:

```

public void addKhoanThu(ActionEvent event) throws ClassNotFoundException, SQLException {
    Pattern pattern;

    // kiểm tra mã khoản thu nhập vào
    // mã khoản thu là dãy số từ 1 tới 11 chữ số
    pattern = Pattern.compile("\\d{1,11}");
    if (!pattern.matcher(tfMaKhoanThu.getText()).matches()) {
        Alert alert = new Alert(AlertType.WARNING, "Hãy nhập vào mã khoản thu hợp lệ!", ButtonType.OK);
        alert.setHeaderText(null);
        alert.showAndWait();
        return;
    }

    // kiểm tra mã khoản thu thêm mới có bị trùng với những mã khoản thu đã tồn tại hay không
    List<KhoanThuModel> listKhoanThuModels = new KhoanThuService().getListKhoanThu();
    for (KhoanThuModel khoanThuModel : listKhoanThuModels) {
        if (khoanThuModel.getMaKhoanThu() == Integer.parseInt(tfMaKhoanThu.getText())) {
            Alert alert = new Alert(AlertType.WARNING, "Mã khoản thu đã bị trùng!", ButtonType.OK);
            alert.setHeaderText(null);
            alert.showAndWait();
            return;
        }
    }

    // kiểm tra tên nhập vào
    // tên nhập vào là chuỗi từ 1 tới 50 kí tự
    if (tfTenKhoanThu.getText().length() >= 50 || tfTenKhoanThu.getText().length() <= 1) {
        Alert alert = new Alert(AlertType.WARNING, "Hãy nhập vào 1 tên khoản thu hợp lệ!", ButtonType.OK);
        alert.setHeaderText(null);
        alert.showAndWait();
        return;
    }

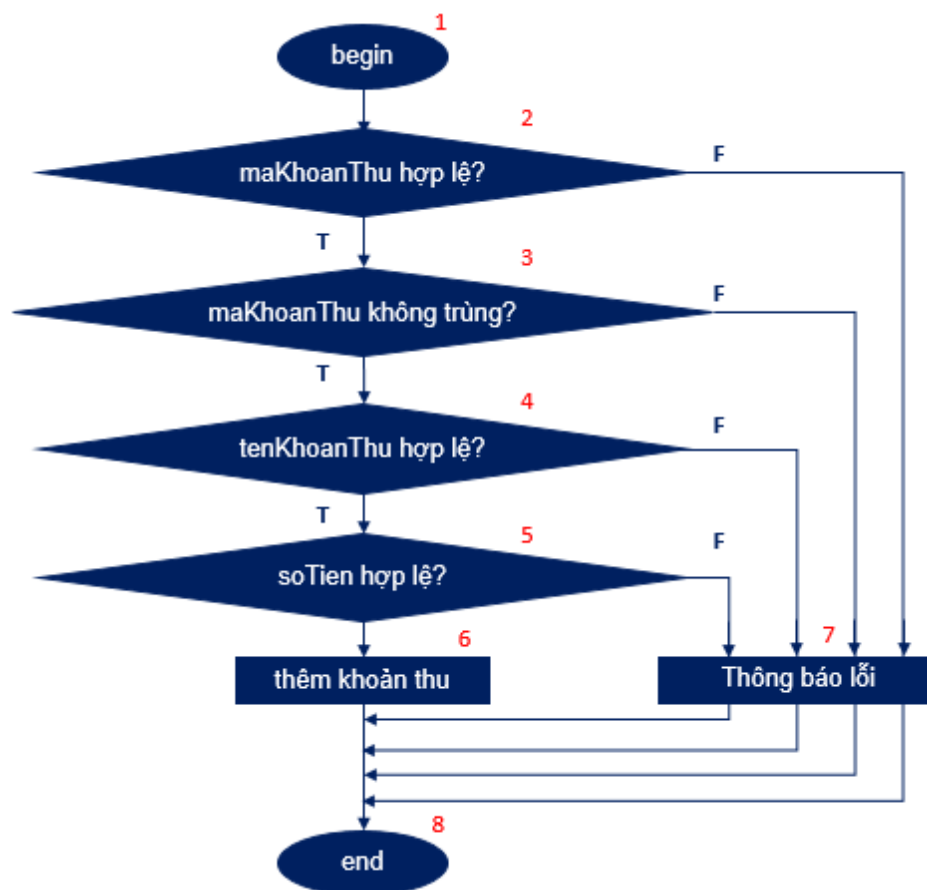
    // kiểm tra số tiền nhập vào
    // số tiền nhập vào phải là số và nhỏ hơn 11 chữ số
    pattern = Pattern.compile("^([1-9]\\d*(\\.\\d+)?)$");
    if (!pattern.matcher(tfSoTien.getText()).matches()) {
        Alert alert = new Alert(AlertType.WARNING, "Hãy nhập vào số tiền hợp lệ!", ButtonType.OK);
        alert.setHeaderText(null);
        alert.showAndWait();
        return;
    }

    // ghi nhận giá trị ghi tất cả đều đã hợp lệ
    SingleSelectionModel<String> loaiKhoanThuSelection = cbLoaiKhoanThu.getSelectionModel();
    String loaiKhoanThu_tmp = loaiKhoanThuSelection.getSelectedItem();

    int maKhoanThuInt = Integer.parseInt(tfMaKhoanThu.getText());
    String tenKhoanThuString = tfTenKhoanThu.getText();
    int soTienDouble = Integer.parseInt(tfSoTien.getText());
    int loaiKhoanThu;
    if (loaiKhoanThu_tmp.equals("Bắt buộc")) {
        loaiKhoanThu = 1;
    } else {
        loaiKhoanThu = 0;
    }

    new KhoanThuService().add(new KhoanThuModel(maKhoanThuInt, tenKhoanThuString, soTienDouble, loaiKhoanThu));
    Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
    stage.close();
}

```



Kiểm thử hộp trắng chức năng tạo khoản thu

Dựa vào cấu trúc của chương trình ta xây dựng đồ thị luồng điều khiển như sau:

Nhận thấy, với đoạn chương trình trên các đường dẫn có thể là:

- (1) 1, 2, 7
- (2) 1, 2, 3, 7
- (3) 1, 2, 3, 4, 7
- (4) 1, 2, 3, 4, 5, 7
- (5) 1, 2, 3, 4, 5, 6, 7

Ngoài ra, ta có dữ liệu của bảng khoản thu trong cơ sở dữ liệu như sau

The screenshot shows a database management interface for a table named 'khoan_thu'. The table has four columns: 'MaKhoanThu', 'TenKhoanThu', 'SoTien', and 'LoaiKhoanThu'. The data is as follows:

MaKhoanThu	TenKhoanThu	SoTien	LoaiKhoanThu
1	tiền thu rác	10000	0
2	nước	10000	1
3	Tiền Điện	200000	1

Hình 9-6: Dữ liệu khoản thu

Ta thiết kế và thực hiện các test case với kết quả thực hiện như sau:

Bảng 9-5: Các test case kiểm thử hộp trắng với tạo khoản thu

TC	maKhoanThu	tenKhoanThu	sotien	expected outcome	Kết quả	Ghi chú
1		a	5000	"Hãy nhập vào mã khoản thu hợp lệ!"	passed	Mã khoản thu trống
2	1	a	50000	"Mã khoản thu đã bị trùng!"	passed	Mã khoản thu trùng
3	4		50000	"Hãy nhập vào 1 tên khoản thu hợp lệ!"	passed	Tên khoản thu trống
4	4	tiền vệ sinh		"Hãy nhập vào số tiền hợp lệ!"	passed	số tiền bị trống

5	4	tiền vệ sinh	50000	"Thêm khoản thu thành công!"	passed	
---	---	--------------	-------	------------------------------	--------	--

Kiểm thử hộp đen chức năng tạo khoản thu

Với các thông tin đầu vào đã được mô tả, ta lập bảng quyết định với các điều kiện cho các thông tin như sau

Bảng 9-6: Bảng quyết định cho tạo khoản thu

		Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	
điều kiện	1<maKhoanThu<11	S	Đ	Đ	Đ	Đ	
	maKhoanThu không trùng	-	S	Đ	Đ	Đ	
	1<=Ten<=50	-	-	S	Đ	Đ	
	Số tiền < 11 chữ số	-	-	-	S	Đ	
kết quả		"Hãy nhập vào mã khoản thu hợp lệ!"	"Mã khoản đã bị trùng!"	"Hãy nhập vào 1 tên khoản thu hợp lệ!"	"Hãy nhập vào số tiền hợp lệ!"	"Thêm khoản thu thành công!"	

các test case được thiết kế như sau

Bảng 9-7: Các test case kiểm thử hộp đen với tạo khoản thu

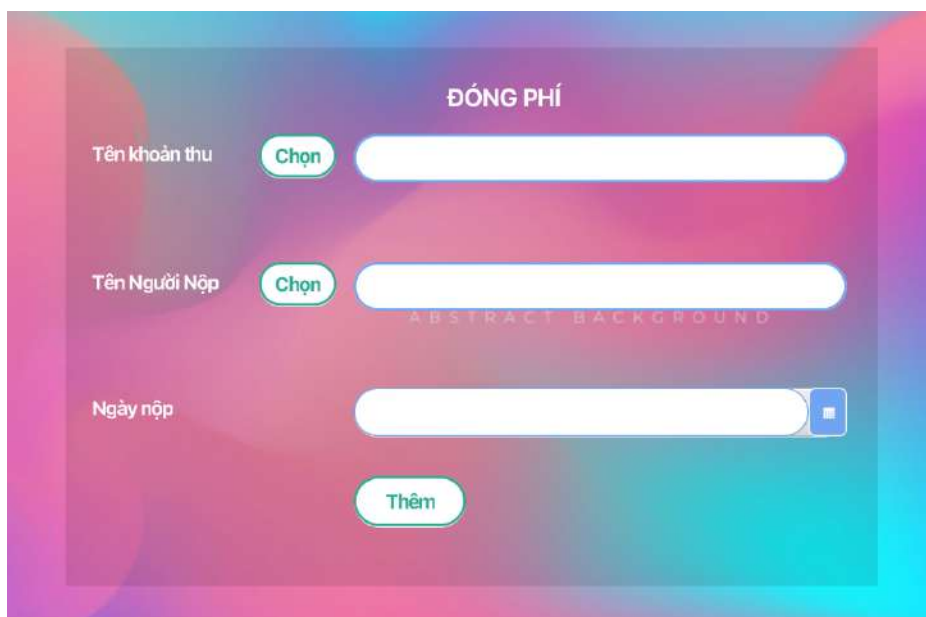
TC	maKhoant hu	tenKhoant hu	sotien	expected outcome	Kết quả	Ghi chú
1		a	5000	"Hãy nhập vào mã khoản thu hợp lệ!"	passed	Mã khoản thu trống
2	1	a	50000	"Mã khoản thu đã bị trùng!"	passed	Mã khoản thu trùng
3	4		50000	"Hãy nhập vào 1 tên khoản thu hợp lệ!"	passed	Tên khoản thu trống
4	4	tiền vệ sinh		"Hãy nhập vào số tiền hợp lệ!"	passed	số tiền bị trống
5	4	tiền vệ sinh	50000	"Thêm khoản thu thành công!"	passed	

Nhận xét: Trong trường hợp này, ta có thể sử dụng chung các test case cho cả kiểm thử hộp trắng lẫn kiểm thử hộp đen.

Bài 9.3 Kiểm thử chức năng thu phí

Mô tả chức năng

Thao tác nộp tiền được thực hiện thông qua giao diện đóng phí.



The image shows a web form titled "ĐÓNG PHÍ" (Payment) set against a vibrant, multi-colored abstract background. The form is organized into three main sections, each with a label on the left and an interactive element on the right. The first section is for "Tên khoản thu" (Revenue name), featuring a green "Chọn" (Select) button and a white text input field. The second section is for "Tên Người Nộp" (Payer name), also with a green "Chọn" (Select) button and a white text input field. The third section is for "Ngày nộp" (Payment date), which includes a white text input field and a small blue square icon with a white calendar symbol. Below these three sections is a single green "Thêm" (Add) button. The text "ABSTRACT BACKGROUND" is faintly visible in the center of the form area.

Hình 9-7: Giao diện đóng phí

Các thao tác thực hiện gồm:

1. Chọn khoản thu
2. Chọn người nộp
3. Chọn ngày nộp
4. Nhấn thêm

Lưu ý:

- các thao tác chọn khoản thu và chọn tên người nộp được thực hiện thông qua các giao diện khác,
- chúng ta không thực hiện việc kiểm tra logic về ngày thu.

Kiểm thử hộp trắng chức năng thu phí

Về mã nguồn chức năng thu phí được cài đặt trên class AddNopTien

```
public class AddNopTien {
    @FXML
    private TextField tfTenKhoanThu;
    @FXML
    private TextField tfTenNguoiNop;
    private KhoanThuModel khoanThuModel;
    private NhanKhuModel nhanKhuModel;

    public void chooseKhoanThu() throws IOException {
    }

    public void chooseNguoiNop() throws IOException {
    }

    public void addNopTien(ActionEvent event) throws ClassNotFoundException, SQLException {
    }
}
```

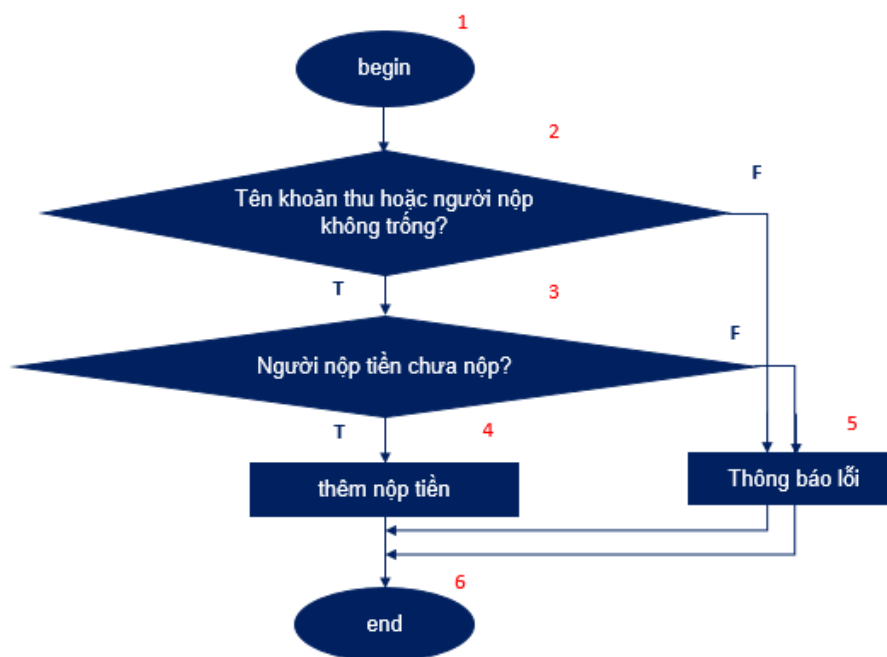
Thao tác thêm khoản thu được thực hiện thông qua hàm addNopTien như sau:

```
public void addNopTien(ActionEvent event) throws ClassNotFoundException, SQLException {
    if(tfTenKhoanThu.getText().length() == 0 || tfTenNguoiNop.getText().length() == 0) {
        Alert alert = new Alert(AlertType.WARNING, "Vui lòng nhập khoản nộp hợp lý!", ButtonType.OK);
        alert.setHeaderText(null);
        alert.showAndWait();
    } else {
        List<NopTienModel> listNopTien = new NopTienService().getListNopTien(null, null);
        for(NopTienModel nopTienModel : listNopTien) {
            if(nopTienModel.getIdNopTien() == nhanKhuModel.getId()
                && nopTienModel.getMaKhoanThu() == khoanThuModel.getMaKhoanThu()) {
                Alert alert = new Alert(AlertType.WARNING, "Người này đã từng nộp khoản phí này!", ButtonType.OK);
                alert.setHeaderText(null);
                alert.showAndWait();
                return;
            }
        }

        new NopTienService().add(new NopTienModel( nhanKhuModel.getId(), khoanThuModel.getMaKhoanThu()));
    }
    Stage stage = (Stage)((Node) event.getSource()).getScene().getWindow();
    stage.setTitle("Thêm nộp tiền");
    stage.setResizable(false);
    stage.close();
}
```

Thiết kế ca kiểm thử

Sơ đồ luồng điều khiển cho đoạn chương trình này được biểu diễn như sau



Thông tin về nộp tiền có trong CSDL như sau:

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

SELECT * FROM `nop_tien`

Profiling [Edit inline] [Edit]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key:

Extra options

	IDNopTien	MaKhoanThu	NgayThu
<input type="checkbox"/> Edit Copy Delete	2	2	2020-12-16
<input type="checkbox"/> Edit Copy Delete	3	1	2020-12-16
<input type="checkbox"/> Edit Copy Delete	3	2	2020-12-16

Hình 9-8: Dữ liệu nộp tiền

Theo đó, chúng ta sẽ thiết kế được các test case như sau

Bảng 9-8: Các test case kiểm thử hộp trắng với thêm khoản thu

TC	IDNopTien	MaKhoanThu	NgayThu	expected outcome	Ghi chú	Kết quả
1	2		2020-12-16	"Vui lòng nhập khoản	Mã khoản thu trống	passed

				nộp hợp lí!"		
2		1	2020-12-16	"Vui lòng nhập khoản nộp hợp lí!"	Mã người đóng tiền trống	passed
3	3	1	2020-12-16	"Người này đã từng nộp khoản phí này!"	Đã đóng tiền	passed
4	3	2	2020-12-16	"Thêm nộp tiền thành công"		passed

Kiểm thử hộp đen chức năng thu phí

Chúng ta lập bảng quyết định như sau

Bảng 9-9: Bảng quyết định với thêm khoản thu

		Rule 1	Rule 2	Rule 3	Rule 4
C	Chọn khoản thu	S	-	-	Đ
	Chọn người dùng	-	S	-	Đ
	Người dùng chưa đóng tiền khoản thu tương ứng	-	-	S	Đ

A		"Vui lòng nhập khoản nộp hợp lí!"	"Vui lòng nhập khoản nộp hợp lí!"	"Người này đã từng nộp khoản phí này!"	"Thêm nộp tiền"
---	--	---	---	---	--------------------

Kết quả thực hiện các test case như sau:

Bảng 9-10: Các test case kiểm thử hộp đen với thêm khoản thu

TC	IDNopTien	MaKhoanThu	NgayThu	expected outcome	Ghi chú	Kết quả
1	2		2020-12-16	"Vui lòng nhập khoản nộp hợp lí!"	Mã khoản thu trống	passed
2		1	2020-12-16	"Vui lòng nhập khoản nộp hợp lí!"	Mã người đóng tiền trống	passed
3	3	1	2020-12-16	"Người này đã từng nộp khoản phí này!"	Đã đóng tiền	passed
4	3	2	2020-12-16	"Thêm nộp tiền thành công"		passed

Nhận xét: Trong việc kiểm thử chức năng thu phí thì kiểm thử hộp đen và kiểm thử hộp trắng chúng ta đều cần đến 4 test case.

TÀI LIỆU THAM KHẢO

- [1] R. Pressman, Software Engineering: A Practitioner's Approach. 8th Ed., McGraw-Hill, 2016.
- [2] I. Sommerville, Software Engineering. 10th Ed., AddisonWesley, 2017.
- [3] Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Ed., Springer.
- [4] Shari Lawrence Pleege, Joanne M. Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009 .