

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**🙠**🕮**🙢**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**A close up of a sign

Description automatically generated**

**BÁO CÁO BÀI TẬP LỚN**

**PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG**

***ĐỀ TÀI***

**QUẢN LÝ PHÒNG THỰC HÀNH**

**- - - - - - - - - - 🙞🕮🙜- - - - - - - - - -**

|  |  |  |
| --- | --- | --- |
| *Giáo viên hướng dẫn* | *:* | *ThS. Vũ Thị Dương* |
| *Nhóm số* | *:* | *10* |
| *Mã lớp* | *:* | *20203IT6029001* |

Hà Nội, 2021

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**🙠**🕮**🙢**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**A close up of a sign

Description automatically generated**

**BÁO CÁO HỌC PHẦN:**

**PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: QUẢN LÝ PHÒNG THỰC HÀNH**

|  |  |  |
| --- | --- | --- |
| Giáo viên hướng dẫn | : | ThS. Vũ Thị Dương |
| Nhóm | : | 10 |
| Mã lớp | : | 20203IT6029001 – Khóa 13 |
| Sinh viên thực hiện | : | Lê Thị Hảo – 2018604750  Phạm Quang Trung – 2018605029  Dương Thị Huyền – 2018604661  Trần Quốc Đạt – 2018604917  Phạm Minh Tuấn– 2018605228 |

Hà Nội, 2021

# LỜI NÓI ĐẦU

Với sự phát triển không ngừng, công nghệ thông tin ngày nay đã len lỏi vào hầu hết mọi lĩnh vực trong đời sống. Trong đó, công nghệ phần mềm ngày càng được ứng dụng nhiều hơn trong việc quản lý cuộc sống thường ngày của chúng ta.

Nắm bắt được nhu cầu ứng dụng CNTT để quản lý trong cuộc sống, áp dụng những gì đã học trong học phần Lập trình trên thiết bị di động, nhóm chúng em dưới sự giúp đỡ của cô Vũ Thị Dương đã quyết định tìm hiểu và xây dựng một phần mềm đơn giản và gần gũi với thực tế là *Phần mềm quản lý phòng thực hành*.

Phần mềm hỗ trợ quản lý phòng thực hành được nhóm em xây dựng nhằm hỗ trợ người quản lý, giáo viên có thể dễ dàng cập nhập tình hình đăng ký giảng dạy ở các phòng thực hành, tình trạng các thiết bị trong các phòng thực hành và quản lý thiết bị tránh việc hỏng hóc mà không phát hiện kịp thời làm mất nhiều thời gian cũng như khó khăn trong việc lưu trữ và xử lý thông tin.

Bài tập lớn này được hoàn thành dưới sự đóng góp của các thành viên trong nhóm và sự hỗ trợ của giảng viên bộ môn Lập trình trên thiết bị di động. Xong trong quá trình nghiên cứu và hoàn thành bài tập lớn không tránh khỏi thiếu sót, rất nhận được đóng góp của cô để đề tài của chúng em được đầy đủ và hoàn chỉnh hơn.

***Chúng em xin chân thành cảm ơn!***

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc81168393)

[DANH MỤC BẢNG BIỂU 4](#_Toc81168394)

[DANH MỤC HÌNH ẢNH 4](#_Toc81168395)

[CHƯƠNG 1. MỞ ĐẦU 6](#_Toc81168396)

[1.1. Tổng quan 6](#_Toc81168397)

[1.1.1. Tổng quan về lập trình trên thiết bị di động 6](#_Toc81168398)

[1.1.2. Lý do chọn đề tài 7](#_Toc81168399)

[1.2. Tóm lược ý tưởng 7](#_Toc81168400)

[1.3. Phân tích tính khả thi 8](#_Toc81168401)

[1.4. Lịch biểu công việc 8](#_Toc81168402)

[1.4.1. Thành viên nhóm 9](#_Toc81168403)

[1.4.2. Thời gian tổng thể 9](#_Toc81168404)

[1.4.3. Thời gian chi tiết 9](#_Toc81168405)

[CHƯƠNG 2. KẾT QUẢ NGHIÊN CỨU 11](#_Toc81168406)

[2.1. Yêu cầu bài toán 11](#_Toc81168407)

[2.2. Xây dựng bài toán 11](#_Toc81168408)

[2.2.1. Phân tích yêu cầu người dùng. 11](#_Toc81168409)

[2.2.1.1. Các yêu cầu chức năng 11](#_Toc81168410)

[2.2.1.2. Các yêu cầu phi chức năng 12](#_Toc81168411)

[2.2.1.3. Biểu đồ Use case của hệ thống 13](#_Toc81168412)

[2.2.1.4. Mô tả biểu đồ lớp cơ sở dữ liệu 16](#_Toc81168413)

[2.2.2. Thiết kế hệ thống 16](#_Toc81168414)

[2.3. Thực hiện bài toán 23](#_Toc81168415)

[2.3.1. Quản lý đăng nhập và người dùng (Phạm Minh Tuấn) 23](#_Toc81168416)

[2.3.2. Quản lý phòng thực hành (Dương Thị Huyền) 27](#_Toc81168417)

[2.3.3. Quản lý thiết bị (Lê Thị Hảo) 32](#_Toc81168418)

[2.3.4. Quản Lý phiếu xác nhận (Trần Quốc Đạt) 36](#_Toc81168419)

[2.3.5. Quản lý phiếu đăng ký (Phạm Quang Trung) 39](#_Toc81168420)

[CHƯƠNG 3. KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM 43](#_Toc81168421)

[3.1. Nội dung đã thực hiện 43](#_Toc81168422)

[3.2. Hướng phát triển 44](#_Toc81168423)

[KẾT LUẬN 45](#_Toc81168424)

[PHỤ LỤC 46](#_Toc81168425)

[TÀI LIỆU THAM KHẢO 47](#_Toc81168426)

# DANH MỤC BẢNG BIỂU

[Bảng 1‑1: Thành viên nhóm 8](#_Toc81154650)

[Bảng 1‑2: Thời gian khảo sát bài toán 8](#_Toc81154651)

[Bảng 1‑3: Thời gian lập kế hoạch thực hiện phần mềm 8](#_Toc81154652)

[Bảng 1‑4: Thời gian phân tích và thiết kế 9](#_Toc81154653)

[Bảng 1‑5: Thời gian viết mã 9](#_Toc81154654)

[Bảng 1‑6: Thời gian bàn giao sản phẩm và báo cáo 9](#_Toc81154655)

[Bảng 2‑1: Phân công các yêu cầu chức năng 11](#_Toc81154656)

# DANH MỤC HÌNH ẢNH

[Hình 2‑1: Sơ đồ tổng quát biểu đồ use case của hệ thống 13](#_Toc81162581)

[Hình 2‑2: Use case đăng nhập 13](#_Toc81162582)

[Hình 2‑3: Use case quản lý phòng thực hành 14](#_Toc81162583)

[Hình 2‑4: Use case quản lý thiết bị 14](#_Toc81162584)

[Hình 2‑5: Use case đăng ký thực hành 15](#_Toc81162585)

[Hình 2‑6: Use case quản lý phiếu xác nhận 15](#_Toc81162586)

[Hình 2‑7: Biểu đồ lớp cơ sở dữ liệu 16](#_Toc81162587)

[Hình 2‑8:Màn hình đăng nhập 17](#_Toc81162588)

[Hình 2‑9: Màn hình chính 17](#_Toc81162589)

[Hình 2‑10: Thông tin đăng nhập 18](#_Toc81162590)

[Hình 2‑11: Đăng ký phòng thực hành 18](#_Toc81162591)

[Hình 2‑12: Quản lý thiết bị trong phòng 18](#_Toc81162592)

[Hình 2‑13: Danh sách thiết bị 18](#_Toc81162593)

[Hình 2‑14: Quản lý thông tin phiếu xác nhận 19](#_Toc81162594)

[Hình 2‑15: Danh sách phòng thực hành 20](#_Toc81162595)

[Hình 2‑16: Cập nhật phòng thực hành 20](#_Toc81162596)

[Hình 2‑17: Màn hình quản lý đăng nhập và người dùng 24](#_Toc81162597)

[Hình 2‑18: Thông tin đăng nhập 24](#_Toc81162598)

[Hình 2‑19: Layout đăng nhập 25](#_Toc81162599)

[Hình 2‑20: Layout đăng ký 25](#_Toc81162600)

[Hình 2‑21: Layout màn hình chính 26](#_Toc81162601)

[Hình 2‑22: Quản lý danh sách phòng học 27](#_Toc81162602)

[Hình 2‑23: Thêm mới phòng thực hành 27](#_Toc81162603)

[Hình 2‑24: Cập nhật phòng thực hành 28](#_Toc81162604)

[Hình 2‑25: Layout chính của chức năng quản lý phòng thực hành 29](#_Toc81162605)

[Hình 2‑26: Layout thêm phòng, sửa phòng 29](#_Toc81162606)

[Hình 2‑27: Danh sách thiết bị 32](#_Toc81162607)

[Hình 2‑28: Quản lý thiết bị trong phòng 32](#_Toc81162608)

[Hình 2‑29: Layout thêm, sửa thông tin thiết bị 33](#_Toc81162609)

[Hình 2‑30: Danh sách phiếu xác nhận 36](#_Toc81162610)

[Hình 2‑31: Quản lý thông tin phiếu xác nhận 36](#_Toc81162611)

[Hình 2‑32: Layout sửa, thêm thông tin của phiếu xác nhận 37](#_Toc81162612)

[Hình 2‑33: Đăng ký phòng thực hành 39](#_Toc81162613)

[Hình 2‑34: Giao diện trên XML của phiếu đăng ký 40](#_Toc81162614)

# MỞ ĐẦU

## Tổng quan

### Tổng quan về lập trình trên thiết bị di động

Trên thị trường ứng dụng di động hiện nay, 3 hệ điều hành chiếm thị phần cao nhất là: Android, iOS và Window phone, tiếp sau là 1 số hệ điều hành khác như BlackBerry…

Trong đó, Android là một hệ điều hành có dạng mã nguồn mở, nó hoạt động dựa trên nền tảng Linux và được thiết kế dành riêng cho những thiết bị di động cảm ứng hoặc máy tính bảng. Trước đây, hệ điều hành này được phát triển bởi tổng công ty Android và được tài trợ bởi Google. Cho đến năm 2005 thì Google đã mua lại hệ điều hành này và cho ra mắt người dùng vào năm 2007.

Android này sở hữu mã nguồn mở nên lập trình viên có thể dễ dàng điều chỉnh và phân phối nó một cách tự do. Đây chính là một trong những yếu tố đã giúp cho Android trở thành nền tảng xây dựng điện thoại thông minh phát triển nhất trên thế giới.

Một số ngôn ngữ sử dụng để lập trình android: Hiện nay, hệ điều hành Android sử dụng một số loại ngôn ngữ lập trình khác nhau như: Java, C, C++, CSS, Python, Lua, XML, … Đây chính là một trong những điểm cộng giúp lập trình viên dễ làm việc hơn với Android. Đặc biệt là các fresher có thể tiếp cận dễ dàng hơn tới môi trường của hệ điều hành Android. Trong số ngôn ngữ trên thì Java được coi là ngôn ngữ lập trình chính thức của Android và đây là ngôn ngữ mà lập trình viên Android cần phải tiếp xúc thường xuyên nhất.

Java là một trong những ngôn ngữ lập trình chính thức được sử dụng chủ yếu trong hệ điều hành Android. Java đã được thiết kế nhằm tương thức với đa số môi trường phát triển nên nó thường linh hoạt hơn so với các ngôn ngữ lập trình C/C++ khác. Bên cạnh đó thì Java có hiệu suất cao và có trình giải phóng bộ nhớ đến các đối tượng không được sử dụng đến.

Hiện nay, Java còn được viết nâng cao để có thể viết ra được những chương trình thực thi từ các vùng tác vụ cùng một lúc nhờ tính năng đa luồng. Ngoài ra, ngôn ngữ lập trình Java còn hỗ trợ bảo mật tốt nhờ các thuật toán dạng mã hóa như: public key hoặc mã one way hashing…

### Lý do chọn đề tài

Trong nhiều trường Đại học hiện tại vẫn còn việc đăng ký lịch thực hành bằng sổ sách, việc đăng ký bằng sổ sách này rất khó khăn trong việc nắm bắt tổng quát về các giảng viên đăng ký, các phòng được đăng ký, các thiết bị trong phòng. Việc lưu lại thông tin trên sổ cũng vô cùng bất tiện, cũng như việc xác định các thiết bị hỏng hóc máy trong phòng sẽ mất rất nhiều thời gian Do đã có nền tảng xây dựng phần mềm sử dụng ngôn Java nên các thành viên trong nhóm đã tiếp tục học về Lập trình trong thiết bị di động. Qua những gì đã học trong học phần Lập trình trên thiết bị di động, nhóm chúng em dưới sự giúp đỡ của cô Vũ Thị Dương đã quyết định tìm hiểu và xây dựng một phần mềm đơn giản và gần gũi với thực tế là *Phần mềm quản lý phòng thực hành*.

## Tóm lược ý tưởng

Phần mềm Quản lý phòng thực hành giúp nâng cao hiệu quả quản lý cũng như tiết kiệm nhiều chi phí cho việc quản lý.

Trong phần mềm chúng em xây dựng bao gồm các nội dung quản lý:

* Quản lý phiếu đăng ký thực hành: cho phép đăng ký lịch thực hành và quản lý danh sách phiếu.
* Quản lý đăng ký và đăng nhập: cho phép người dùng đăng ký và đăng nhập vào hệ thống.
* Quản lý phiếu xác nhận: quản lý các thông tin sau mỗi ca thực hành.
* Quản lý phòng thực hành: quản lý tổng quát số lượng thiết bị và thông tin của từng phòng.
* Quản lý thiết bị trong phòng: quản lý tình trạng các thiết bị trong phòng

Nhóm sinh viên chúng em cùng nhau xây dựng Phần mềm quản lý phòng thực hành để hỗ trợ người quản lý và giáo viên của họ quản lý, đăng ký phòng một cách hiện đại chuyên nghiệp. Do số lượng các giáo viên đăng ký lớn nếu quản lý thủ công sẽ tốn nhiều công sức và nhiều khi không đảm bảo được chính xác. Việc dùng phần mềm trong quản lý này sẽ giúp đỡ rất nhiều trong việc quản lý phòng thực hành.

*Mục tiêu:* Hệ thống giúp quản lý dễ dàng các thông tin giảng viên, phòng thực hành, phiếu đăng ký thực hành, phiếu xác nhân, và các thiết bị trong phòng thực hành. Hệ thống dễ sử dụng và tiết kiệm thời gian, giao diện phù hợp người dùng.

## Phân tích tính khả thi

Phần mềm được phát triển dựa trên ngôn ngữ Java và hệ quản trị CSDL MySQL là những mã nguồn mở miễn phí hỗ trợ mạnh, ít tốn tài nguyên và giúp tiết kiệm chi phí.

Được hỗ trợ đa nền tảng và độc lập với nền nên có thể chạy trên nhiều hệ điều hành khác nhau.

Phần mềm sau khi hoàn thành sẽ được người quản lý phòng thực hành, giáo viên nắm bắt được tình hình các phòng, buổi, ca thực hành của mình, các thiết bị được quản lý chặt chẽ, đăng ký giảng dạy trên phần mềm nhanh chóng … sử dụng dễ dàng và hiệu quả.

## Lịch biểu công việc

### Thành viên nhóm

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSV | Ghi chú |
| 1 | Lê Thị Hảo | 2018604750 |  |
| 2 | Phạm Quang Trung | 2018605029 |  |
| 3 | Dương Thị Huyền | 2018604661 |  |
| 4 | Trần Quốc Đạt | 2018606917 |  |
| 5 | Phạm Minh Tuấn | 201860528 |  |

Bảng 1‑1: Thành viên nhóm

### Thời gian tổng thể

Thời gian dự kiến thực hiện đề tài là từ tuần 4 - 6 tuần, hoàn thành trước hạn là 4 ngày:

* Phân tích và mô tả chức năng: 5 ngày
* Lập trình chức năng hệ thống, sửa lỗi, nộp báo cáo: 4 - 5 tuần

### Thời gian chi tiết

* ***Giai đoạn 1:*** Khảo sát bài toán (cả nhóm)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nhiệm vụ | Số ngày | Phân công |
| 1 | Tìm hiểu đề tài, xác định yêu cầu cần thực hiện. | 1 | Cả nhóm |
| 2 | Xây dựng tài liệu đặc tả theo yêu cầu muốn đặt ra. | 1 | Đạt |

Bảng 1‑2: Thời gian khảo sát bài toán

* ***Giai đoạn 2:*** Lập kế hoạc thực hiện phần mềm (Lê Thị Hảo)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nhiệm vụ | Số ngày | Phân công |
| 1 | Phân công công việc cho các thành viên trong nhóm để thực hiện. | 1 | Hảo |

Bảng 1‑3: Thời gian lập kế hoạch thực hiện phần mềm

* ***Giai đoạn 3:*** Phân tích và Thiết kế (Cả nhóm)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nhiệm vụ | Số ngày | Phân công |
| 1 | Phân tích yêu cầu, thiết kế về mặt chức năng và mô tả khung giao diện | 4 | Cá nhân |
| 2 | Hoàn thành tài liệu phân tích thiết kế hệ thống | 3 | Huyền |
| 3 | Thiết kế giao diện của phần mềm | 7 | Cá nhân |

Bảng 1‑4: Thời gian phân tích và thiết kế

* ***Giao đoạn 4:*** Viết mã (Cả nhóm)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nhiệm vụ | Số ngày | Phân công |
| 1 | Lập trình viết code các chức năng phần mềm dựa trên bản phân tích đã có. | 3 tuần | Các thành viên thực hiện dưới sự phần công của trưởng nhóm. [*[Phụ lục 1]*](#_PHỤ_LỤC) |
| 2 | Tiến hành kiểm lỗi và thực hiện chỉnh sửa để hoàn thiện. | 2 ngày |

Bảng 1‑5: Thời gian viết mã

* ***Giai đoạn 5:*** Bàn giao sản phẩm và báo cáo (Cả nhóm)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nhiệm vụ | Số ngày | Phân công |
| 1 | Xây dựng bản báo cáo hoàn thiện sản phẩm | 2 | Hảo, Huyền |

Bảng 1‑6: Thời gian bàn giao sản phẩm và báo cáo

# KẾT QUẢ NGHIÊN CỨU

## Yêu cầu bài toán

Bài toán “Quản Lý phòng thực hành” được đặt ra nhằm giải quyết vấn đề đăng ký phòng thực hành và xem lịch thực hành của giảng viên mà không cần đến giấy tờ. Có thể quản lý được mọi thiết bị cũng như các thông tin liên quan đến phòng thực hành. Nhóm 10 chúng em đã thực hiện bài toán dựa theo yêu cầu của giảng viên bộ môn.

Theo yêu cầu ban đầu bài toán chủ yếu giải quyết 5 chức năng chính:

1. Quản lý danh sách tài khoản đăng ký và đăng nhập hệ thống: đây là chức năng quan trọng của hệ thống nhằm mục đích quản lý và không có những tài khoản lậu, không được phép xâm nhập vào hệ thống sửa đổi thông tin.
2. Quản lý phòng thực hành: Quản lý các thông tin phòng như là: tên phòng, vị trí, số lượng máy và thông tin tổng quan về số lượng các thiết bị khác.
3. Quản lý phiếu đăng ký: Giảng viên có thể đăng ký lịch thực hành và có thể thay đổi buổi lý thuyết sang buổi thực hành
4. Quản lý phiếu xác nhận: Quản lý các thông tin mà giảng viên hoặc lớp trưởng xác nhận lại số lượng các thiết bị sau khi ca thực hành kết thúc.
5. Quản lý người dùng: Quản lý các tài khoản có thể sử dụng hệ thống.

## Xây dựng bài toán

### Phân tích yêu cầu người dùng.

#### Các yêu cầu chức năng

|  |  |
| --- | --- |
| Chức năng | Người thực hiện |
| Đăng nhập, đăng xuất | Phạm Minh Tuấn |
| Quản lý người dùng |
| Quản lý phòng thực hành | Dương Thị Huyền |
| Quản lý thiết bị | Lê Thị Hảo |
| Quản lý phiếu xác nhận | Trần Quốc Đạt |
| Quản lý phiếu đăng ký | Phạm Quang Trung |

Bảng 2‑1: Phân công các yêu cầu chức năng

#### Các yêu cầu phi chức năng

Các yêu cầu phi chức năng chung cho các hệ thống thông tin cung cấp dịch vụ công trực tuyến, gồm 7 chức năng cơ bản, đó là:

* Hiệu năng hoạt động: Yêu cầu về thời gian, tài nguyên sử dụng, công suất tối đa.
* Tương thích: Cùng tồn tại, tương tác liên thông.
* Tính khả dụng: Phù hợp với nhu cầu, dễ dàng học cách sử dụng, giao diện người sử dụng, khả năng truy cập.
* Tính tin cậy: Trưởng thành, sẵn sàng, khả năng phục hồi
* An toàn thông tin: Bảo mật, toàn vẹn, xác thực.
* Văn hóa và chính trị: sử dụng ngôn ngữ việt, hợp pháp và sử dụng hợp lý.
* Các yêu cầu phi chức năng khác: Yêu cầu phi chức năng cho công cụ tim kiếm, thống kê, xây dựng báo cáo, phát triển ứng dụng là yêu cầu, định hướng lựa chọn công nghệ nền tảng dùng để xây dựng, phát triển hệ thống trong các dịch vụ công trực tuyến được triển khai độc lập, ưu tiên lựa chọn hệ quản trị cơ sở dữ liệu miễn phí đối với các phần mềm ứng dụng nội bộ đang có trên thị trường, ưu tiên lựa chọn các phần mềm miễn phí mã nguồn mở.

#### Biểu đồ Use case của hệ thống

* **Sơ đồ tổng quát**

****

Hình 2‑1: Sơ đồ tổng quát biểu đồ use case của hệ thống

* **Usecase đăng nhập (Phạm Minh Tuấn)**

Use case cho phép người dùng đăng nhập vào hệ thống trước khi muốn sử dụng các chức năng tiếp theo của hệ thống.



Hình 2‑2: Use case đăng nhập

* **Usecase Quản lý phòng thực hành (Dương Thị Huyền)**

Use case quản lý thông tin phòng thực hành và tổng quát số lượng các thiết bị của phòng và cho phép người dùng xem thông tin phòng thực hành, có thể thêm sửa, xóa thông tin phòng thực hành.

****

Hình 2‑3: Use case quản lý phòng thực hành

* **Usecase Quản lý thiết bị (Lê Thị Hảo)**

Use case này quản lý các thông tin của thiết bị, tình trạng của thiết bị và thêm, sửa, xóa thiết bị.



Hình 2‑4: Use case quản lý thiết bị

* **Usecase đăng kí thực hành (Phạm Quang Trung)**

Use case này cho phép giảng viên đăng ký phòng thực hành.



Hình 2‑5: Use case đăng ký thực hành

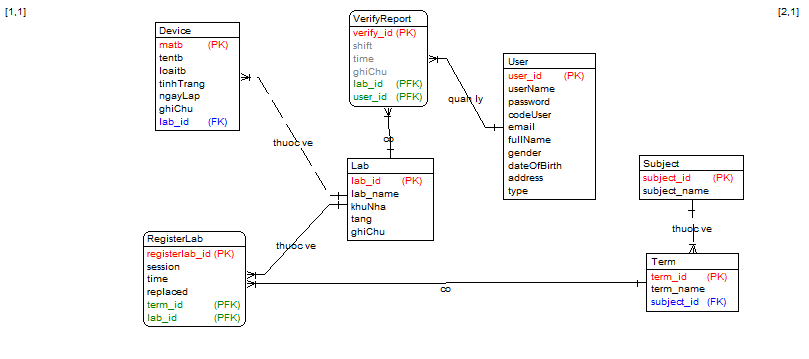
* **Usecase Quản lý phiếu xác nhận (Trần Quốc Đạt)**

Use case này quản lý danh sách các phiếu xác nhận của giảng viên sau khi ca thực hành kết thúc.



Hình 2‑6: Use case quản lý phiếu xác nhận

#### Mô tả biểu đồ lớp cơ sở dữ liệu



Hình 2‑7: Biểu đồ lớp cơ sở dữ liệu

Lớp Device ánh xạ sang bảng Device

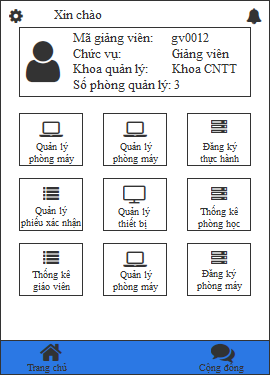
|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| matb | int | Mã thiết bị |
| tentb | String | Tên thiết bị |
| loaitb | String | Loại thiết bị |
| tinhTrang | String | Tình trạng |
| ngayLap | Date | Ngày lắp thiết bị |
| ghiChu | String | Ghi chú |
| lab\_id | int | Mã phòng thực hành |

### Thiết kế hệ thống

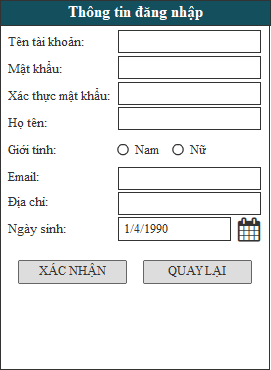
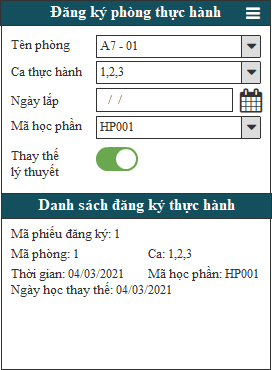
* **Thống nhất khi thiết kế giao diện**
* Màu sắc: màu chủ đạo là màu xanh và màu trắng tạo cảm giác dịu mắt cho người dùng không mỏi mắt.
* Bố trí nút lệnh: các nút đều là option menu trên góc phải màn hình và button.
* Ô nhập liệu: chủ yếu là Spinner, EditText và TextView
* Giao diện: Có màn hình đăng nhập và trang chủ là màn hình menu hiển thị các chức năng có trong hệ thống:

1. Quản lý phòng thực hành: hiển thị list danh sách các phòng và thông tin phòng. Có button (+) thêm phòng, chọn phòng để sửa thông tin của phòng.
2. Quản lý thiết bị: hiển thị list danh sách các thiết bị và thông tin chi tiết, tình trạng của thiết bị. Có button (+) thêm thiết bị, chọn thiết bị để sửa thông tin.
3. Quản lý phiếu xác nhận: hiển thị list danh sách phiếu xác nhận của giảng viên sau ca thực hành, thêm, sửa và xóa phiếu.
4. Đăng ký phòng thực hành: hiển thị các ô nhập liệu và danh sách các lịch đã đăng ký.

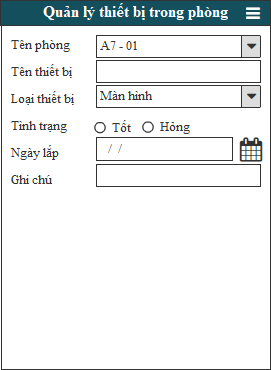
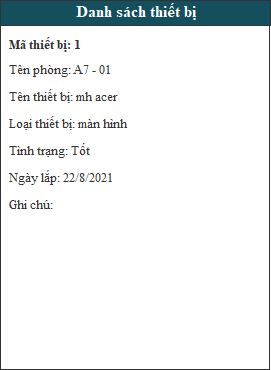
* Quy tắc đặt tên:
* Java: tên chức năng Activity
* XML: activity\_ten chuc năng
* Menu: menu\_chuc nang
* Bố trí package:
* Tạo package riêng cho từng chức năng
* Một package object gồm tất cả các đối tượng của bài toán.
* Một package sqlite
* Một package util: gồm lớp FakeData
* **Giao diện chính của bài toán:**

** **

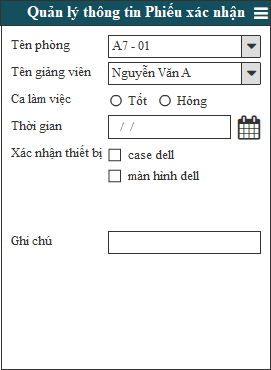
|  |  |
| --- | --- |
| Hình 2‑8:Màn hình đăng nhập | Hình 2‑9: Màn hình chính |

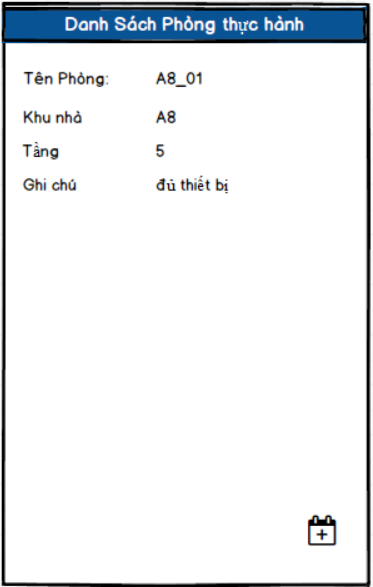
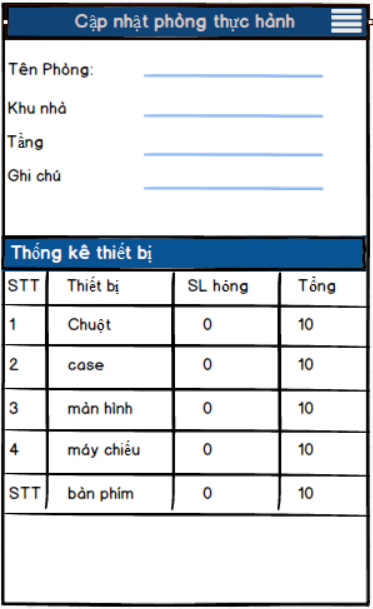
** **

|  |  |
| --- | --- |
| Hình 2‑10: Thông tin đăng nhập | Hình 2‑11: Đăng ký phòng thực hành |

** **

|  |  |
| --- | --- |
| Hình 2‑12: Quản lý thiết bị trong phòng | Hình 2‑13: Danh sách thiết bị |

****

Hình 2‑14: Quản lý thông tin phiếu xác nhận

|  |  |
| --- | --- |
| Hình 2‑15: Danh sách phòng thực hành | Hình 2‑16: Cập nhật phòng thực hành |

* **Thiết kế dữ liệu**

Sử dụng SQLite để thiết kế dữ liệu:

1. Lớp VerityReport: gồm 6 thuộc tính là: ID, maphong, user\_id, shift, time, note. Tương ứng với các thuộc tính này, ta tạo bảng TBL\_VERITY tương ứng trog SQLite:

*// tbl\_VerifyReport***public static final** String ***TBL\_VERIFY*** = **"verify\_report"**;  
**public static final** String ***VERIFY\_ID*** = **"verify\_id"**;  
**public static final** String ***VERIFY\_LAB\_ID*** = **"lab\_id"**;  
**public static final** String ***VERIFY\_USER\_ID*** = **"user\_id"**;  
**public static final** String ***VERIFY\_SHIFT*** = **"verify\_shift"**;  
**public static final** String ***VERIFY\_TIME*** = **"verify\_time"**;  
**public static final** String ***VERIFY\_NOTE*** = **"verify\_note"**;

**public static final** String ***CREATE\_TBL\_VERIFY*** = String.*format*(**""** +  
 **"create table if not exists %s ("** +  
 **" %s integer primary key autoincrement,"** +  
 **" %s integer not null, "** +  
 **" %s integer not null,"** +  
 **" %s integer not null,"** +  
 **" %s text,"** +  
 **" %s text)"**,  
 ***TBL\_VERIFY***,***VERIFY\_ID***, ***VERIFY\_LAB\_ID***, ***VERIFY\_USER\_ID***, ***VERIFY\_SHIFT***, ***VERIFY\_TIME***, ***VERIFY\_NOTE***);

1. Lớp User: gồm 10 thuộc tính là: user\_id, user\_name, user\_pasword, user\_code\_teacher, user\_email, user\_fullName, user\_gender, user\_dob, user\_address, user\_type. Tương ứng với các thuộc tính này, ta tạo bảng TBL\_USER tương ứng trog SQLite:

**public static final** String ***TBL\_USER*** = **"user"**;  
**public static final** String ***USER\_ID*** = **"user\_id"**;  
**public static final** String ***USER\_NAME*** = **"user\_name"**;  
**public static final** String ***USER\_PASSWORD*** = **"user\_password"**;  
**public static final** String ***USER\_CODE\_TEACHER*** = **"user\_code\_teacher"**;  
**public static final** String ***USER\_EMAIL*** = **"user\_email"**;  
**public static final** String ***USER\_FULL\_NAME*** = **"user\_full\_name"**;  
**public static final** String ***USER\_GENDER*** = **"user\_gender"**;  
**public static final** String ***USER\_DATE\_OF\_BIRTH*** = **"user\_dob"**;  
**public static final** String ***USER\_ADDRESS*** = **"user\_address"**;  
**public static final** String ***USER\_TYPE*** = **"user\_type"**;

**public static final** String ***CREATE\_TBL\_USER*** = String.*format*(**""** +  
 **"create table if not exists %s ("** +  
 **" %s integer primary key autoincrement,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s integer)"**,  
 ***TBL\_USER***, ***USER\_ID***, ***USER\_NAME***, ***USER\_PASSWORD***, ***USER\_CODE\_TEACHER***, ***USER\_EMAIL***, ***USER\_FULL\_NAME***,  
 ***USER\_GENDER***, ***USER\_DATE\_OF\_BIRTH***, ***USER\_ADDRESS***, ***USER\_TYPE***);  
**private static** SqLiteHelper *sqliteHelper* = **null**;

1. Lớp Lab: gồm 5 thuộc tính là: lab\_id, lab\_name, lab\_area, lab\_floor, lab\_note. Tương ứng với các thuộc tính này, ta tạo bảng TBL\_LAB tương ứng trog SQLite:

**public static final** String ***TBL\_LAB*** = **"lab"**;  
**public static final** String ***LAB\_ID***= **"lab\_id"**;  
**public static final** String ***LAB\_NAME*** = **"lab\_name"**;  
**public static final** String ***LAB\_AREA*** = **"lab\_area"**;  
**public static final** String ***LAB\_FLOOR*** = **"lab\_floor"**;  
**public static final** String ***LAB\_NOTE*** = **"lab\_note"**;

**public static final** String ***CREATE\_TBL\_LAB*** = String.*format*(**""** +  
 **"create table if not exists %s ("** +  
 **" %s integer primary key autoincrement,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text)"**,

***TBL\_LAB***, ***LAB\_ID***, ***LAB\_NAME***, ***LAB\_AREA***,***LAB\_FLOOR***, ***LAB\_NOTE***);

1. Lớp RegisterLab: gồm 7 thuộc tính là: labID, registerID, userID, session, time, termID, replaced. Tương ứng với các thuộc tính này, ta tạo bảng TBL\_REGISTERLAB tương ứng trog SQLite:

*// tbl RegisterLab***public static final** String ***TBL\_REGISTERLAB*** = **"registerlab"**;  
**public static final** String ***REGISTERLAB\_ID*** = **"registerlab\_id"**;  
*//public static final String USER\_ID = "user\_id";***public static final** String ***REGISTERLAB\_SESSION*** = **"registerlab\_session"**;  
**public static final** String ***REGISTERLAB\_TIME*** = **"registerlab\_time"**;  
*//public static final String TERM\_ID = "term\_id";***public static final** String ***REGISTERLAB\_REPLACED*** = **"registerlab\_replaced"**;

**public static final** String ***CREATE\_TBL\_REGISTERLAB*** =  
 **"CREATE TABLE IF NOT EXISTS "** + ***TBL\_REGISTERLAB*** + **"("** +  
 ***REGISTERLAB\_ID*** + **" INTEGER PRIMARY KEY AUTOINCREMENT,"** +  
 ***LAB\_ID*** + **" INTEGER,"** +  
 ***USER\_ID*** + **" INTEGER, "** +  
 ***REGISTERLAB\_SESSION*** + **" TEXT,"** +  
 ***REGISTERLAB\_TIME*** + **" TEXT,"** +  
 ***TERM\_ID*** + **" INTEGER,"** +  
 ***REGISTERLAB\_REPLACED*** + **" TEXT)"**;

1. Lớp Subject: gồm 3 thuộc tính là: Index, subjectID, subjectName. Tương ứng với các thuộc tính này, ta tạo bảng TBL\_SUBJECT tương ứng trog SQLite:

**public static final** String ***TBL\_SUBJECT*** = **"subject"**;  
**public static final** String ***SUBJECT\_ID*** = **"subject\_id"**;  
**public static final** String ***SUBJECT\_NAME*** = **"subject\_name"**;

**public static final** String ***CREATE\_TBL\_SUBJECT*** =  
 **"CREATE TABLE IF NOT EXISTS "** + ***TBL\_SUBJECT*** + **"("** +  
 ***SUBJECT\_ID*** + **" INTEGER PRIMARY KEY AUTOINCREMENT,"** +  
 ***SUBJECT\_NAME*** + **" TEXT)"**;

1. Lớp Term: gồm 3 thuộc tính là: TermID, subjectID, TermName. Tương ứng với các thuộc tính này, ta tạo bảng TBL\_TERM tương ứng trog SQLite:

**public static final** String ***TBL\_TERM*** = **"term"**;  
**public static final** String ***TERM\_ID*** = **"term\_id"**;  
**public static final** String ***TERM\_NAME*** = **"term\_name"**;

**public static final** String ***CREATE\_TBL\_TERM*** =  
 **"CREATE TABLE IF NOT EXISTS "** + ***TBL\_TERM*** + **"("** +  
 ***TERM\_ID*** + **" INTEGER PRIMARY KEY AUTOINCREMENT,"** +  
 ***TERM\_NAME*** + **" TEXT,"** +  
 ***SUBJECT\_ID*** + **" INTEGER)"**;

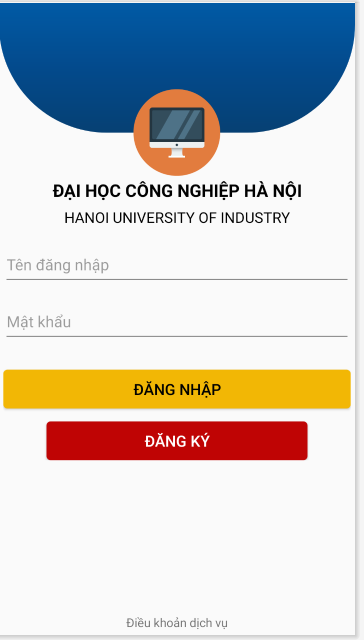
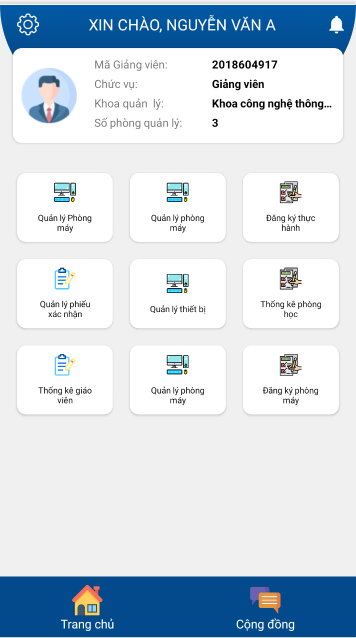
1. Lớp Device : gồm 7 thuộc tính là: maPhong, maTB, tenTb, loaitb, tinhTrang, ngaynhap, ghichu. Tương ứng với các thuộc tính này, ta tạo bảng TBL\_DEVICE tương ứng trog SQLite:

**public static final** String ***TBL\_DEVICE*** = **"device"**;  
**public static final** String ***DEVICE\_ID*** = **"id"**;  
**public static final** String ***DEVICE\_NAME*** = **"name"**;  
**public static final** String ***DEVICE\_TYPE*** = **"type"**;  
**public static final** String ***DEVICE\_LAB\_ID*** = **"lab\_id"**;  
**public static final** String ***DEVICE\_STATUS*** = **"status"**;  
**public static final** String ***DEVICE\_DATE*** = **"date"**;  
**public static final** String ***DEVICE\_NOTE*** = **"note"**;  
**public static final** String ***CREATE\_TBL\_DEVICE*** = String.*format*(**""** +  
 **"create table if not exists %s("** +  
 **" %s integer primary key autoincrement,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s integer not null,"** +  
 **" %s text,"** +  
 **" %s text,"** +  
 **" %s text)"**,  
 ***TBL\_DEVICE***, ***DEVICE\_ID***, ***DEVICE\_NAME***, ***DEVICE\_TYPE***, ***DEVICE\_LAB\_ID***, ***DEVICE\_STATUS***, ***DEVICE\_DATE***, ***DEVICE\_NOTE***);

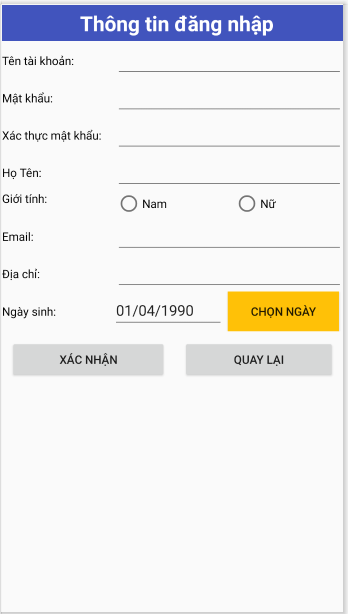
## Thực hiện bài toán

### Quản lý đăng nhập và người dùng (Phạm Minh Tuấn)

* **Màn hình thực hiện:**



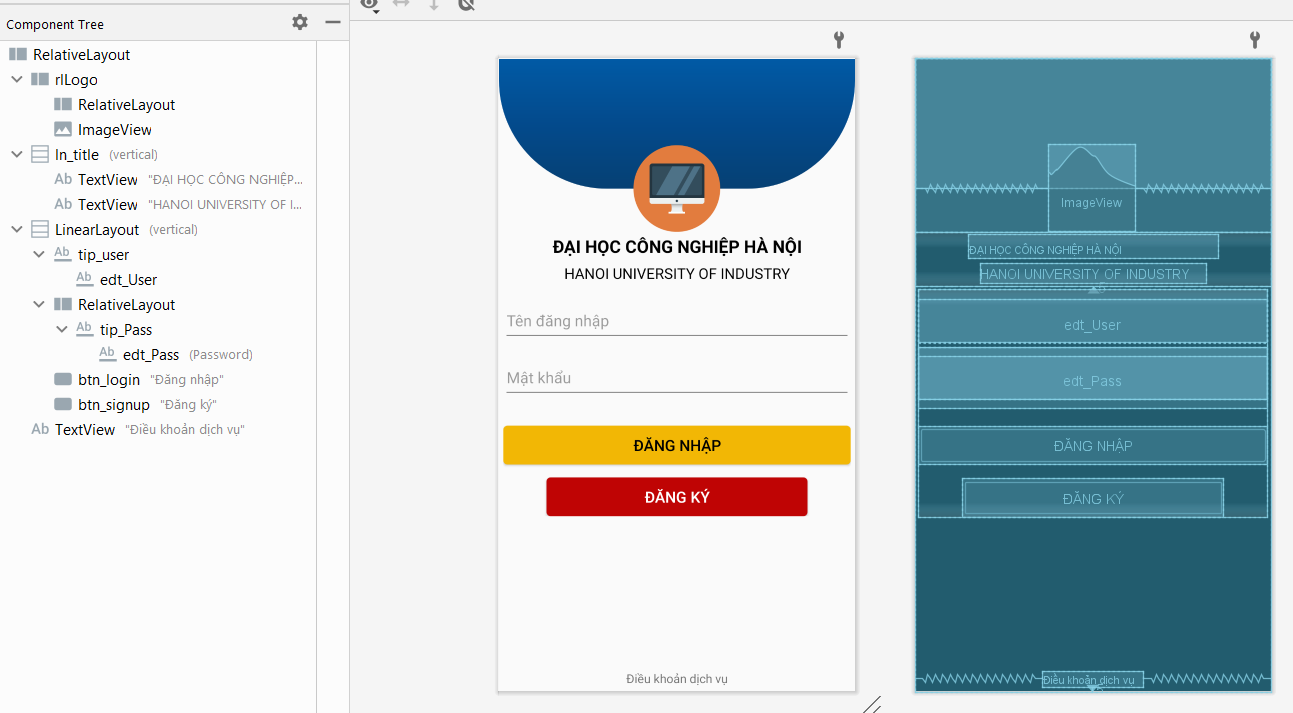
Hình 2‑17: Màn hình đăng nhập và màn hình chính



Hình 2‑18: Màn hình thông tin đăng nhập

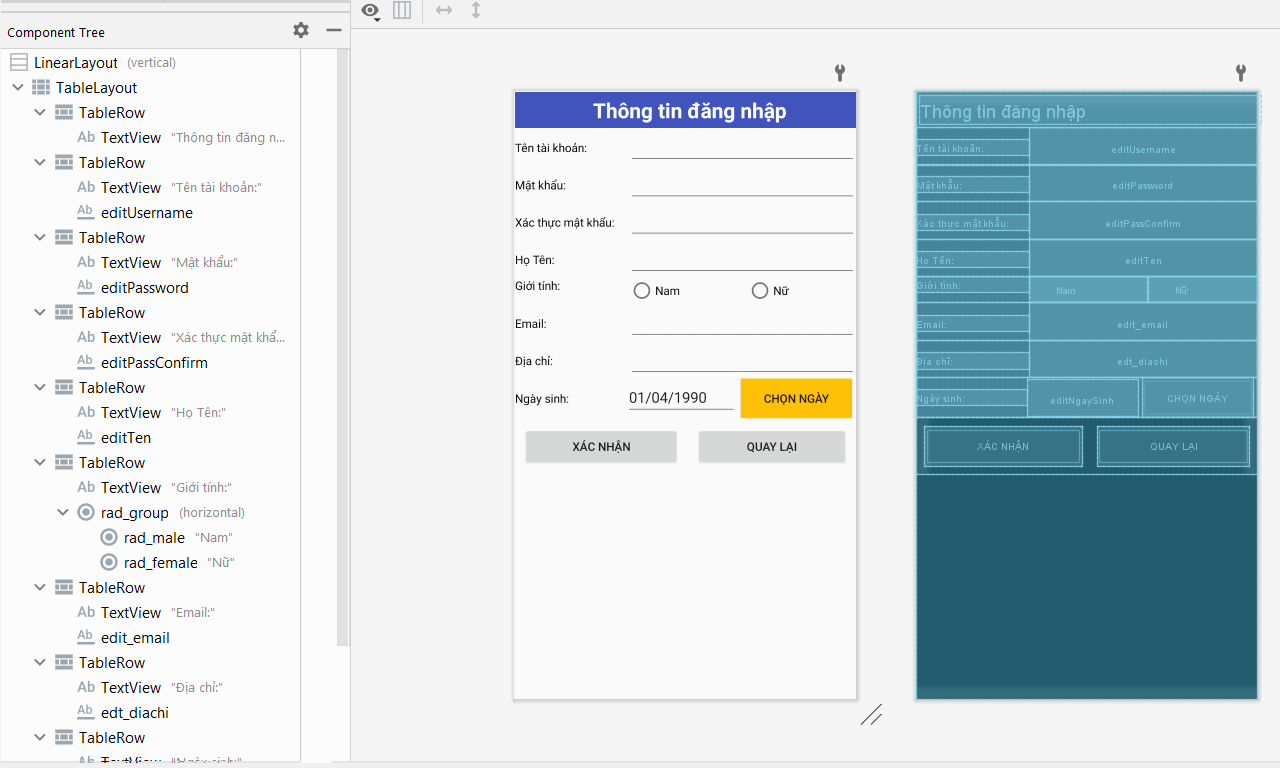
* **Nhiệm vụ:** Thực hiện chức năng đăng nhập vào hệ thống và sau khi đăng nhập thành công thì sẽ gọi đến MainActivity.java, hàm này hiển thị giao diện trang chủ của ứng dụng. Hoặc người dùng có thể đăng ký tài khoản
* **Thiết kế giao diện trên XML:**

**Layout đăng nhập:**

****

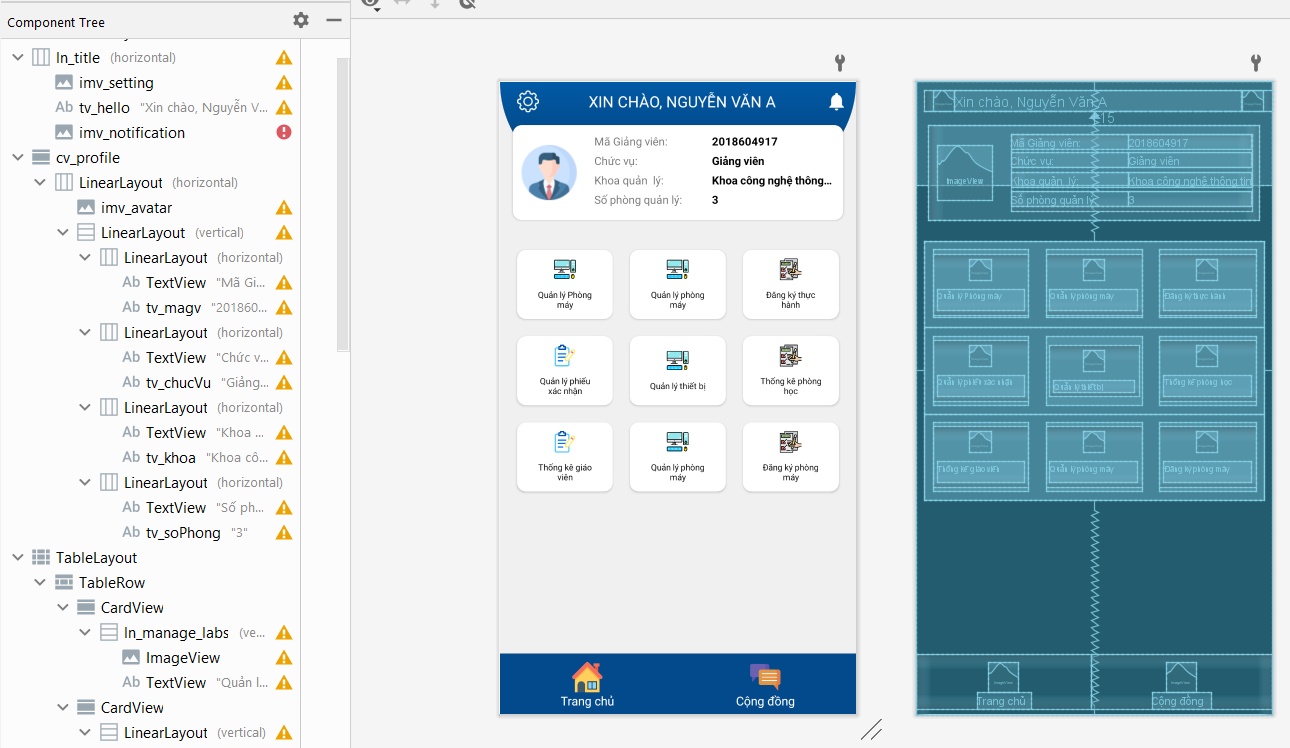
Hình 2‑19: Layout đăng nhập

**Layout đăng ký:**

****

Hình 2‑20: Layout đăng ký

**Layout màn hình chính:**

****

Hình 2‑21: Layout màn hình chính

* **Mã nguồn chính trong bài toán:**
* **Hàm đăng kí tài khoản ( SignUp.java):** nếu đăng kí thành công sẽ quay lại màn hình đăng nhập của hệ thóng.

**public void** addUser() {  
 String gender;  
 String name = **edt\_name**.getText()+**""**;  
 String username = **edt\_username**.getText()+**""**;  
 String password = **edt\_password**.getText()+**""**;  
 **if**(**radioGroup**.getCheckedRadioButtonId() == R.id.***rad\_male***) {  
 gender = **"Nam"**;  
 }  
 **else** gender = **"Nữ"**;  
 String email = **edt\_email**.getText()+**""**;  
 String address = **edt\_address**.getText()+**""**;  
 String birth = **edt\_birth**.getText()+**""**;  
 **selectedUser**.setUserName(username);  
 **selectedUser**.setPassword(password);  
 **selectedUser**.setFullName(name);  
 **selectedUser**.setGender(gender);  
 **selectedUser**.setDateOfBirth(birth);  
 **selectedUser**.setAddress(address);  
 **selectedUser**.setEmail(email);  
 **long** flag = **sqlHelper**.Insert\_User(**selectedUser**);  
 **if**(flag != -1) {  
 Toast.*makeText*(SignUp.**this**,**"Đăng ký thành công"**,Toast.***LENGTH\_SHORT***).show();  
 } **else** Toast.*makeText*(SignUp.**this**,**"Đăng ký không thành công"**,Toast.***LENGTH\_SHORT***).show();  
}

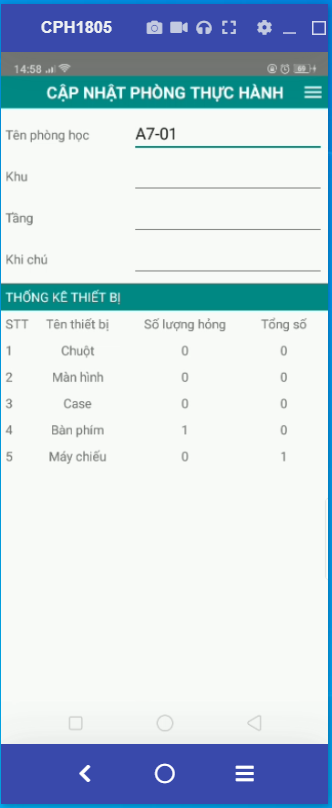
* **Hàm đăng Nhập ( LoginActivity):** nếu đăng nhập thành công màn hình sẽ chuyển vào trang chủ.

**public void** login() {  
 String user = **edt\_username**.getText()+**""**;  
 String pass = **edt\_password**.getText()+**""**;  
 **if**(user.equals(**""**) || pass.equals(**""**)) {  
 Toast.*makeText*(**this**,**"Tên đăng nhập và mật khẩu không được để trống"**,Toast.***LENGTH\_SHORT***).show();  
 }  
 **else if**(**null**!=checkUser(user,pass)) {  
 String userDB = checkUser(user,pass);  
 Intent intent = **new** Intent(LoginActivity.**this**,MainActivity.**class**);  
 intent.putExtra(**"user\_name"**,userDB);  
 startActivity(intent);  
 }  
 **else** {  
 Toast.*makeText*(**this**,**"Tên tài khoản hoặc mật khẩu không đúng"**,Toast.***LENGTH\_SHORT***).show();  
 **edt\_username**.setText(**""**);  
 **edt\_password**.setText(**""**);  
 **edt\_username**.requestFocus();  
 }  
}

### Quản lý phòng thực hành (Dương Thị Huyền)

* **Màn hình thực hiện:**

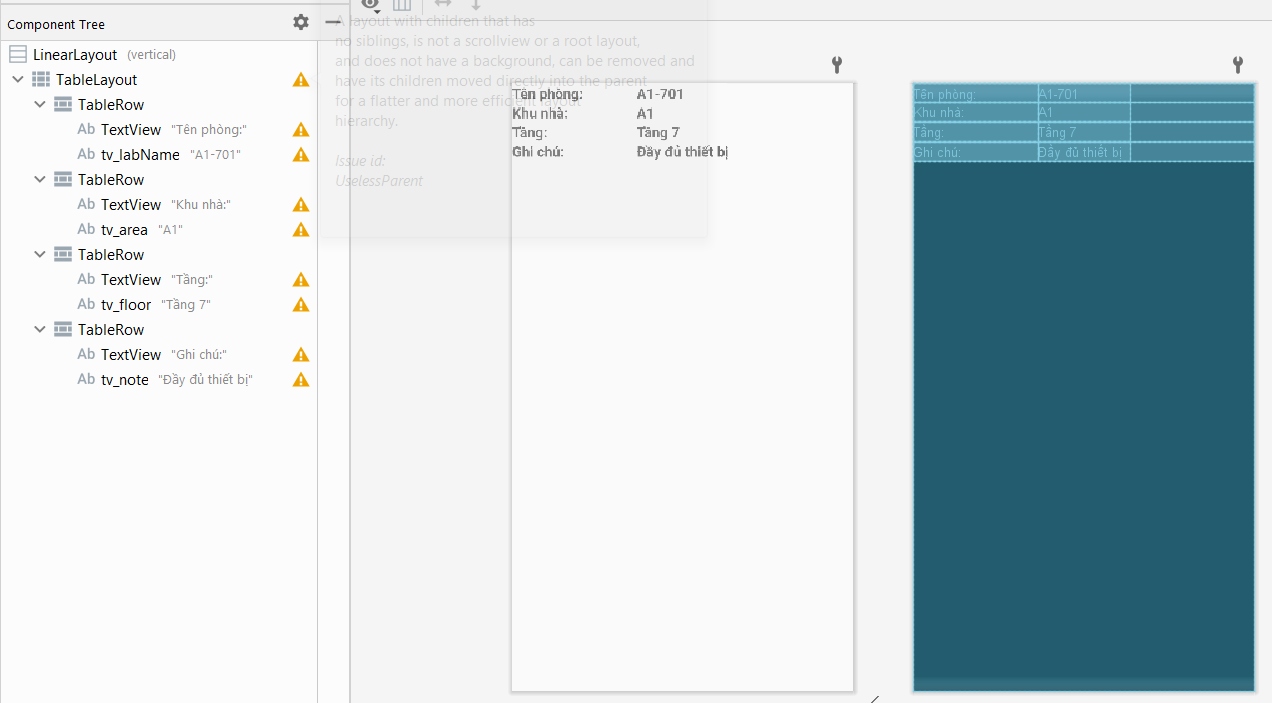
|  |  |
| --- | --- |
| Hình 2‑22: Quản lý danh sách phòng học | Hình 2‑23: Thêm mới phòng thực hành |

****

Hình 2‑24: Cập nhật phòng thực hành

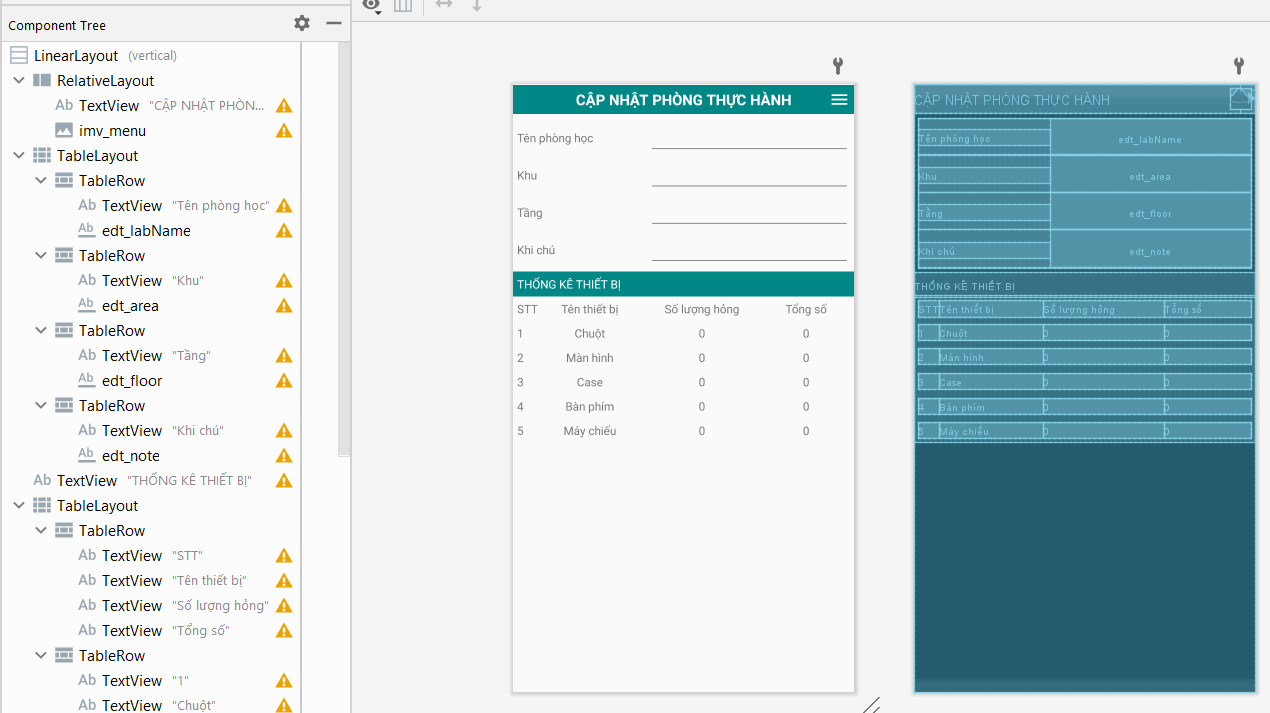
* **Nhiệm vụ:** Thực hiện chức năng Quản lý phòng thực hành: thêm phòng, sửa thông tin của phòng và xóa thông tin phòng và cập nhật số lượng của thiết bị trong phòng.
* **Thiết kế giao diện trên XML:**

**Layout chính của chức năng:**

****

Hình 2‑25: Layout chính của chức năng quản lý phòng thực hành

**Layout thêm phòng, sửa phòng:**

****

Hình 2‑26: Layout thêm phòng, sửa phòng

* **Mã nguồn chính trong bài toán:**
* **Hàm thêm phòng (AddLabActivity.java):** thêm thành công hệ thống sẽ hiển thị phòng vừa thêm sang list trong màn hình chính của chức năng.

**private void** processAdd() {  
 Lab lab = **new** Lab();  
 String labName = **edt\_labName**.getText()+**""**;  
 String area = **edt\_area**.getText()+**""**;  
 String floor = **edt\_floor**.getText()+**""**;  
 String note = **edt\_note**.getText()+**""**;  
  
 lab.setLab\_Name(labName);  
 lab.setTang(floor);  
 lab.setKhuNha(area);  
 lab.setGhiChu(note);  
 **long** result = **sqLiteHelper**.Insert\_Lab(lab);  
 **if**(result!=-1)  
 {  
 Toast.*makeText*(AddLabActivity.**this**, **"Thêm thành công!"**, Toast.***LENGTH\_SHORT***).show();  
 finish();  
 }**else**{  
 Toast.*makeText*(AddLabActivity.**this**, **"Thêm thất bại!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
}

* **Insert\_lab:** cập nhật và lưu lại thông tin trong Sql

**public long** Insert\_Lab(Lab lab){  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(***LAB\_NAME***, lab.getLab\_Name());  
 contentValues.put(***LAB\_AREA***, (lab.getKhuNha()!=**null**?lab.getKhuNha():**""**));  
 contentValues.put(***LAB\_FLOOR***, (lab.getTang()!=**null**?lab.getTang():**""**));  
 contentValues.put(***LAB\_NOTE***, (lab.getGhiChu()!= **null**?lab.getGhiChu():**""**));  
  
 **long** rowID = db.insert(***TBL\_LAB***, **null**, contentValues);  
 db.close();  
 **return** rowID;  
}

* **Hàm lấy thông tin số lượng của các thiết bị từ lớp Device**

**deviceList** = **sqLiteHelper**.GetListDeviceByLabID(**lab**.getLab\_ID());  
  
 **int** mouse = 0, mouse\_br = 0, monitors = 0, monitors\_br = 0, cases = 0, cases\_br = 0;  
 **int** kbs = 0, kb\_br = 0, prjs = 0, prjs\_br = 0;  
 **for**(Device device: **deviceList**){  
 **switch** (device.getLoaitb()){  
 **case "Chuột"**:  
 mouse++;  
 **if**(device.getTinhtrang().equals(**"Hỏng"**))  
 mouse\_br++;  
 **break**;  
 **case "Màn hình"**:  
 monitors++;  
 **if**(device.getTinhtrang().equals(**"Hỏng"**))  
 monitors\_br++;  
 **break**;  
 **case "Case"**:  
 cases++;  
 **if**(device.getTinhtrang().equals(**"Hỏng"**))  
 cases\_br++;  
 **break**;  
 **case "Bàn phím"**:  
 kbs++;  
 **if**(device.getTinhtrang().equals(**"Hỏng"**))  
 kb\_br++;  
 **break**;  
 **case "Máy chiếu"**:  
 prjs++;  
 **if**(device.getTinhtrang().equals(**"Hỏng"**))  
 prjs\_br++;  
 **break**;  
 }  
 }  
 **tv\_cases**.setText(cases+**""**);  
 **tv\_cases\_broken**.setText(cases\_br+**""**);  
  
 **tv\_keyboards**.setText(kbs+**""**);  
 **tv\_keyboards\_broken**.setText(kb\_br+**""**);  
  
 **tv\_Monitor\_broken**.setText(monitors\_br+**""**);  
 **tv\_Monitors**.setText(monitors+**""**);  
  
 **tv\_mouse**.setText(mouse+**""**);  
 **tv\_mouse\_broken**.setText(mouse\_br+**""**);  
  
 **tv\_projectors**.setText(prjs+**""**);  
 **tv\_projectors\_broken**.setText(prjs\_br+**""**);  
  
  
}

* **Hàm sửa thông tin phòng (EditLabActivity.java):** sửa thành công hệ thống sẽ cập nhật lại dữ liệu sang list chính.

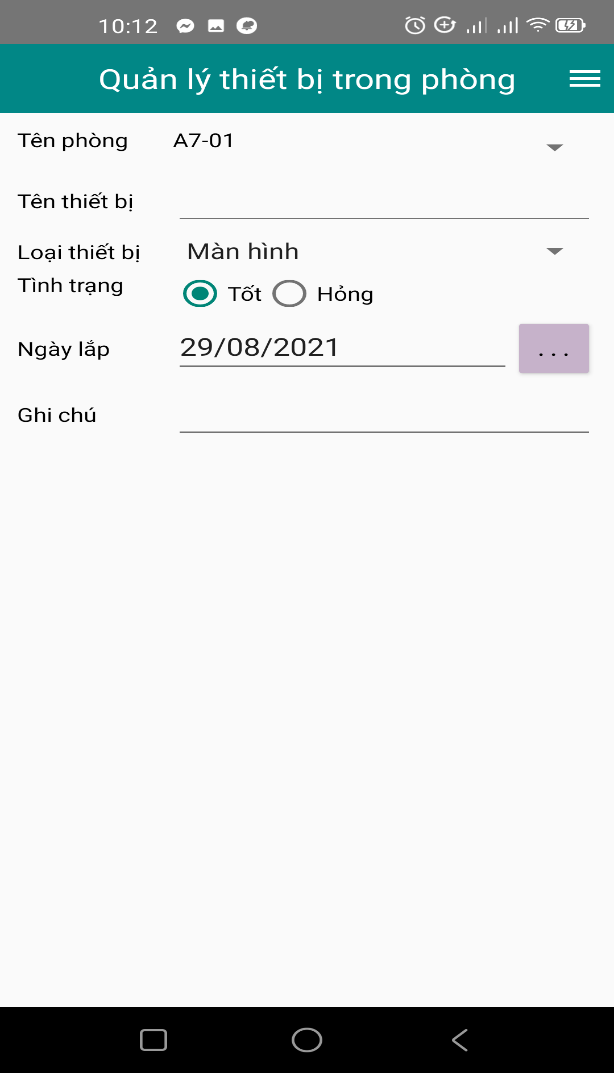
**private void** processEdit() {  
 Lab lab = **new** Lab();  
 String labName = **edt\_labName**.getText()+**""**;  
 String area = **edt\_area**.getText()+**""**;  
 String floor = **edt\_floor**.getText()+**""**;  
 String note = **edt\_note**.getText()+**""**;  
  
 lab.setLab\_Name(labName);  
 lab.setTang(floor);  
 lab.setKhuNha(area);  
 lab.setGhiChu(note);  
 **long** result = **sqLiteHelper**.Update\_Labs(lab);  
 **if**(result!=-1)  
 {  
 Toast.*makeText*(EditLabActivity.**this**, **"Cập nhật thành công!"**, Toast.***LENGTH\_SHORT***).show();  
 finish();  
 }**else**{  
 Toast.*makeText*(EditLabActivity.**this**, **"Cập nhật thất bại!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
}

* **Update\_lab :cập nhật lại thông tin và lưu lại thông tin trong Sql.**

**public long** Update\_Labs(Lab lab){  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
  
 contentValues.put(***LAB\_NAME***, lab.getLab\_Name());  
 contentValues.put(***LAB\_AREA***, lab.getKhuNha());  
 contentValues.put(***LAB\_FLOOR***, lab.getTang());  
 contentValues.put(***LAB\_NOTE***, lab.getGhiChu());  
  
 String id = lab.getLab\_ID()+**""**;  
 **int** rowID = db.update(***TBL\_LAB***, contentValues, **"id=?"**, **new** String[]{id});  
 db.close();  
 **return** rowID;  
}

### Quản lý thiết bị (Lê Thị Hảo)

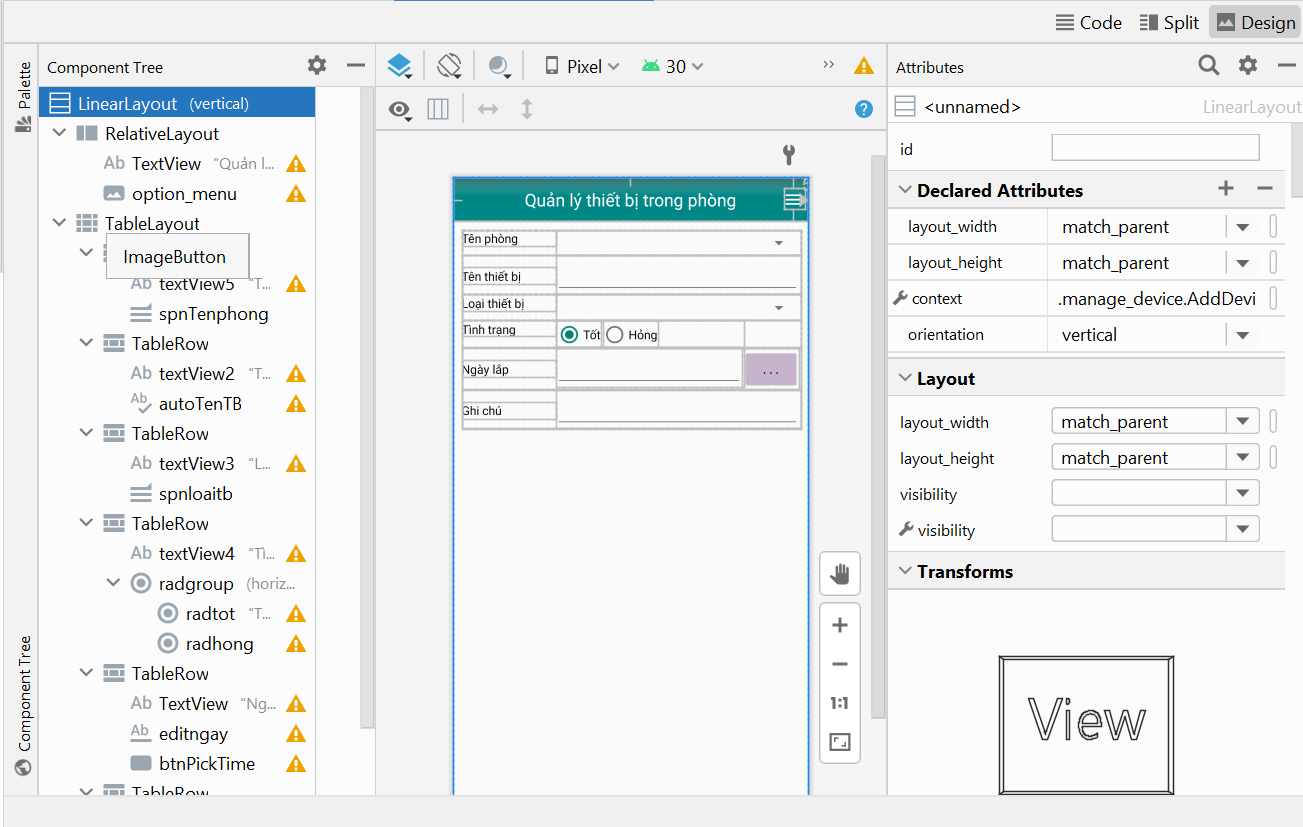
* **Màn hình thực hiện:**

** **

|  |  |
| --- | --- |
| Hình 2‑27: Danh sách thiết bị | Hình 2‑28: Quản lý thiết bị trong phòng |

* **Nhiệm vụ:** cho phép người dùng thêm thông tin của thiết bị, sửa thông tin của thiết bị và cập nhật tình trạng (tốt, hỏng) của thiết bị trong phòng học.
* **Thiết kế giao diện trên XML:**

**Layout thêm, sửa thông tin thiết bị.**

****

Hình 2‑29: Layout thêm, sửa thông tin thiết bị

* **Mã nguồn chính trong bài toán:**
* **Hàm thêm thiết bị cho chức năng:**

**private void** processAdd(){  
 Lab lab = (Lab) **spnTenphong**.getSelectedItem();  
 String tenTb = **autoTenTb**.getText()+**""**;  
 String loaitb = **spnLoaiTb**.getSelectedItem().toString();  
 String tinhTrang = **""**;  
 **if**(**radTot**.isChecked()){  
 tinhTrang=**"Tốt"**;  
 }  
 **else if**(**radHong**.isChecked()){  
 tinhTrang = **"Hỏng"**;  
 }  
 Pattern special = Pattern.*compile*(**"[!@#$,?.\*%&\*()\_+=|<>?{}\\[\\]~-]"**);  
 Matcher hasSpecial = special.matcher(tenTb);  
 String ngaylap = **editNgayNhap**.getText().toString().trim();  
 String ghiChu = **editGhiChu**.getText().toString().trim();  
 **if**(tenTb.isEmpty()){  
 Toast.*makeText*(getBaseContext(), **"Vui lòng nhập tên thiết bị thông tin"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else if**(hasSpecial.find()==**true**){  
 Toast.*makeText*(getBaseContext(), **"Tên thiết bị không được chứa ký tự đặc biệt"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else**{  
 Device dv = **new** Device(0,tenTb,loaitb,lab.getLab\_ID(),tinhTrang,ngaylap,ghiChu);  
 processAutoTenTb(tenTb);  
 **long** result = **sqliteHelper**.Insert\_Device(dv);  
 **if**(result==-1)  
 {  
 Toast.*makeText*(AddDeviceActivity.**this**, **"Thêm thất bại!"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 **else** Toast.*makeText*(AddDeviceActivity.**this**, **"Thêm thành công!"**, Toast.***LENGTH\_SHORT***).show();  
 clearText();  
 }  
}

* **Cập nhật thông tin vào sql:**

**public long** Insert\_Device(Device device){  
 SQLiteDatabase database = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(***DEVICE\_NAME***, device.getTenTb());  
 contentValues.put(***DEVICE\_TYPE***, device.getLoaitb());  
 contentValues.put(***DEVICE\_LAB\_ID***, device.getMaPhong());  
 contentValues.put(***DEVICE\_STATUS***, device.getTinhtrang());  
 contentValues.put(***DEVICE\_DATE***, device.getNgaynhap());  
 contentValues.put(***DEVICE\_NOTE***, device.getGhichu());  
  
  
 **long** rowID = database.insert(***TBL\_DEVICE***, **null**, contentValues);  
 database.close();  
 **return** rowID;  
}

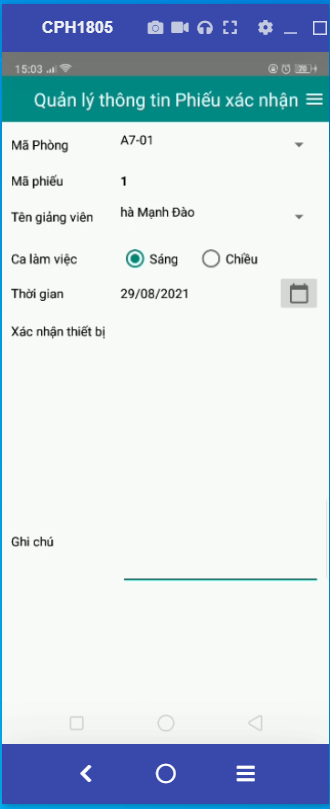
* **Sửa thông tin thiết bị:**

**void** processEdit() {  
 Lab lab = (Lab) **spnTenphong**.getSelectedItem();  
 Device dv = **new** Device();  
 String tinhTrang = **""**;  
 **if** (**radTot**.isChecked()) {  
 tinhTrang = **"Tốt"**;  
 } **else if** (**radHong**.isChecked()) {  
 tinhTrang = **"Hỏng"**;  
 }  
 **int** maTB = **device**.getMaTb();  
 Pattern special = Pattern.*compile*(**"[!@#$,?.\*%&\*()\_+=|<>?{}\\[\\]~-]"**);  
 Matcher hasSpecial = special.matcher(**autoTenTb**.getText().toString());  
 **if**(**autoTenTb**.getText().toString().isEmpty()){  
 Toast.*makeText*(getBaseContext(), **"Vui lòng nhập tên thiết bị muốn sửa"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else if**(hasSpecial.find()==**true**){  
 Toast.*makeText*(getBaseContext(), **"Tên thiết bị không được chứa ký tự đặc biệt"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else**{  
 dv = **new** Device(maTB, **autoTenTb**.getText().toString(),  
 **spnLoaiTb**.getSelectedItem().toString(), lab.getLab\_ID(), tinhTrang, **editNgayNhap**.getText().toString(), **editGhiChu**.getText().toString());  
 processAutoTenTb(**autoTenTb**.getText().toString());  
 **long** result = **sqliteHelper**.Update\_Device\_By\_ID(dv);  
 **if** (result <= 0) {  
 Toast.*makeText*(EditDevice.**this**, **"Cập nhật thất bại!"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 Toast.*makeText*(EditDevice.**this**, **"Cập nhật thành công!"**, Toast.***LENGTH\_SHORT***).show();  
 finish();   
 }  
}

* **Cập nhật lại thông tin bên Sql:**

**public long** Update\_Device\_By\_ID(Device device){  
 SQLiteDatabase database = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(***DEVICE\_NAME***, device.getTenTb());  
 contentValues.put(***DEVICE\_TYPE***, device.getLoaitb());  
 contentValues.put(***DEVICE\_LAB\_ID***, device.getMaPhong());  
 contentValues.put(***DEVICE\_STATUS***, device.getTinhtrang());  
 contentValues.put(***DEVICE\_DATE***, device.getNgaynhap());  
 contentValues.put(***DEVICE\_NOTE***, device.getGhichu());  
 **int** rowID = database.update(***TBL\_DEVICE***, contentValues, **"id=?"**, **new** String[]{device.getMaTb()+**""**});  
 database.close();  
 **return** rowID;  
}

### Quản Lý phiếu xác nhận (Trần Quốc Đạt)

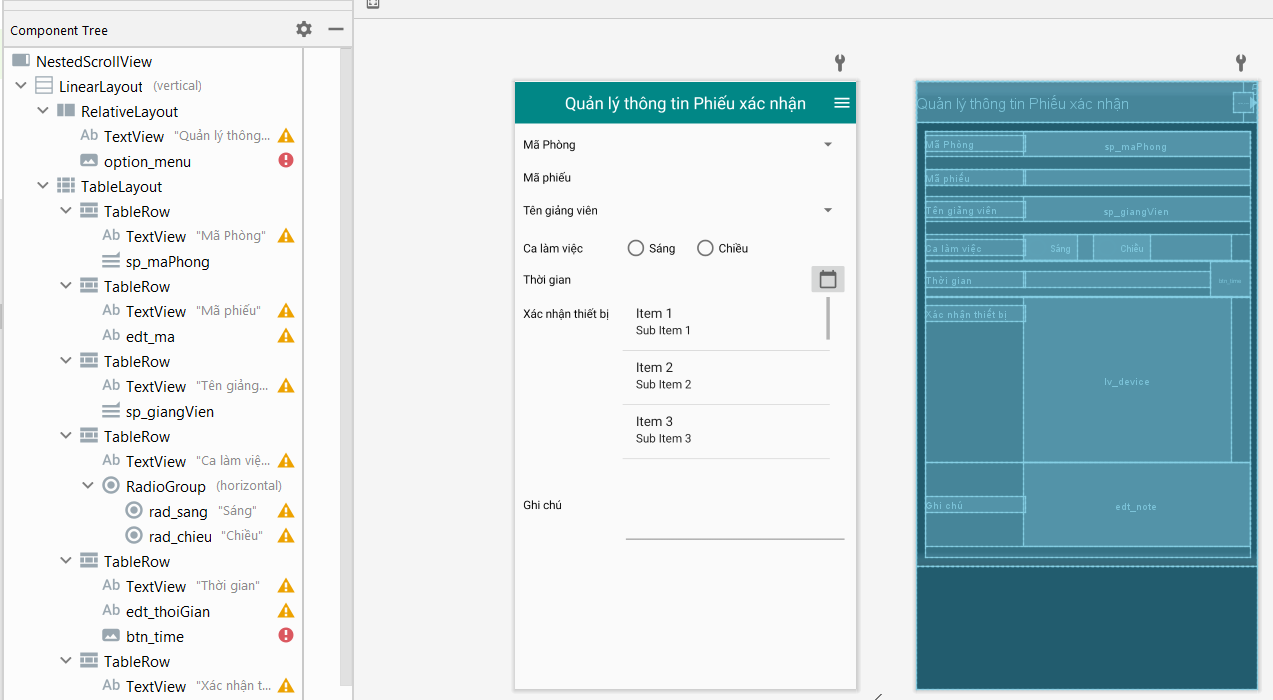
* **Màn hình thực hiện:**

****

|  |  |
| --- | --- |
| Hình 2‑30: Danh sách phiếu xác nhận | Hình 2‑31: Quản lý thông tin phiếu xác nhận |

* **Nhiệm vụ:** người dùng cập nhật và quản lý thông tin của phòng thực hành sau khi học xong.
* **Thiết kế giao diện trên XML:**

**Layout sửa, thêm thông tin của phiếu xác nhận:**

****

Hình 2‑32: Layout sửa, thêm thông tin của phiếu xác nhận

* **Mã nguồn chính trong bài toán:**
* **Hàm thêm phiếu xác nhận:**

**private void** processAdd() {  
 Lab lab = (Lab) **sp\_lab**.getSelectedItem();  
 User tenGV =(User) **sp\_giangVien**.getSelectedItem();  
  
 String tenPhong = lab.getLab\_Name() +**""**;  
 **boolean** shift = **true**;  
 **if** (**rad\_chieu**.isChecked())  
 shift = **false**;  
 String time = **tv\_Time**.getText() + **""**;  
 **if**(**note\_device** ==**""**)  
 **note** = **"Đầy đủ thiết bị"**;  
 **else note** = **"Thiếu "** + **note\_device**;  
  
 String note\_final = **edt\_note**.getText()+**"("** + **note**+**")"**;  
  
 VerifyReport verifyReport = **new** VerifyReport(0, lab.getLab\_ID(),tenGV.getId\_user(), shift, time, note\_final);  
 **long** result = **sqliteHelper**.Insert\_Verify(verifyReport);  
 **if**(result==-1)  
 {  
 Toast.*makeText*(AddNewVerify.**this**, **"Thêm thất bại!"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 Toast.*makeText*(AddNewVerify.**this**, **"Thêm thành công!"**, Toast.***LENGTH\_SHORT***).show();  
 clearText();  
 finish();  
}

* **Cập nhật thông tin trong Sql:**

**public long** Insert\_Verify(VerifyReport report){  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
  
 contentValues.put(***VERIFY\_LAB\_ID***, report.getMaPhong());  
 contentValues.put(***VERIFY\_USER\_ID***, report.getUser\_id());  
 contentValues.put(***VERIFY\_SHIFT***, report.getShift()?1:0);  
 contentValues.put(***VERIFY\_TIME***, report.getTime());  
 contentValues.put(***VERIFY\_NOTE***, report.getNote());  
  
  
  
 **long** rowID = db.insert(***TBL\_VERIFY***, **null**, contentValues);  
 db.close();  
 **return** rowID;  
}

* **Hàm sửa thông tin:**

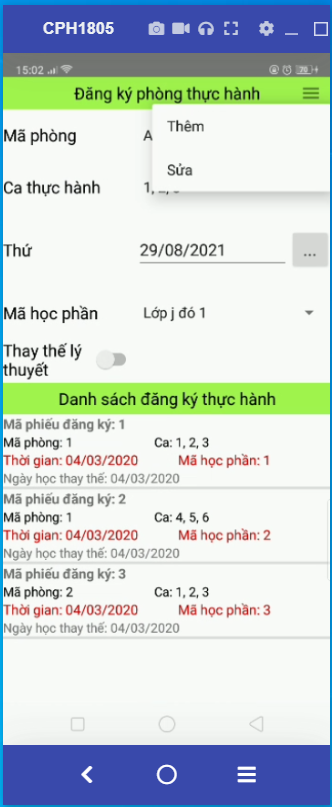
**private void** processEdit() {  
 Lab lab = (Lab) **sp\_lab**.getSelectedItem();  
 User tenGV =(User) **sp\_giangVien**.getSelectedItem();  
 **int** id = Integer.*parseInt*(**tv\_maPhieu**.getText() +**""**);  
  
 String tenPhong = lab.getLab\_Name() +**""**;  
 **boolean** shift = **true**;  
 **if** (**rad\_chieu**.isChecked())  
 shift = **false**;  
 String time = **tv\_Time**.getText() + **""**;  
 **if**(**note\_device** ==**""**)  
 **note** = **"Đầy đủ thiết bị"**;  
 **else note** = **"Thiếu "** + **note\_device**;  
  
 String note\_final = **edt\_note**.getText()+**"("** + **note**+**")"**;  
  
 VerifyReport verifyReport = **new** VerifyReport(id, lab.getLab\_ID(),tenGV.getId\_user(), shift, time, note\_final);  
 **long** result = **sqliteHelper**.Update\_Verify\_By\_ID(verifyReport);  
 **if**(result<=0)  
 {  
 Toast.*makeText*(EditVerify.**this**, **"Cập nhật thất bại!"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 Toast.*makeText*(EditVerify.**this**, **"Cập nhật thành công!"**, Toast.***LENGTH\_SHORT***).show();  
 finish();  
}

* **Cập nhật và lưu lại thông tin vào Sql**

**public int** Update\_Verify\_By\_ID(VerifyReport report){  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
  
 contentValues.put(***VERIFY\_LAB\_ID***, report.getMaPhong());  
 contentValues.put(***VERIFY\_USER\_ID***, report.getUser\_id());  
 contentValues.put(***VERIFY\_SHIFT***, report.getShift()?1:0);  
 contentValues.put(***VERIFY\_TIME***, report.getTime());  
 contentValues.put(***VERIFY\_NOTE***, report.getNote());  
 String id = report.getId()+**""**;  
 **int** rowID = db.update(***TBL\_VERIFY***, contentValues, **"id=?"**, **new** String[]{id});  
 db.close();  
 **return** rowID;  
}

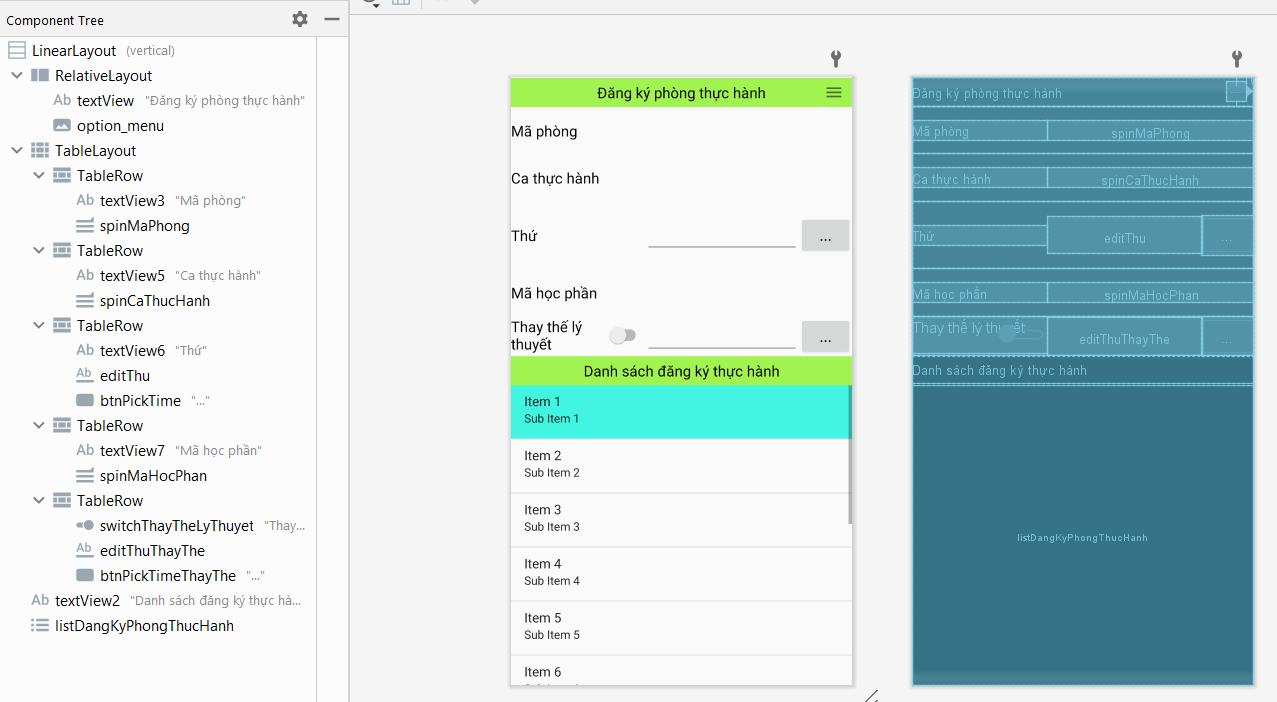
### Quản lý phiếu đăng ký (Phạm Quang Trung)

* **Màn hình thực hiện:**

****

Hình 2‑33: Đăng ký phòng thực hành

* **Nhiệm vụ:** cho phép người dùng đăng kí lịch thực hành , sửa hoặc xóa lịch theo mong muốn.
* **Thiết kế giao diện trên XML:**

****

Hình 2‑34: Giao diện trên XML của phiếu đăng ký

* **Mã nguồn chính trong bài toán:**
* **Hàm đăng ký phòng thực hành :**

**void** AddRegiterLab() {  
 RegisterLab registerLabInfo = **new** RegisterLab();  
 *//registerLabInfo.setRegisterID();* SharedPreferences share = getSharedPreferences(**"username"**,***MODE\_PRIVATE***);  
 **int** userID = Integer.*parseInt*(Objects.*requireNonNull*(share.getString(**"user\_id"**, **"1"**)));  
 registerLabInfo.setUserID(userID); *//* ***TODO: get user static*** registerLabInfo.setSession(**spinSession**.getSelectedItem() + **""**);  
  
 registerLabInfo.setTime(**editTime**.getText() + **""**);  
  
 registerLabInfo.setTermID(**termsID**.get(**spinTerm**.getSelectedItemPosition()));  
 registerLabInfo.setLabID(**labsID**.get(**spinLab**.getSelectedItemPosition()));  
 **if** (**switchReplaced**.isChecked()){  
 registerLabInfo.setReplaced(**editReplaced**.getText()+**""**);  
 }  
 **else**{  
 registerLabInfo.setReplaced(**null**);  
 }  
 **boolean** result = SqLiteHelper.*getInstance*(RegisterLabActivity.**this**).Insert\_RegisterLab(registerLabInfo);  
 **if** (result){  
 Toast.*makeText*(RegisterLabActivity.**this**, **"Thêm thành công"**, Toast.***LENGTH\_SHORT***).show();  
 **curIndexSelected** = -1;  
 }  
 **else**{  
 Toast.*makeText*(RegisterLabActivity.**this**, **"Thêm không thành công"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 refreshDataForListview();  
}

* **Cập nhật và lưu thông tin trong sql:**

**public long** Insert\_Subject(Subject subject) {  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
  
 contentValues.put(***SUBJECT\_ID***, subject.getSubjectID());  
 contentValues.put(***SUBJECT\_NAME***, subject.getSubjectName());  
  
 **long** rowID = db.insert(***TBL\_SUBJECT***, **null**, contentValues);  
 db.close();  
 **return** rowID;  
}  
  
**public long** Insert\_Term(Term term) {  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
  
 contentValues.put(***TERM\_ID***, term.getTermID());  
 contentValues.put(***TERM\_NAME***, term.getTermName());  
 contentValues.put(***SUBJECT\_ID***, term.getSubjectID());  
  
 **long** rowID = db.insert(***TBL\_TERM***, **null**, contentValues);  
 db.close();  
 **return** rowID;  
}

* **Hàm Sửa lịch đăng ký:**

**void** UpdateRegisterLab(){  
 **if** (**curIndexSelected** >= 0){  
 RegisterLab registerLabInfo = **new** RegisterLab();  
  
 registerLabInfo.setRegisterID(**adapterRegisterLab**.getItem(**curIndexSelected**).getRegisterID());  
 registerLabInfo.setLabID(**labsID**.get(**spinLab**.getSelectedItemPosition()));  
 SharedPreferences share = getSharedPreferences(**"user"**,***MODE\_PRIVATE***);  
 **int** userID = share.getInt(**"user\_id"**, 1);  
 registerLabInfo.setUserID(userID);  
 registerLabInfo.setSession(**spinSession**.getSelectedItem() + **""**);  
  
 registerLabInfo.setTime(**editTime**.getText() + **""**);  
  
 registerLabInfo.setTermID(**termsID**.get(**spinTerm**.getSelectedItemPosition()));  
 **if** (**switchReplaced**.isChecked()){  
 registerLabInfo.setReplaced(**editReplaced**.getText() + **""**);  
 }  
 **else**{  
 registerLabInfo.setReplaced(**null**);  
 }  
  
 **boolean** result = SqLiteHelper.*getInstance*(RegisterLabActivity.**this**).Update\_RegisterLab(registerLabInfo);  
 **if** (result){  
 Toast.*makeText*(RegisterLabActivity.**this**, **"Sửa thành công"**, Toast.***LENGTH\_SHORT***).show();  
 **curIndexSelected** = -1;  
 }  
 **else**{  
 Toast.*makeText*(RegisterLabActivity.**this**, **"Sửa không thành công"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 refreshDataForListview();  
 }  
}

# KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM

## Nội dung đã thực hiện

Thông qua việc thực hiện bài tập lớn với đề tài “Quản lý phòng thực hành”, chúng em đã có được những kiến thức và kỹ năng hữu ích cho quá trình học tập và áp dụng trong công việc.

* Về kiến thức:
* Biết sử dụng các cú pháp xml để xây dựng giao diện cho chương trình
* Nắm được các kiến thức cơ bản về Activity và vòng đời của nó
* Biết sử dụng Adapter để chuyển đổi dữ liệu hiển thị ra màn hình
* Sử dụng Intent hay Bundle để lưu trữ dữ liệu tạm thời và truyền từ Activity này sang Activity khác
* Củng cố lại các kiến thức về các câu lệnh SQL từ đó áp dụng để lưu trữ DL của chương trình với SQLite
* Về kỹ năng:
* Nâng cao kỹ năng lập trình
* Biết thêm về 1 số phím tắt hữu ích tgrong Android Studio như: Ctrl+tab, Ctrl+D, Alt+enter, Alt+insert, Ctrl+Alt+L, … Các phím tắt này rất tiện dụng, giúp rút ngắn thời gian lập trình
* Tăng cường kỹ năng làm việc nhóm

Những chuẩn đầu ra đạt được trong quá trình thực hiện bài tập lớn:

* Cài đặt được ứng dụng cơ bản trên thiết bị di động Android
* Xây dựng được bản mô tả yêu cầu và các chưc năng cần thiết của ứng dụng cần xây dựng

Ngoài những kiến thức và kỹ năng có được sau khi hoàn thành bài tập lớn, chúng em còn rút ra được bài học kinh nghiệm. Đó là tính tự giác để hoàn thành công việc chung của cả nhóm, cần phải lắng nghe, tiếp thu những góp ý của mọi người để công việc đạt kết quả cao hơn. Ngoài ra, sau khi hoàn thành xong bài tập lớn chúng em cảm thấy kiến thức và khả năng của bản thân còn khá hạn chế, cần tự học, tìm tòi nhiều hơn để phát triển bản thân

## Hướng phát triển

Tính khả thi của chủ đề nghiên cứu: bài toán này tương đối khả thi, nó khá thiết thực và phù hợp với nhà trường. Nó giúp người dùng trong việc quản lý các lịch thực hành, thiết bị, … Các yêu cầu chính được đưa ra khá phù hợp với thực tế.

Bài toán đặt ra còn chưa được hoàn mĩ, mọi chức năng còn thô sơ, chưa có tính thẩm mĩ cao và các giao diện bị lặp lại dễ gây cho người dùng nhàm chán. Mặt khác, hệ thống lại có ưu điểm dễ sử dụng và thân thiện. nhóm chúng em sẽ cố gắng để có một sản phẩm tốt nhất phù hợp với khả năng.

Những thuận lợi và khó khăn trong quá trình thực hiện:

* Thuận lợi:
* Nội dung yêu cầu của bài toán có thể xác định được định hướng.
* Công cụ Android Studio rất thông minh nên không mất nhiều thời gian để làm quen.
* Có thể áp dụng theo các bài giảng viên đã giao.
* Khó khăn:
* Mất khá nhiều thời gian để xác định chính xác được nghiệp vụ của từng chức năng.
* Rất ít tài liệu liên quan đến bài toán trên web nên khó để hoàn thành theo đúng định hướng.
* Các kiến thức về lập trình Android quá rộng, khó khăn trong việc tìm hiểu và học bổ sung thêm những kiến thức mới

# KẾT LUẬN

Thông qua việc nghiên cứu kiến thức lý thuyết về android, chúng em đã biết được kỹ năng và cách thức làm việc của android và có thể tạo được app với các chức năng đơn giản. Tuy nhiên kiến thức về android rất nhiều và rộng nên em chưa lĩnh hội được hết toàn bộ về nó, đó là cái hạn chế mà chúng em chưa khắc phục được trong thời gian học môn phát triển ứng dụng trên điện thoại di động.

Kết quả chúng em đạt được không được đúng như mục tiêu đã đặt ra về đề tài này. Về kiến thức chúng em cần sự trợ giúp của bạn và trang mạng rất nhiều mới có thể hoàn thành được bài toán. Sản phẩm hoàn thành còn nhiều lỗi về mặt thẩm mỹ, nghiệp vụ, … và chức năng của ứng dụng còn hạn chế chứ tối ưu.

Trong suốt thời gian học môn phát triển ứng dụng trên điện thoại di động của cô giáo – Vũ Thị Dương chúng em đã hiểu biết hơn về mảng mobile. Để hoàn thiện được bài tập lớn của bộ môn cô giáo đã giúp đỡ và chỉ bảo nhóm chúng em rất nhiều trong những khúc mắc. Cô giáo đã tạo cơ hội và thời gian cho chúng em để hoàn thành bài tập một cách tốt nhất. Tuy nhiên vì khả năng và kinh nghiệm chúng em còn nhiều thiếu sót nên phần mềm chưa được hoàn chỉnh mong cô giáo thông cảm cho chúng em. Chúng em sẽ cố gắng để có thể hoàn thiện sản phẩm một cách tốt nhất.

Chúng em xin chân thành cảm ơn cô!

# PHỤ LỤC

***Phụ lục 1:***

**BẢNG PHÂN CÔNG CODE**

|  |  |  |
| --- | --- | --- |
| STT | Người thực hiện | Nội dung công việc |
| 1 | Lê Thị Hảo | Quản lý thiết bị trong phòng |
| 2 | Phạm Quang Trung | Quản lý phiếu đăng ký thực hành |
| 3 | Dương Thị Huyền | Quản lý phòng |
| 4 | Trần Quốc Đạt | Quản lý phiếu xác nhận |
| 5 | Phạm Minh Tuấn | Quản lý người dùng |

# TÀI LIỆU THAM KHẢO

1. Nguyễn Bá Nghiễn; Lập trình java; NXB thông tin và truyền thông 2018.
2. <https://www.javatpoint.com/java-tutorial>
3. <https://teamvietdev.com/huong-dan-lap-trinh-java-swing-voi-ung-dung-quan-ly-hoc-vien/>