TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

**PHÂN HIỆU TẠI THÀNH PHỐ HỐ CHÍ MINH**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**2017-2018**

**XÂY DỰNG ỨNG DỤNG GHI NHẬN VI PHẠM NỘI QUY CỦA SINH VIÊN Ở PHÂN HIỆU TẠI TP.HCM DỰA TRÊN BARCODE API TRONG HỆ ĐIỀU HÀNH ANDROID**

**Sinh viên thực hiện:**

1. **Nguyễn Hoàng Minh Châu Lớp: CQ.57.CNTT**
2. Nguyễn Minh Mẫn Lớp: CQ.57.CNTT
3. Trần Xuân Lâm Lớp: CQ.57.CNTT
4. Huỳnh Phúc Trường Lớp: CQ.57.CNTT
5. Nguyễn Tấn Phát Lớp: CQ.57.CNTT

**Giảng viên hướng dẫn:**

ThS. Lê Nhật Tùng

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

**PHÂN HIỆU TẠI THÀNH PHỐ HỐ CHÍ MINH**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**2017-2018**

**XÂY DỰNG ỨNG DỤNG GHI NHẬN VI PHẠM NỘI QUY CỦA SINH VIÊN Ở PHÂN HIỆU TẠI TP.HCM DỰA TRÊN BARCODE API TRONG HỆ ĐIỀU HÀNH ANDROID**

**Sinh viên thực hiện:**

1. **Nguyễn Hoàng Minh Châu Lớp: CQ.57.CNTT**
2. Nguyễn Minh Mẫn Lớp: CQ.57.CNTT
3. Trần Xuân Lâm Lớp: CQ.57.CNTT
4. Huỳnh Phúc Trường Lớp: CQ.57.CNTT
5. Nguyễn Tấn Phát Lớp: CQ.57.CNTT

**Giảng viên hướng dẫn:**

ThS. Lê Nhật Tùng

**Mục lục**

[DANH MỤC HÌNH ẢNH](#_Toc513473722)

[DANH MỤC BẢNG BIỂU](#_Toc513473723)

[DANH MỤC NHỮNG TỪ VIẾT TẮT](#_Toc513473724)

[THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI 1](#_Toc513473725)

[THÔNG TIN VỀ SINH VIÊN CHỊU TRÁCH NHIỆM CHÍNH 4](#_Toc513473726)

[CHƯƠNG 1: TỔNG QUAN 5](#_Toc513473727)

[1.1 Đặt vấn đề 5](#_Toc513473728)

[1.2 Tình hình nghiên cứu 6](#_Toc513473729)

[1.3 Mục tiêu của đề tài 6](#_Toc513473730)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 8](#_Toc513473731)

[2.1 Công nghệ được sử dụng 8](#_Toc513473732)

[2.1.1 Barcode API 8](#_Toc513473733)

[2.1.2 SQLite 12](#_Toc513473734)

[2.2 Ngôn ngữ lập trình 15](#_Toc513473735)

[2.2.1 Java 15](#_Toc513473736)

[2.2.2 Android 18](#_Toc513473737)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 21](#_Toc513473738)

[3.1. Phân tích hệ thống 21](#_Toc513473739)

[3.1.1 Kiến trúc phần cứng và phần mềm 21](#_Toc513473740)

[3.1.2 Quy trình thực hiện việc ghi nhận lỗi vi phạm 22](#_Toc513473741)

[3.1.3 Các bước hoạt động của ứng dụng 23](#_Toc513473742)

[3.1.4 Cấu trúc tập tin dữ liệu 27](#_Toc513473743)

[3.2. Các đoạn mã lập trình chính trong ứng dụng Android quản lý ghi nhận vi phạm 30](#_Toc513473744)

[Trả về kết quả sau kết quả sau khi quét 30](#_Toc513473745)

[Thêm sinh viên vào danh sách vi phạm 31](#_Toc513473746)

[Lấy file bao gồm ảnh từ đường dẫn sau đó đọc dữ liệu từ file và hình ảnh 31](#_Toc513473747)

[Ghi file 34](#_Toc513473748)

[3.3. Xây dựng giao diện người dùng 38](#_Toc513473749)

[Giao diện đăng nhập 38](#_Toc513473750)

[Giao diện chính 38](#_Toc513473751)

[Giao diện chọn lỗi ý thức 39](#_Toc513473752)

[Giao diện danh sách lỗi vi phạm của sinh viên đã ghi nhận được 40](#_Toc513473753)

[Giao diện xem danh sách ghi nhận vi phạm theo ngày 40](#_Toc513473754)

[CHƯƠNG 4: KHẢO SÁT, KẾT LUẬN, KIẾN NGHỊ VÀ DỰ KIẾN HƯỚNG NGHIÊN CỨU TIẾP THEO 41](#_Toc513473755)

[4.1 Thực nghiệm 41](#_Toc513473756)

[4.2 Kết quả 42](#_Toc513473757)

[4.3 Kết luận 42](#_Toc513473758)

[4.3.1 Ưu điểm 42](#_Toc513473759)

[4.3.2 Nhược điểm 43](#_Toc513473760)

[4.4. Hướng phát triển 43](#_Toc513473761)

[TÀI LIỆU THAM KHẢO 45](#_Toc513473762)

# DANH MỤC HÌNH ẢNH

[Hình 2. 1 Mã vạch tuyến tính trên sản phẩm 10](#_Toc513473832)

[Hình 2. 2 Mã QR chứa tên của trường 10](#_Toc513473833)

[Hình 3. 1 Kiến trúc phần cứng và phần mềm 21](#_Toc513474172)

[Hình 3. 2 Sơ đồ khối chức năng của ứng dụng 22](#_Toc513474173)

[Hình 3. 3 Ứng dụng trên giao diện Android 23](#_Toc513474174)

[Hình 3. 4 Kiểm tra ID và Password 23](#_Toc513474175)

[Hình 3. 5 Kiểm tra nhập liệu 24](#_Toc513474176)

[Hình 3. 6 Kiểm tra thêm sinh viên vào danh sách ghi nhận 25](#_Toc513474177)

[Hình 3. 7 File Excel ghi nhận vi phạm đã xuất 26](#_Toc513474178)

[Hình 3. 8 Danh sách thông tin sinh viên 27](#_Toc513474179)

[Hình 3. 9 Danh sách hình ảnh của sinh viên 28](#_Toc513474180)

[Hình 3. 10 Danh sách ghi nhận vi phạm của sinh viên 29](#_Toc513474181)

[Hình 3. 11 Giao diện đăng nhập 38](#_Toc513474182)

[Hình 3. 12 Giao diện chính 38](#_Toc513474183)

[Hình 3. 13 Giao diện chọn lỗi tác phong 39](#_Toc513474184)

[Hình 3. 14 Giao diện chọn lỗi ý thức 39](#_Toc513474185)

[Hình 3. 15 Giao diện danh sách lỗi vi phạm của sinh viên đã ghi nhận được 40](#_Toc513474186)

[Hình 3. 16 Giao diện xem danh sách ghi nhận vi phạm theo ngày 40](#_Toc513474187)

[Hình 4. 1 Thống kê thời gian quét 41](#_Toc513473853)

# DANH MỤC BẢNG BIỂU

[Bảng 2. 1 Barcode tuyến tính 9](#_Toc513473862)

[Bảng 2. 2 Danh sách không hỗ trợ 13](#_Toc513473863)

[Bảng 2.3 Data Definition Language 14](#_Toc513473864)

[Bảng 2.4 Data Manipulation Language 14](#_Toc513473865)

[Bảng 2.5 Data Query Language 14](#_Toc513473866)

[Bảng 3. 1 Code quét mã sinh viên 30](#_Toc513473869)

[Bảng 3. 2 Code trả về kết quả sau khi quét 31](#_Toc513473870)

[Bảng 3. 3 Code thêm sinh viên vào danh sách vi phạm 31](#_Toc513473871)

[Bảng 3. 4 Code Lấy file bao gồm ảnh từ đường dẫn sau đó đọc dữ liệu từ file và hình ảnh 33](#_Toc513473872)

[Bảng 3. 5 Code ghi file 37](#_Toc513473873)

[Bảng 4. 1 Tỉ lệ thành công 41](#_Toc513473875)

[Bảng 4. 2 Thống kê thời gian quét 41](#_Toc513473876)

# DANH MỤC NHỮNG TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Viết Tắt** | **Tiếng Anh** | **Tiếng Việt** |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| CTCT&SV |  | Công tác chính trị & Sinh viên |
| KTX |  | Kí túc xá |
| **DBMS** | **Data Base Management System** | Hệ quản trị cơ sở dữ liệu |
| ACID | Atomicity, Consistency, Isolation, Durability | Tính nguyên tố, Tính nhất quán, Tính độc lập, Tính bền vững |
| ANSI-C |  | Là chuẩn C do tổ chức ANSI (Mỹ) đặt ra |
| DDL | Data Definition Language | Ngôn ngữ định nghĩa dữ liệu |
| DML | Data Manipulation Language | Ngôn ngữ thao tác dữ liệu |
| DQL | Data Query Language | Ngôn ngữ truy vấn dữ liệu |
| OOP | Object-oriented programming | Lập trình hướng đối tượng |
| JVM | Java Virtual Machine | Một thiết bị trừu tượng (ảo) |
| JDK | Java Development Kit | Hộp công cụ phát triển Java |
| IO | Input - Output | Nhập – Xuất |
| J2ME | Java 2 Micro Edition | Một nhánh của ngôn ngữ lập trình Java |
| UI | User Interface | Giao diện người dùng |

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**

# THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

**Thông tin chung**

Tên đề tài: Xây dựng ứng dụng ghi nhận vi phạm nội quy của sinh viên ở phân hiệu tại Tp.HCM dựa trên Barcode API trong hệ điều hành Android

Sinh viên thực hiện:

**Nguyễn Hoàng Minh Châu**

Nguyễn Minh Mẫn

Trần Xuân Lâm

Huỳnh Phúc Trường

Nguyễn Tấn Phát

Lớp: CQ.57.CNTT Khoa: Công nghệ thông tin

Năm thứ: 2 Số năm đào tạo: 4

Người hướng dẫn: ThS. Lê Nhật Tùng

**Mục tiêu đề tài**

Xây dựng một ứng dụng triển khai trên thiết bị di động hỗ trợ cho Phòng CTCT&SV trong việc ghi nhận vi phạm của sinh viên. Vì bên cạnh việc di chuyển người kiểm tra phải ghi chép thông tin phạm vi của các đối tượng. Việc vừa di chuyển liên tục vừa ghi chép tạo ra nhiều khó khăn, bên cạnh đó công tác thống kê báo cáo đòi hỏi nguồn nhân lực thực hiện và tiêu tốn nhiều thời gian.

**Tính mới và sáng tạo**

Trường Đại học Giao thông vận tải Phân hiệu tại Tp.Hồ Chí Minh là một trong những trung tâm đào tạo nguồn nhân lực chất lượng cao hàng đầu tại Khu vực phía Nam. Nhà trường hiện nay có gần 7000 học viên và sinh viên các hệ. Công tác thanh kiểm tra việc chấp hành nội quy của Nhà trường được thực hiện thường xuyên, góp phần nâng cao ý thức sinh viên trong việc học tập và rèn luyện tác phong ngay khi còn ngồi trên ghế Nhà trường.

Với sự phát triển nhanh chóng của điện thoại thông minh như hiện nay, đặc biệt là điện thoại sử dụng hệ điều hành Android. Một ứng dụng ghi nhận trên điện thoại sẽ rất thuận tiện cho người sử dụng khi không ở bên máy tính, đặc biệt là với những người phải thường xuyên di chuyển. Bên cạnh việc di chuyển người kiểm tra phải ghi chép thông tin phạm vi của các đối tượng. Việc vừa di chuyển liên tục vừa ghi chép tạo ra nhiều khó khăn, bên cạnh đó công tác thống kê báo cáo đòi hỏi nguồn nhân lực thực hiện và tiêu tốn nhiều thời gian.

Do đó nhóm đã đề xuất xây dựng một ứng dụng triển khai trên thiết bị di động sử dụng hệ điều hành Android nhằm quản lý và ghi nhận lỗi vi phạm của sinh viên, hỗ trợ cho Phòng CTCT&SV tiết kiệm thời gian và thực hiện công tác quản lý thanh kiểm tra việc vi phạm nội quy hiệu quả, chính xác hơn, xử lý báo cáo và thống kê một cách nhanh chóng hơn.

**Kết quả nghiên cứu**

Ứng dụng hỗ trợ điểm danh trên thiết bị di động Android với các chức năng sau: Thêm danh sách thông tin của sinh viên, ghi nhận các vi phạm của sinh viên, kết xuất thống kê và báo cáo.

Quyển báo cáo đề tài.

**Đóng góp về mặt kinh tế - xã hội, giáo dục và đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài**

Tích hợp vào bộ phận quản lý nhân sự ở các trường đại học, cao đẳng , cơ quan…. Nhằm tiết kiệm thời gian thực hiện công tác quản lý chính xác hơn, xử lý báo cáo và thống kê một cách nhanh chóng hơn.

**Công bố khoa học của sinh viên từ kết quả nghiên cứu của đề tài** *(ghi rõ họ tên tác giả, nhan đề và các yếu tố về xuất bản nếu có) hoặc nhận xét, đánh giá của cơ sở đã áp dụng các kết quả nghiên cứu (nếu có):*

|  |  |
| --- | --- |
|  | **TP. Hồ Chí Minh, ngày 02 tháng 05 năm 2018 Sinh viên thực hiện**  **Nguyễn Hoàng Minh Châu** |

**Nhận xét của người hưỡng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài** (Phần này do người hưỡng dẫn ghi):

* Nhóm sinh viên thực hiện nghiêm tức công việc nghiên cứu. Mặc dù là sinh viên năm thứ 2, các bạn đã chịu khó tự học để nắm bắt các kiến thực cần thiết khi thực hiện đề tài.
* Đề tài có tính ứng dụng cao, có khả năng áp dụng vào trong thực tế.
* Đề tài có khả năng và mở rộng sang các dạng bài toán tương tự.

|  |  |
| --- | --- |
|  | **TP. Hồ Chí Minh, ngày 02 tháng 05 năm 2018 Giảng viên hướng dẫn**  **ThS. Lê Nhật Tùng** |

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**

# THÔNG TIN VỀ SINH VIÊN CHỊU TRÁCH NHIỆM CHÍNH

**THỰC HIỆN ĐỀ TÀI**

I. SƠ LƯỢC VỀ SINH VIÊN

Họ và tên : Nguyễn Hoàng Minh Châu

Sinh ngày : 08 tháng 6 năm 1998

Nơi sinh : Bình Dương

Lớp: CQ.CNTT57 Khóa: 57

Khoa: Công Nghệ Thông Tin

Địa chỉ liên hệ: KTX Đại học Giao thông vận tải phân hiệu tại TP.HCM - 450, P. Tăng

Nhơn Phú A, Quận 9, Tp. Hồ Chí Minh

Điện thoại : 0967 065 246 Email: minhchauk1k@gmail.com

II. QUÁ TRÌNH HỌC TẬP

***\* Năm thứ 1:***

Ngành học: Công nghệ thông tin Khoa: Công nghệ thông tin

Kết quả xếp loại học tập: Trung bình

Sơ lược thành tích: Tích cực học tập, dần thích nghi môi trường đại học

***\* Kỳ 1 năm thứ 2:***

Ngành học: Công nghệ thông tin Khoa: Công nghệ thông tin

Kết quả xếp loại học tập: Trung bình

Sơ lược thành tích: Chăm chỉ học tập, có nhiều tiến bộ

# CHƯƠNG 1: TỔNG QUAN

* 1. **Đặt vấn đề**

Văn hóa là những nét đẹp, những thói quen hành xử đẹp đẽ của cá nhân hoặc cộng đồng. Là yếu tố quan trọng để xây dựng xã hội tốt đẹp, dân chủ, công bằng văn minh và đó cũng là nền tảng để xã hội phát triển một cách bền vững. Sinh viên là tầng lớp xã hội đặc thù, đang trong quá trình định hình nhân cách, đạo đức và lối sống. Đồng thời họ là những người tràn đầy sức sống, có hoài bão, ước mơ, năng động, được đào tạo một cách cơ bản, toàn diện theo tất cả các chuyên ngành; họ nhạy bén với cái mới, nắm trong tay tri thức thời đại...Văn hóa sinh viên là những định hướng, quy định chuẩn mực cả về hành động cũng như tinh thần cho sinh viên một cách đúng đắn ngay từ khi ngồi trên ghế giảng đường đại học. Một trong những điểm văn trọng trong việc thể hiện văn hóa sinh viên là việc chấp hành tốt nội quy, quy định của Nhà trường từ học tập cho đến tác phong và lề lối làm việc.

Trường Đại học Giao thông vận tải Phân hiệu tại Tp.Hồ Chí Minh, hiện nay có gần 7000 học viên và sinh viên các hệ, mỗi sinh viên đều được nhà trường cung cấp cho một thẻ sinh viên chứa mã số sinh viên và một số thông tin cá nhân của sinh viên ấy. Phòng CTCT & SV với chức năng tham mưu, giúp Ban Giám đốc về công tác tuyên truyền, giáo dục chính trị, tư tưởng cho cán bộ, viên chức và sinh viên; công tác quản lý sinh viên trong Trường.

Việc thực hiện công tác thanh - kiểm tra một cách thường xuyên và chặt chẽ mà Phòng CTCT&SV đã thực hiện nhằm đảm bảo việc chấp hành nội quy của Nhà trường và góp phần nâng cao ý thức sinh viên trong việc học tập và rèn luyện tác phong khi còn ngồi trên ghế Nhà trường. Tuy nhiên công tác nay hiện nay vẫn được cán bộ kiểm tra thực hiện thủ công đó là ghi chép và tổng hợp. Việc này tạo ra nhiều khó khăn trong việc tra cứu cũng như lưu trữ thông tin đã ghi nhận được, bên cạnh đó công tác thống kê báo cáo đòi hỏi nguồn nhân lực thực hiện và tiêu tốn nhiều thời gian.

Với việc đề xuất nghiên cứu và xây dựng ứng dụng ghi nhận vi phạm nội quy của sinh viên ở phân hiệu tại TP.HCM dựa trên Barcode API trong hệ điều hành Android đã dược nhóm đề ra, nhóm tác giả mong muốn giúp quá trình ghi nhận lỗi vi phạm của sinh viên của các cán bộ nhà trường được thực hiện một cách thông minh, nhanh chóng, dễ dàng, và hạn chế một cách tối đa sai sót trong việc ghi nhận và tra cứu thông tin, tiết kiệm được thời gian và nhân lực thực hiện công tác này.

* 1. **Tình hình nghiên cứu**

Hiện nay tại Phân hiệu Trường Đại học Giao thông Vận tải tại TP. HCM chưa có ứng dụng phục vụ công tác thanh kiểm tra và ghi nhận các lỗi vi phạm của sinh viên trên thiết bị di động Android.

Trên thị trường, ứng dụng Barcode đã và đang được áp dụng một cách phổ biến trên các sản phẩm, hàng hóa để quản lý, kiểm tra thông tin sản phẩm, hàng hóa một cách nhanh chóng. Như trong việc quản lý sách trong thư viện của trường; quản lý nguồn gốc, thông tin sản phẩm hàng hóa trong các siêu thị, cửa khẩu, hải quan, y tế, giáo dục… Tại Trường Đại học Giao thông vận tải Phân hiệu tại TP. Hồ Chí Minh này, mỗi sinh viên khi đăng ký tham gia nhập học ở trường thì đều được nhà trường cung cấp cho một thẻ sinh viên (bao gồm cả mã vạch Barcode 128 trên thẻ).

Từ những lý do trên nên nhóm đã, nhóm đã nghiên cứu phương thức và cách quét mã vạch bằng camera của thiết bị di động thông minh Android, tiến hành nghiên cứu, phân tích, cũng như giải quyết bài toán quản lý sinh viên của trường (mà ở đây là quản lý vi phạm của sinh viên trong nhà trường); thiết kế, xây dựng cơ sở dữ liệu để lưu trữ và xử lý thông tin của mỗi sinh viên theo học tại trường.

* 1. **Mục tiêu của đề tài**

- Tìm hiểu kỹ thuật Barcode API.

- Tìm hiểu ngôn ngữ lập trình Android.

- Tìm hiểu cách đọc, xuất file excel sử dụng ngôn ngữ lập trình Android.

- Phân tích bài toán ghi nhận vi phạm quy định Nhà trường của sinh viên: mặc áo không cổ, mang dép lê …. .

- Thiết kế và cài đặt ứng dụng hỗ trợ vị ghi nhận vi phạm của sinh viên. Với các chức năng cụ thể như: thêm vào danh sách thông tin của sinh viên, ghi nhận các vi phạm của sinh viên, kết xuất thống kê và báo cáo.

- Thực nghiệm và đánh giá kết quả

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Công nghệ được sử dụng

### 2.1.1 Barcode API [ 6]

#### 2.1.1.1 Sơ lược về Barcode

Barcode (Mã vạch) là sự thể hiện thông tin trong các dạng nhìn thấy trên các bề mặt của sản phẩm, hàng hóa mà máy móc có thể đọc được. Nguyên thủy thì mã vạch lưu trữ dữ liệu theo bề rộng của các vạch được in song song cũng như của khoảng trống giữa chúng, nhưng ngày nay chúng còn được in theo các mẫu của các điểm, theo các vòng tròn đồng tâm hay chúng ẩn trong các hình ảnh. Mã vạch có thể được đọc bởi các thiết bị quét quang học gọi là máy đọc mã vạch hay được quét từ hình ảnh bằng các ứng dụng chuyên biệt.

Nội dung của mã vạch là thông tin về sản phẩm như: Nước đăng ký mã vạch, tên doanh nghiệp, lô, tiêu chuẩn chất lượng đăng ký, thông tin về kích thước sản phẩm, nơi kiểm tra...

***Các ứng dụng chính:***

Mã vạch (và các thẻ khác mà máy có thể đọc được như RFID) được sử dụng ở những nơi mà các đồ vật cần phải đánh số với các thông tin liên quan để các máy tính có thể xử lý. Thay vì việc phải đánh một chuỗi dữ liệu vào phần nhập liệu của máy tính thì người thao tác chỉ cần quét mã vạch cho thiết bị đọc mã vạch. Chúng cũng làm việc tốt trong điều kiện tự động hóa hoàn toàn, chẳng hạn như trong luân chuyển hành lý ở các sân bay.

Các dữ liệu chứa trong mã vạch thay đổi tùy theo ứng dụng. Trong trường hợp đơn giản nhất là một chuỗi số định danh được sử dụng như là chỉ mục trong cơ sở dữ liệu trong đó toàn bộ các thông tin khác được lưu trữ. Các mã EAN-13 và UPC tìm thấy phổ biến trên hàng bán lẻ làm việc theo phương thức này.

Trong các trường hợp khác, mã vạch chứa toàn bộ thông tin về sản phẩm, mà không cần cơ sở dữ liệu ngoài. Điều này dẫn tới việc phát triển mã vạch tượng trưng mà có khả năng biểu diễn nhiều hơn là chỉ các số thập phân, có thể là bổ sung thêm các ký tự hoa và thường của bảng chữ cái cho đến toàn bộ bảng mã ký tự ASCII và nhiều hơn thế. Việc lưu trữ nhiều thông tin hơn đã dẫn đến việc phát triển của các ma trận mã (một dạng của mã vạch 2D), trong đó không chứa các vạch mà là một lưới các ô vuông. Các mã vạch cụm là trung gian giữa mã vạch 2D thực thụ và mã vạch tuyến tính, và chúng được tạo ra bằng cách đặt các mã vạch tuyến tính truyền thống trên các loại giấy hay các vật liệu có thể in mà cho phép có nhiều hàng.

#### 2.1.1.2 Các dạng Barcode cơ bản

**Mã vạch tuyến tính**

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại** | **Thuộc tính** | **Độ rộng** | **Sử dụng** |
| Plessey | Liên tục | 2 | Catalog, các giá hàng trong cửa hàng, hàng tồn kho |
| EAN-UCC | Liên tục | Nhiều | Bán lẻ khắp thế giới |
| Codabar | Rời rạc | 2 | Thư viện, ngân hàng máu, vé máy bay |
| Code 39 | Rời rạc | 2 | Đa dạng |
| Code 93 | Liên tục | 2 | Đa dạng |
| \*Code 128 | Liên tục | Nhiều | Đa dạng |
| Code 11 | Rời rạc | 2 | Điện thoại |
| POSTNET | Liên tục | Cao/Thấp | Bưu điện |
| PostBar | Rời rạc | Nhiều | Bưu điện |
| CPC Binary | Rời rạc | 2 | Bưu điện |

Bảng 2. 1 Barcode tuyến tính

Chương trình của nhóm và mã vạch được in trên thẻ sinh viên của Trường Đại học giao thông vận tải Phân hiệu tại TP.HCM đều sử dụng Barcode – Code 128.



Hình 2. 1 Mã vạch tuyến tính trên sản phẩm

**Mã vạch ma trận**

Mã QR một mã vạch ma trận (hay mã vạch hai chiều) xuất phát từ "*Quick Response*", trong tiếng Anh có nghĩa là *đáp ứng nhanh*, vì người tạo ra nó có ý định cho phép mã được giải mã ở tốc độ cao. Các mã QR được sử dụng phổ biến nhất ở Nhật Bản, và hiện là loại mã hai chiều thông dụng nhất ở Nhật Bản.



Hình 2. 2 Mã QR chứa tên của trường

Mã QR trên thể hiện dòng chữ: “**Đại Học Giao Thông Vận Tải Phân Hiệu Tại TP.HCM**” sau khi được quét.

#### 2.1.1.3 Sơ lược về API [7]

Một API là một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng.

Chẳng hạn, một chương trình máy tính có thể (và thường là phải) dùng các hàm API của hệ điều hành để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng hiện thực API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính. Đây là ứng dụng hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các lập trình viên có thể rút ra từ đó để tạo nên các tính năng giao tiếp người- máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện.

Hệ giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn.

Một trong các mục đích chính của một API là cung cấp khả năng truy xuất đến một tập các hàm hay dùng — ví dụ: hàm để vẽ các cửa sổ hay các icon trên màn hình.

#### 2.1.1.4 Định nghĩa Barcode API

Barcode API là một ứng dụng trung gian, khi ta mở ứng dụng quản lý vi phạm sinh viên lên thì ứng dụng trung gian này sẽ yêu cầu lầy dữ liệu từ camera của thiết bị di động thông minh của ta, dữ liệu lấy được sẽ được gửi riêng về ứng dụng trung gian để diễn giải dữ liệu, thực hiện các hành động cần thiết và gửi dữ liệu/kết quả (mà ở đây được hiểu là kết quả sau khi quét được của Barcode – Code 128 sẽ là *Mã sinh viên* mà chúng ta cần) trở lại thiết bị của người dùng.

### 2.1.2 SQLite [8]

#### 2.1.2.1 Sơ lược về SQLite

SQLite là một thư viện ứng dụng mà triển khai một SQL Database Engine truyền thống, không cần Server, không cần cấu hình và nhỏ gọn. SQLite là một trong những Database Engine phát triển nhanh nhất, nhưng sự phát triển của nó là về mặt tính phổ biến, không phải là về mặt kích cỡ. SQLite là mã nguồn mở.

Nói một cách đơn giản SQLite là ứng dụng quản lý cơ sở dữ liệu (DBMS) tương tự như Mysql, PostgreSQL... Đặc điểm của SQLite là gọn, nhẹ, đơn giản. Chương trình gồm 1 file duy nhất vỏn vẹn chưa đến 400kB, không cần cài đặt, không cần cấu hình hay khởi động mà có thể sử dụng ngay. Dữ liệu Database cũng được lưu ở một file duy nhất. Không có khái niệm user, password hay quyền hạn trong SQLite Database.

SQLite không thích hợp với những hệ thống lớn nhưng ở quy mô vừa tầm thì SQLite phát huy uy lực và không hề yếu kém về mặt chức năng hay tốc độ. Với các đặc điểm trên SQLite được sử dụng nhiều trong việc phát triển, thử nghiệm … và là sự lưa chọn phù hợp cho những người bắt đầu học Database.

#### 2.1.2.2 Đặc điểm của SQLite

**Ưu điểm:**

- SQLite không yêu cầu một tiến trình Server riêng rẽ để hoạt động.

- SQLite không cần cấu hình, nghĩa là không cần thiết phải cài đặt.

- Một SQLite Database đầy đủ được lưu giữ trong một disk file đơn.

- SQLite là rất nhỏ gọn, nhỏ hơn 400kB đã đươc cấu hình đầy đủ hoặc nhỏ hơn 250kB khi đã bỏ qua các tính năng tùy ý.

- SQLite là tự chứa, nghĩa là không có sự phụ thuộc vào ngoại vi.

- Các Transaction trong SQLite là tuân theo đầy đủ chuẩn ACID, đảm bảo truy cập an toàn từ nhiều tiến trình hoặc thread.

- SQLite hỗ trợ hầu hết các tính năng của một ngôn ngữ truy vấn trong chuẩn SQL92.

- SQLite được viết bằng ANSI-C và cung cấp API đơn giản và dễ dàng sử dụng.

- SQLite là có sẵn trên UNIX (Linux, Mac OS-X, Android, iOS) và Windows (Win32, WinCE, WinRT).

**Nhược điểm:**

SQLite không hỗ trợ một số tính năng của chuẩn SQL92, bao gồm:

|  |  |
| --- | --- |
| **Đặc điểm** | **Miêu tả** |
| RIGHT OUTER JOIN | Chỉ có LEFT OUTER JOIN được triển khai |
| FULL OUTER JOIN | Chỉ có LEFT OUTER JOIN được triển khai |
| ALTER TABLE | Các biến thể RENAME TABLE và ADD COLUMN của lệnh ALTER TABLE được hỗ trợ. Trong khi DROP COLUMN, ALTER COLUMN, ADD CONSTRAINT không được hỗ trợ |
| Hỗ trợ Trigger | Hỗ trợ FOR EACH ROW trigger nhưng không hỗ trợ FOR EACH STATEMENT |
| VIEWs | View trong SQLite là read-only. Bạn không thể thực thi một lệnh DELETE, INSERT, hoặc UPDATE trên một View |
| GRANT và REVOKE | Chỉ hỗ trợ các quyền truy cập mà áp dụng trên normal file dưới hệ điều hành |

Bảng 2. 2 Danh sách không hỗ trợ

**Câu lệnh trong SQLite**

Các lệnh SQLite chuẩn để tương tác với Cơ sở dữ liệu quan hệ là giống như SQL. Chúng là CREATE, SELECT, INSERT, UPDATE, DELETE và DROP. Những lệnh này có thể được phân chia thành 3 nhóm tùy theo tính năng hoạt động của chúng:

**Nhóm DDL (Data Definition Language)**

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| CREATE | Tạo một bảng mới, một View của một bảng hoặc đối tượng khác trong Database |
| ALTER | Sửa đổi một đối tượng cơ sở dữ liệu đang tồn tại, ví dụ một bảng |
| DROP | Xóa cả một bảng, một View của một bảng hoặc đối tượng khác trong Database |

Bảng 2. Data Definition Language

**Nhóm DML (Data Manipulation Language)**

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| INSERT | Tạo một bản ghi |
| UPDATE | Sửa đổi các bản ghi |
| DELETE | Xóa các bản ghi |

Bảng 2. Data Manipulation Language

**Nhóm DQL (Data Query Language)**

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| SELECT | Lấy các bản ghi cụ thể từ một hoặc nhiều bảng |

Bảng 2. Data Query Language

## 2.2 Ngôn ngữ lập trình

### 2.2.1 Java [9]

#### 2.2.1.1 Khái niệm

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như Python, Perl, PHP gấp nhiều lần. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy.

Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác". Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.

#### 2.2.1.2 Lịch sử hình thành

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun Microsystems năm 1991. Ban đầu ngôn ngữ này được gọi là Oak (có nghĩa là cây sồi; do bên ngoài cơ quan của ông Gosling có trồng nhiều loại cây này), họ dự định ngôn ngữ đó thay cho C++, nhưng các tính năng giống Objective C. Không nên lẫn lộn Java với JavaScript, hai ngôn ngữ đó chỉ giống tên và loại cú pháp như C. Công ty Sun Microsystems đang giữ bản quyền và phát triển Java thường xuyên. Tháng 04/2011, công ty Sun Microsystems tiếp tục cho ra bản JDK 1.6.24.

Java được tạo ra với tiêu chí "Viết (code) một lần, thực thi khắp nơi" ("Write Once, Run Anywhere" (WORA)). Chương trình ứng dụng viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó. Môi trường thực thi của Sun Microsystems hiện hỗ trợ Sun Solaris, Linux, Mac OS, FreeBSD & Windows. Ngoài ra, một số công ty, tổ chức cũng như cá nhân khác cũng phát triển môi trường thực thi Java cho những hệ điều hành khác như BEA, IBM, HP.... Trong đó đáng nói đến nhất là IBM Java Platform hỗ trợ Windows, Linux, AIX & z/OS.

Những chi tiết về ngôn ngữ, máy ảo và API của Java được giữ bởi Cộng đồng Java (do Sun quản lý). Java được tạo ra vào năm 1991 do một số kỹ sư ở Sun, bao gồm ông James Gosling, một phần của Dự án Xanh (Green Project). Java được phát hành vào năm 1994, rồi nó trở nên nổi tiếng khi Netscape tuyên bố tại hội thảo SunWorld năm 1995 là trình duyệt Navigator của họ sẽ hỗ trợ Java. Về sau Java được hỗ trợ trên hầu hết các trình duyệt như Internet Explorer (Microsoft), Firefox (Mozilla), Safari (Apple), ...

Java được sử dụng chủ yếu trên môi trường NetBeans và Oracle. Sau khi Oracle mua lại công ty Sun Microsystems năm 2009-2010, Oracle đã mô tả họ là "người quản lý công nghệ Java với cam kết không ngừng để bồi dưỡng một cộng đồng tham gia và minh bạch".

#### 2.2.1.3 Đặc điểm của Java [10]

**Ưu điểm:**

+ Ưu điểm được nhắc đến đầu tiên, cũng chính là niềm tự hào đặc trưng của JAVA: **Cross – Platform**. Vậy **Cross – Platform** có nghĩa là bạn có thể viết mã nguồn trên một IDE ở một máy tính và thực thi chương trình ở bất cứ máy tính sử dụng hệ điều hành nào. Với điều kiện máy tính thực thi chương trình phải có hệ thống trình biên-thông dịch riêng được gọi là Java Virtual Machine (JVM), JVM sẽ có nhiều phiên bản dành riêng cho từng hệ điều hành (Windows, Linus và MAC OS).

+ Ưu điểm thứ 2, Java là ngôn ngữ hướng đối tượng – phù hợp với xu thế phát triển. “Chương trình phải được thiết kế trước khi xây dựng”, không lập trình viên nào có thể thực hiện toàn bộ một chương trình ngay từ đầu. Công việc đầu tiên phải thiết kế phân tích chia nhỏ các phần công việc độc lập, tiếp tục chia các phần nhỏ thành nhỏ hơn nữa. Với cơ chế đặc trưng “sử dụng lại”(kế thừa) , “mở rộng” (đa hình) trong kiến trúc OOP, lập trình viên có thể đầu tư phân tích và kiến tạo cấu trúc của toàn bộ một chương trình (Software Engineering) sao cho phù hợp với tính chất công việc.

+ Ưu điểm thứ 3, công nghệ mở và phát triển không ngừng. Cạnh tranh với các công nghệ trong .NET, cộng đồng công nghệ trong JAVA cũng không kém cạnh và miễn phí đường đường chính chính.

+ Các ưu điểm khác, phát triển mạnh hướng network, di động (phổ biến hiện nay là Android, ngoài ra còn có J2ME), an toàn trong công tác quản lí tài nguyên cấp phát động (cơ chế Garbage Collection – tương đồng trong C#), bảo mật ở mức quản lí chương trình với nhiều mức. Khi Java lần đầu tiên ra mắt đã gây nhiều ấn tượng với khả năng viết applet, là một một ứng dụng cho phép nhúng trên trang web, tuy nhiên về sau dần trở nên mờ nhạt vì nhược điểm thiếu an toàn bảo mật.

**Nhược điểm:**

+ Bảo mật sourcecode không tốt.

+ Tốc độ khá chậm, nhưng vẫn chấp nhận được.

### 2.2.2 Android [11]

#### 2.2.2.1 Khái niệm

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Vào tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác. Bản chất mở của Android cũng khích lệ một đội ngũ đông đảo lập trình viên và những người đam mê sử dụng mã nguồn mở để tạo ra những dự án do cộng đồng quản lý. Những dự án này bổ sung các tính năng cao cấp cho những khách hàng thích tìm tòi hoặc đưa Android vào các thiết bị ban đầu chạy hệ điều hành khác.

#### 2.2.2.2 Lịch sử hình thành

Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, ứng dụng, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

Từ năm 2008, Android đã trải qua nhiều lần cập nhật để dần dần cải tiến hệ điều hành, bổ sung các tính năng mới và sửa các lỗi trong những lần phát hành trước.

Mỗi bản nâng cấp được đặt tên lần lượt theo thứ tự bảng chữ cái, theo tên của một món ăn tráng miệng; ví dụ như phiên bản 1.5 Cupcake (bánh bông lan nhỏ có kem) tiếp nối bằng phiên bản 1.6 Donut (bánh vòng). Phiên bản mới nhất hiện nay là 8.1 Oreo ra mắt vào ngày 5 tháng 12 năm 2017.

Vào năm 2010, Google ra mắt loạt thiết bị Nexus—một dòng sản phẩm bao gồm điện thoại thông minh và máy tính bảng chạy hệ điều hành Android, do các đối tác phần cứng sản xuất. HTC đã hợp tác với Google trong chiếc điện thoại thông minh Nexus đầu tiên, Nexus One.

Kể từ đó nhiều thiết bị mới hơn đã gia nhập vào dòng sản phẩm này, như điện thoại Nexus 4 và máy tính bảng Nexus 10, lần lượt do LG và Samsung sản xuất. Google xem điện thoại và máy tính bảng Nexus là những thiết bị Android chủ lực của mình, với những tính năng phần cứng và ứng dụng mới nhất của Android.

#### 2.2.2.3 Đặc điểm của Android

**Ưu điểm:**

+ Là hệ điều hành có mã nguồn mở nên khả năng tuỳ biến cao, có thể tùy ý chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google.

+ Đa dạng sản phẩm, rất nhiều hãng điện thoại, thiết bị công nghệ đã ưu ái chọn Android cho thiết bị của họ, giá cả thì hợp lý từ bình dân đến cao cấp.

+ Kho ứng dụng Google Play Store đồ sộ.

+ Thân thiện và dễ sử dụng.

+ Khả năng đa nhiệm, chạy cùng lúc nhiều ứng dụng cao.

**Nhược điểm:**

+ Dễ nhiễm ứng dụng độc hại và virus: Do tính chất mã nguồn mở, nhiều ứng dụng không được kiểm soát có chất lượng không tốt hoặc lỗi bảo mật vẫn được sử dụng.

+ Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.

+ Sự phân mảnh lớn: Trong khi một số thiết bị Android xuất sắc đã trình làng như Huawei Mate RS Porsche Design (59 triệu), Samsung Galaxy S9+ (25 triệu), Xiaomi Mi Mix 2S (15 triệu),…, tuy nhiên vẫn còn rất nhiều sản phẩm giá rẻ bình dân khác (dưới 10 triệu).

+ Cập nhật không tự động với tất cả thiết bị: Khi một phiên bản hệ điều hành mới ra mắt, không phải tất cả sản phẩm đều được cập nhật, thậm chí nếu muốn trải nghiệm bạn thường xuyên phải mua mới thiết bị.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1. Phân tích hệ thống

### 3.1.1 Kiến trúc phần cứng và phần mềm

****

Hình 3. 1 Kiến trúc phần cứng và phần mềm

Bằng việc sử dụng công nghệ Barcode API, mà chúng ta có thể định dạng, quét và quan trọng nhất là lấy được Mã sinh viên được in trên thẻ sinh viên của các bạn sinh viên, thông qua camera của thiết bị di động thông minh.

Ngoài ra ngày nay do công nghệ đang liên tục phát triển và đổi mới không ngừng, nên các thiết bị di động thông minh cũng được cải tiến theo, điều đó giúp ta có thể chèn/thêm vào thiết bị di động thông minh một file Excel để đọc danh sách thông tin của những sinh viên trong trường mà không gặp bất cứ khó khăn gì.

Khi đã quét - lấy được mã sinh viên và ứng dụng đã được thêm vào danh sách thông tin của các sinh viên cần thiết thì ứng dụng ghi nhận của nhóm sẽ tiến hành kiểm tra thông tin sinh viên xem có khớp với danh sách đã thêm vào hay không (để tránh trường hợp giả mạo hoặc thẻ sinh viên giả). Tiếp đến là sẽ thêm sinh viên vừa được quét thẻ ấy vào danh sách vi phạm đã được tạo trước đó theo tùy ý sử dụng của người dùng (file Excel mà ứng dụng tạo riêng để lưu trữ thông tin vi phạm của sinh viên trong trường). Và đến khi cần thiết thì người dùng có thể thao tác Export (xuất) file Excel đó ra để đưa lên máy tính, sử dụng ứng dụng Microsoft Excel của hãng ứng dụng Microsoft để có thể đọc file, xử lý, định dạng và thống kê danh sách vi phạm của sinh viên mà ứng dụng của nhóm đã ghi nhận được trước đó.

Bằng phương pháp này, sẽ giúp cho công việc ghi nhận vi phạm cũng nhu quản lý, giám sát sinh viên đối với cán bộ nhà trường sẽ trở nên dễ dàng và nhanh chóng hơn bao giờ hết. Hạn chế tối đa sai sót và kiểm tra thông tin sinh viên trong việc ghi nhận bằng sổ sách thủ công như trước kia.

### 3.1.2 Quy trình thực hiện việc ghi nhận lỗi vi phạm

****

Hình 3. 2 Sơ đồ khối chức năng của ứng dụng

### 3.1.3 Các bước hoạt động của ứng dụng

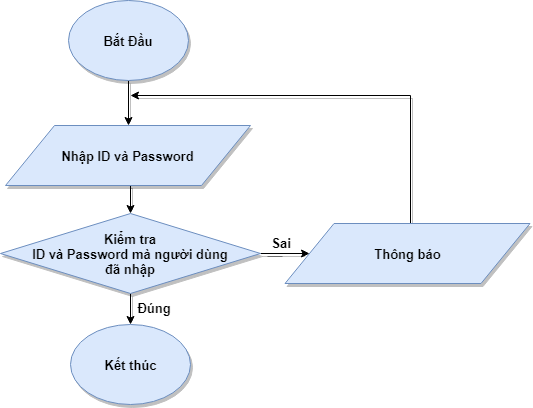
- Bước 1: Công việc đầu tiên cần phải thực hiện khởi động ứng dụng.



Hình 3. 3 Ứng dụng trên giao diện Android

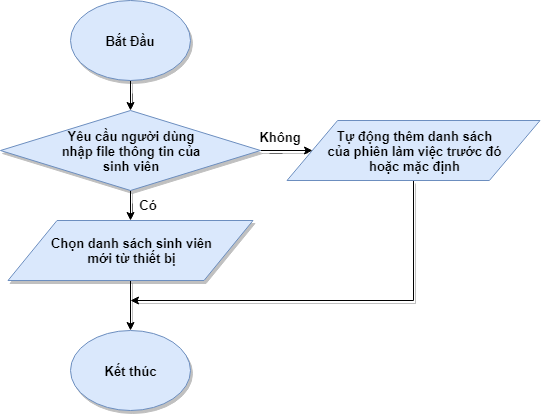
- Bước 2: Kế đến là nhập Tài khoản và Mật khẩu đã được cấp từ trước. Nếu nhập sai sẽ yêu cầu người dùng nhập lại.

*(Tài khoản mặc định ở đây là: admin@ và Mật khẩu: admin)*



Hình 3. 4 Kiểm tra ID và Password

- Bước 3 Sau khi đã nhập thành công ID và Password thì ứng dụng sẽ yêu cầu người dùng Import – chèn file *(file Excel chứa danh sách thông tin của sinh viên)*.



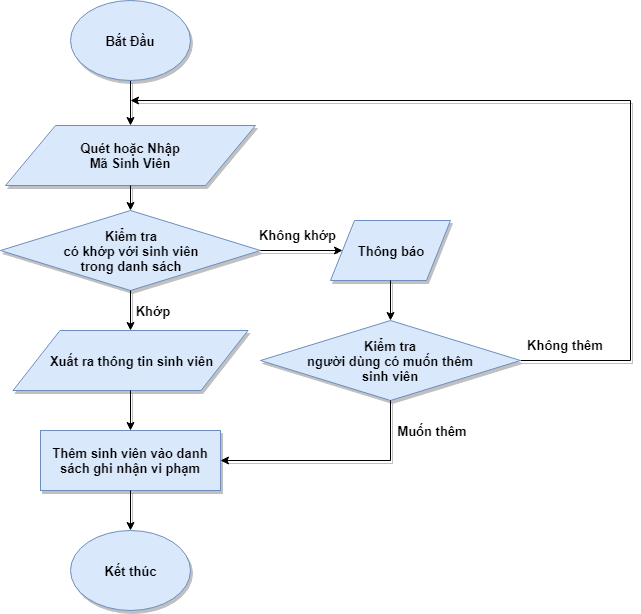
Hình 3. 5 Kiểm tra nhập liệu

- Bước 4: Tiếp đến là người dùng sẽ tạo danh sách ghi nhận sinh viên vi phạm theo ý của mình, bằng việc tick chọn những mục mà sinh viên đó đã vi phạm.

- Bước 5 Kế tiếp là ứng dụng sẽ sử dụng camera để quét thẻ sinh viên, còn nếu thẻ bị mờ, khó định dạng thì người dùng có thể nhập trực tiếp Mã sinh viên vào khung tìm kiếm trên ứng dụng. Sau đó thì ứng dụng sẽ thêm Mã sinh viên đó vào danh sách ghi nhận vi phạm:

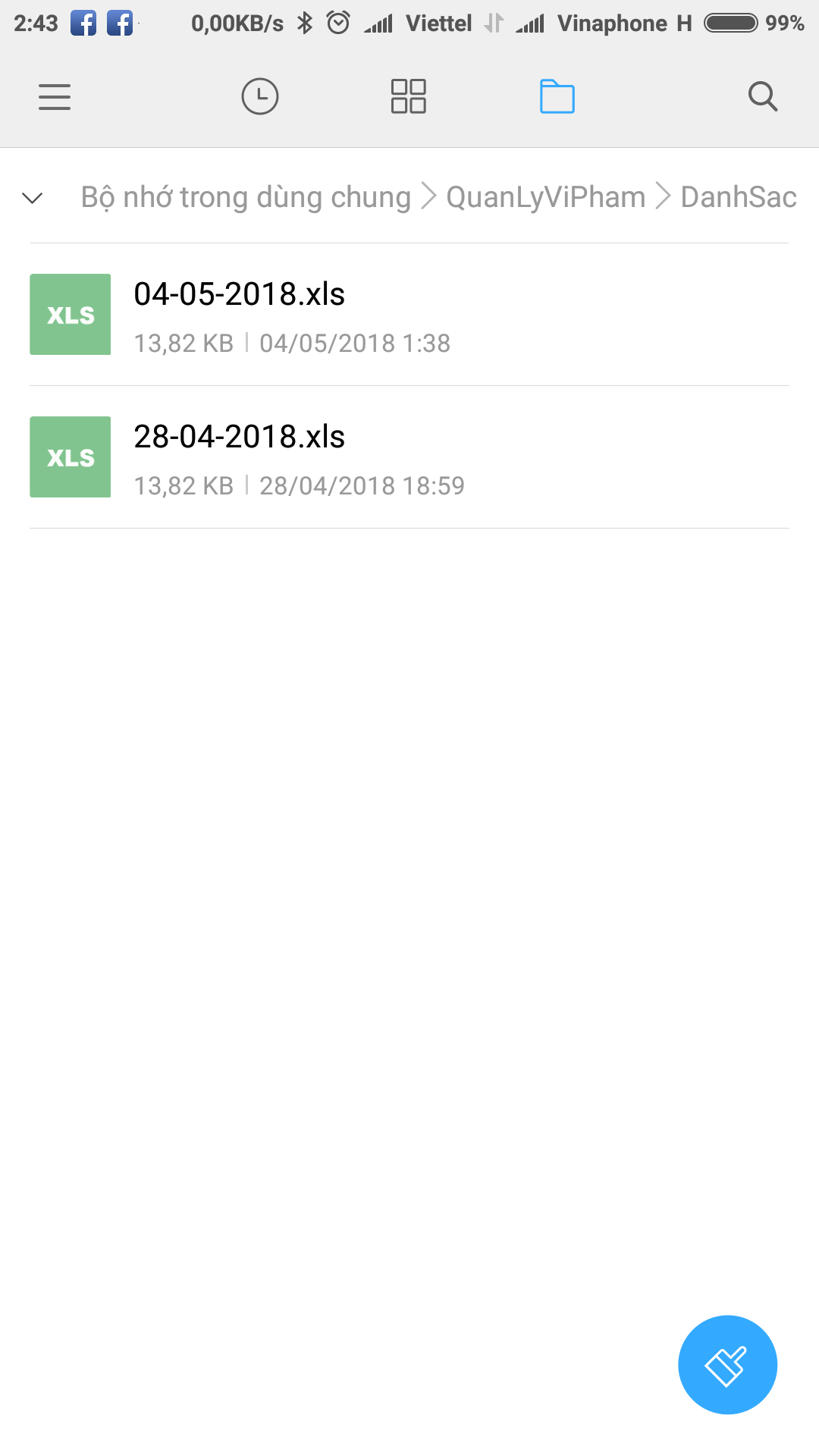
Nếu khớp với cơ sở dữ liệu (danh sách thông tin của sinh viên trong file Excel đã được Import ban đầu), thì sẽ có thể xem thông tin sinh viên (chú ý để thanh “Mã sinh viên” trống, vì nó ưu tiên xem thông tin sinh viên trong thanh “Mã sinh viên).

Ngược lại không khớp với cơ sở dữ liệu thì ứng dụng sẽ hiện thông báo “Mã sinh viên không tồn tại”.



Hình 3. 6 Kiểm tra thêm sinh viên vào danh sách ghi nhận

- Bước 6: Sau đó thì người dùng xuất ra file Excel chứa danh sách ghi nhận lỗi vi phạm của sinh viên. File được lưu trữ mặc định trong thư mục **QuanLyViPham>DanhSachViPham** trong bộ nhớ của thiết bị. Và được sắp xếp theo ngày ghi nhận vi phạm.

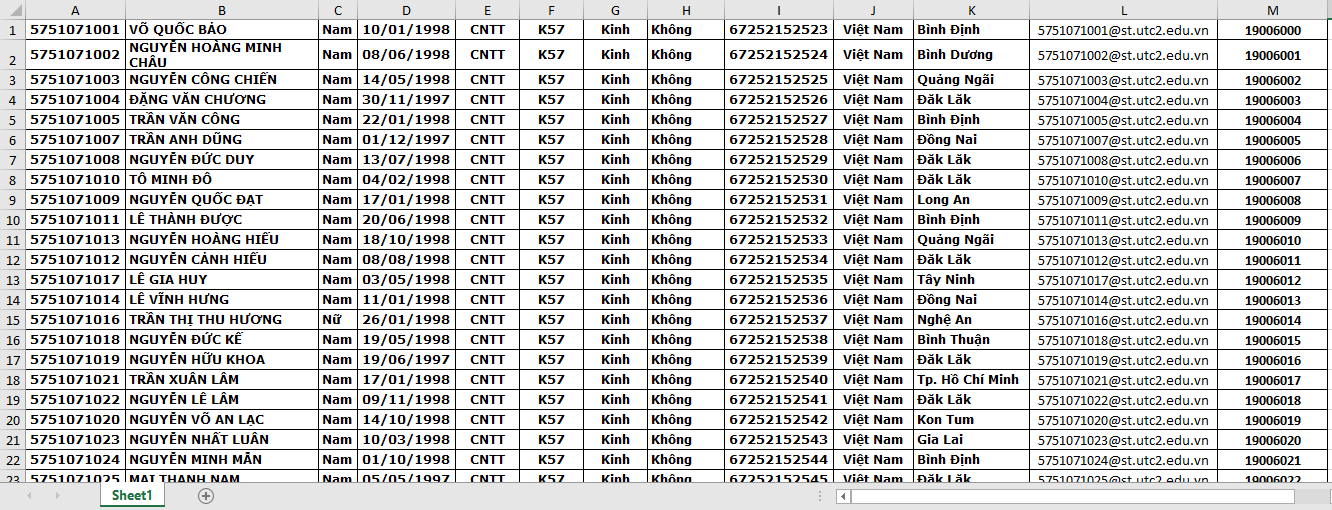


Hình 3. 7 File Excel ghi nhận vi phạm đã xuất

- Bước 7: Và việc cuối cùng là thoát khỏi ứng dụng.

### 3.1.4 Cấu trúc tập tin dữ liệu

#### 3.1.4.1 Danh sách thông tin sinh viên

****

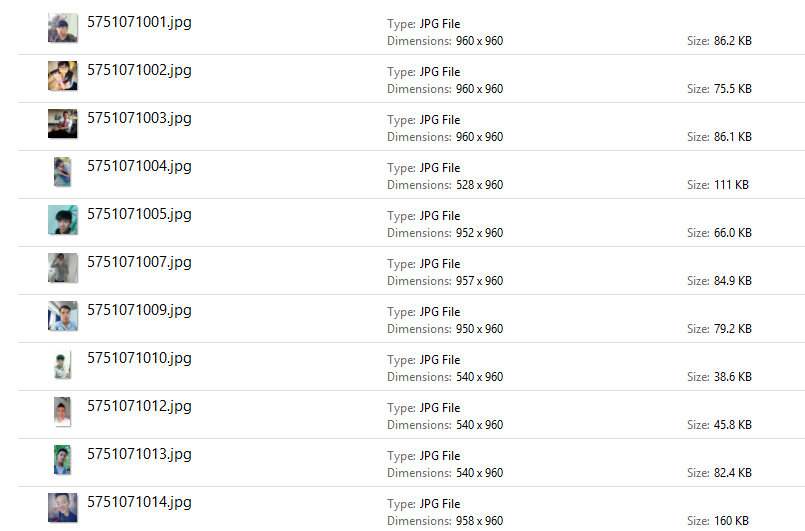
Hình 3. 8 Danh sách thông tin sinh viên

Cấu trúc tập tin danh sách thông tin sinh viên được thực hiện trên chương trình Microsoft Exel. Cấu trúc bao gồm các thành phần như sau:

* Cột A: chứa Mã sinh viên của sinh viên
* Cột B: chứa họ và tên của sinh viên
* Cột C: chứa giới tính của sinh viên
* Cột D: chứa ngày tháng năm sinh của sinh viên theo định dạng DD/MM/YYYY
* Cột E: chứa lớp/ngành mà sinh viên đó đang theo học
* Cột F: chứa khóa của sinh viên đó tại trường
* Cột G: chứa dân tộc của sinh viên
* Cột H: chứa tôn giáo của sinh viên
* Cột I: chứa CMND của sinh viên
* Cột J: chứa quốc tịch của sinh viên
* Cột K: chứa quê quán của sinh viên
* Cột L: chứa Email của sinh viên

- Cột M: chứa số điện thoại liên lạc của sinh viên

#### 3.1.4.2 Danh sách hình ảnh của sinh viên



Hình 3. 9 Danh sách hình ảnh của sinh viên

Là danh sách hình ảnh của mỗi sinh viên đi kèm với danh sách thông tin sinh viên và được đặt tên theo Mã sinh viên.

*Ví dụ: ảnh có tên* ***5751071003.jpg*** *– là hình ảnh của sinh viên có Mã sinh viên khớp với* ***5751071003****.*

Trường hợp sinh viên không có ảnh trong trong danh sách này thì ứng dụng sẽ hiển thị hình ảnh mặc định cho sinh viên ấy.

#### 3.1.4.3 Danh sách ghi nhận vi phạm của sinh viên



Hình 3. 10 Danh sách ghi nhận vi phạm của sinh viên

Cấu trúc tập tin danh sách ghi nhận vi phạm của sinh viên được thực hiện trên chương trình Microsoft Exel. Cấu trúc bao gồm các thành phần như sau:

* Cột ID: chứa số thứ tự của sinh viên trong danh sách ghi nhận vi phạm
* Cột Mã sinh viên: chứa Mã sinh viên của sinh viên vi phạm
* Cột Lỗi tác phong: chứa những lỗi tác phong mà sinh viên đó vi phạm
* Cột Lỗi ý thức: chứa những lỗi ý thức mà sinh viên đó vi phạm
* Cột Ngày vi phạm: chứa ngày mà sinh viên đó vi phạm theo định dạng DD-MM-YYYY
* Cột Giờ vi phạm: chứa thời gian ghi nhận vi phạm của sinh viên đó theo định dạng hh-mm-ss

## 3.2. Các đoạn mã lập trình chính trong ứng dụng Android quản lý ghi nhận vi phạm

**Quét mã sinh viên**

|  |
| --- |
| final Activity activity = this;  IntentIntegrator integrator = new IntentIntegrator(activity);  integrator.setDesiredBarcodeFormats(IntentIntegrator.ALL\_CODE\_TYPES);  integrator.setPrompt("Đang quét");  integrator.setCameraId(0);  integrator.setBeepEnabled(false);  integrator.setBarcodeImageEnabled(false);  integrator.initiateScan(); |

Bảng 3. 1 Code quét mã sinh viên

### Trả về kết quả sau kết quả sau khi quét

|  |
| --- |
| @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);  final MediaPlayer scanSoundMediaPlayer = MediaPlayer.create(this, R.raw.scan);  if(result != null) {  if(result.getContents() == null) {  Log.d("MainActivity", "Đã tắt Scan");  Toast.makeText(this, "Đã tắt", Toast.LENGTH\_LONG).show();  } else {  Log.d("MainActivity", "Hoàn thành");  scanSoundMediaPlayer.start();  Toast.makeText(this, "Kết quả: " + result.getContents(), Toast.LENGTH\_LONG).show();  resultString=result.getContents();  }  } else {  super.onActivityResult(requestCode, resultCode, data);  }  } |

Bảng 3. 2 Code trả về kết quả sau khi quét

### Thêm sinh viên vào danh sách vi phạm

|  |
| --- |
| //Them sinh vien  Student student = new Student(resultString,AwarenessFragment.loiTacPhong(),  MannersFragment.loiYThuc(),NgayViPham(),GioViPham(),MenusFragment.filePath);  if (student != null) {  MenusFragment.dbManager.addStudent(student);  }  studentList.clear(); // du lieu la 0 khong phai null  studentList.addAll(MenusFragment.dbManager.getAllStudent());  setAdapter();  activityB.finish(); |

Bảng 3. 3 Code thêm sinh viên vào danh sách vi phạm

### Lấy file bao gồm ảnh từ đường dẫn sau đó đọc dữ liệu từ file và hình ảnh

|  |
| --- |
| try{  File is = new File(filePath);  WorkbookSettings workbookSettings = new WorkbookSettings();  workbookSettings.setEncoding("Cp1252");  Workbook wb = Workbook.getWorkbook(is, workbookSettings);  Sheet s = wb.getSheet(0);  int row = s.getRows();  for (int i = 0; i < row; i++) {  Cell z = s.getCell(1, i);  if (ma.equalsIgnoreCase(z.getContents())){  tvMaSV.setText(z.getContents());  z = s.getCell(2, i);  tvHoten = (TextView) findViewById(R.id.hoTen);  tvHoten.setText(z.getContents());  z = s.getCell(3, i);  tvGioiTinh = (TextView) findViewById(R.id.gioiTinh);  tvGioiTinh.setText(z.getContents());  z = s.getCell(4, i);  tvNgaySinh = (TextView) findViewById(R.id.ngaySinh);  tvNgaySinh.setText(z.getContents());  z = s.getCell(5, i);  tvLop = (TextView) findViewById(R.id.lop);  tvLop.setText(z.getContents());  z = s.getCell(6, i);  tvKhoa = (TextView) findViewById(R.id.khoa);  tvKhoa.setText(z.getContents());  z = s.getCell(7, i);  tvDanToc = (TextView) findViewById(R.id.danToc);  tvDanToc.setText(z.getContents());  z = s.getCell(8, i);  tvTonGiao = (TextView) findViewById(R.id.tonGiao);  tvTonGiao.setText(z.getContents());  z = s.getCell(9, i);  tvCMND = (TextView) findViewById(R.id.CMND);  tvCMND.setText(z.getContents());  z = s.getCell(10, i);  tvQuocTich = (TextView) findViewById(R.id.quocTich);  tvQuocTich.setText(z.getContents());  z = s.getCell(11, i);  tvQueQuan = (TextView) findViewById(R.id.queQuan);  tvQueQuan.setText(z.getContents());  z = s.getCell(12, i);  tvEmail = (TextView) findViewById(R.id.email);  tvEmail.setText(z.getContents());  z = s.getCell(13, i);  tvSdt = (TextView) findViewById(R.id.sdt);  tvSdt.setText(z.getContents());  File imgFile = new File(filePath.substring(0,filePath.lastIndexOf  ("."))+"/"+tvMaSV.getText().toString()+".jpg");  if(imgFile.exists()){  Bitmap myBitmap = BitmapFactory.decodeFile(imgFile.getAbsolutePath());  igAvatar = (ImageView) findViewById(R.id.profile\_image);  igAvatar.setImageBitmap(myBitmap);  }  }  }  }catch (Exception e){  } |

Bảng 3. 4 Code Lấy file bao gồm ảnh từ đường dẫn sau đó đọc dữ liệu từ file và hình ảnh

### Ghi file

|  |
| --- |
| public void Write() {  String selectQuery = "SELECT \* FROM " + TABLE\_NAME;  SQLiteDatabase db = this.getWritableDatabase();  Cursor cursor = db.rawQuery(selectQuery, null); //Cursor dung de hung ket qua tra ve  String Fnamexls = NgayViPham() + ".xls";  File sdCard = Environment.getExternalStorageDirectory();  File directory = new File(sdCard.getAbsolutePath() + "/QuanLyViPham/DanhSachViPham");  directory.mkdirs();  File file = new File(directory, Fnamexls);  WritableWorkbook workbook;  int r = 3;  if (cursor.moveToFirst()) { // phuong thuc nay false la danh sach khong co gi (rong)  do {  try {  if (!file.exists()) {  workbook = Workbook.createWorkbook(file);  WritableSheet sheet = workbook.createSheet("Sheet 1", 0);  sheet.addCell(new Label(0, 0, "THÔNG TIN SINH VIÊN"));  sheet.addCell(new Label(0, 2, "ID"));  sheet.addCell(new Label(1, 2, "Mã sinh viên"));  sheet.addCell(new Label(2, 2, "Lỗi tác phong"));  sheet.addCell(new Label(3, 2, "Lỗi ý thức"));  sheet.addCell(new Label(4, 2, "Ngày vi phạm"));  sheet.addCell(new Label(5, 2, "Giờ vi phạm"));  Label labe1 = new Label(0, r, String.valueOf(cursor.getInt(0))); // id  Label labe2 = new Label(1, r, cursor.getString(1));  Label labe3 = new Label(2, r, cursor.getString(2));  Label labe4 = new Label(3, r, cursor.getString(3)); //  Label labe5 = new Label(4, r, cursor.getString(4)); //  Label labe6 = new Label(5, r, cursor.getString(5)); //  try {  sheet.addCell(labe1);  sheet.addCell(labe2);  sheet.addCell(labe3);  sheet.addCell(labe4);  sheet.addCell(labe5);  sheet.addCell(labe6);  } catch (RowsExceededException e) {  e.printStackTrace();  } catch (WriteException e) {  e.printStackTrace();  }  workbook.write();  try {  workbook.close();  } catch (WriteException e) {  e.printStackTrace();  }  } else {  Workbook wb = Workbook.getWorkbook(file);  workbook = Workbook.createWorkbook(file, wb);  WritableSheet sheet = workbook.getSheet(0);  Label labe1 = new Label(0, r, String.valueOf(cursor.getInt(0))); // id  Label labe2 = new Label(1, r, cursor.getString(1)); // ten  Label labe3 = new Label(2, r, cursor.getString(2)); // email  Label labe4 = new Label(3, r, cursor.getString(3)); // dia chi  Label labe5 = new Label(4, r, cursor.getString(4)); //  Label labe6 = new Label(5, r, cursor.getString(5)); //  try {  sheet.addCell(labe1);  sheet.addCell(labe2);  sheet.addCell(labe3);  sheet.addCell(labe4);  sheet.addCell(labe5);  sheet.addCell(labe6);  } catch (RowsExceededException e) {  e.printStackTrace();  } catch (WriteException e) {  e.printStackTrace();  }  workbook.write();  try {  workbook.close();  } catch (WriteException e) {  e.printStackTrace();  }  }  } catch (IOException e) {  e.printStackTrace();  } catch (Exception e) {  }  r++;  } while (cursor.moveToNext()); // sau ket qua co it nhat mot ket qua  }  } |

Bảng 3. 5 Code ghi file

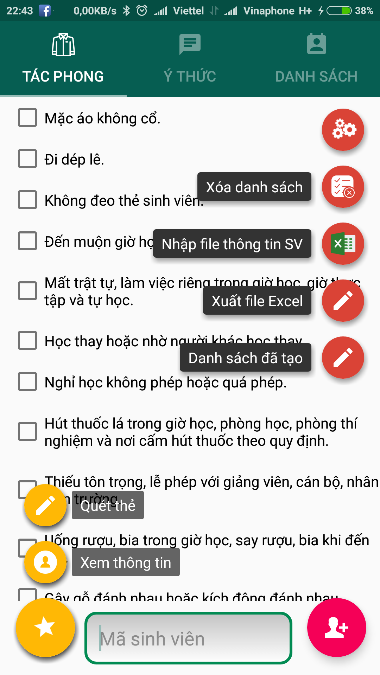
## 3.3. Xây dựng giao diện người dùng

### Giao diện đăng nhập

****

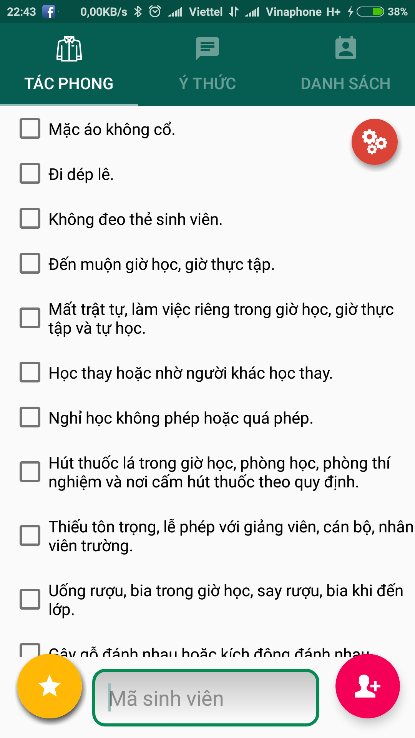
Hình 3. 11 Giao diện đăng nhập

### Giao diện chính



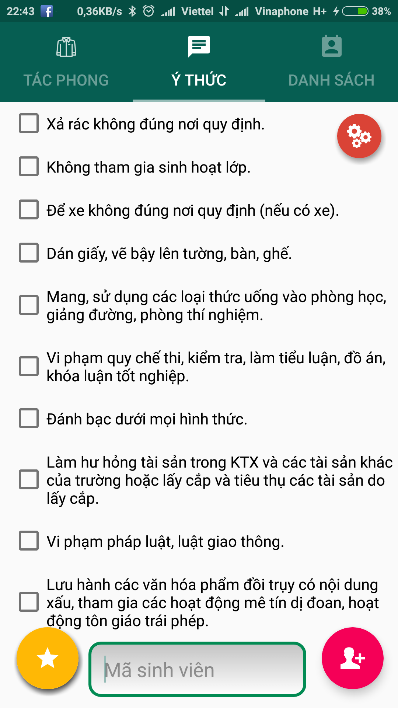
Hình 3. 12 Giao diện chính

**Giao diện chọn lỗi tác phong**



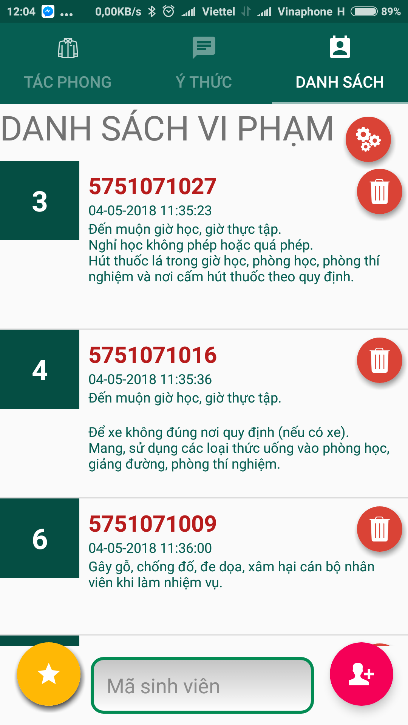
Hình 3. 13 Giao diện chọn lỗi tác phong

### Giao diện chọn lỗi ý thức



Hình 3. 14 Giao diện chọn lỗi ý thức

### Giao diện danh sách lỗi vi phạm của sinh viên đã ghi nhận được



Hình 3. 15 Giao diện danh sách lỗi vi phạm của sinh viên đã ghi nhận được

### Giao diện xem danh sách ghi nhận vi phạm theo ngày

****

Hình 3. 16 Giao diện xem danh sách ghi nhận vi phạm theo ngày

# CHƯƠNG 4: KHẢO SÁT, KẾT LUẬN, KIẾN NGHỊ VÀ DỰ KIẾN HƯỚNG NGHIÊN CỨU TIẾP THEO

## 4.1 Thực nghiệm

- Nhóm đã tiến hành thực nghiệm ứng dụng trong việc quét thẻ với 20 đối tượng khác nhau. Kết quả thực nghiệm cụ thể như sau:

**Tỉ lệ quét thành công trên 20 đối tượng**

|  |  |
| --- | --- |
| **Quét thẻ** | **Tỉ lệ** |
| Thành công | 100% |
| Không thành công | 0% |

Bảng 4. 1 Tỉ lệ thành công

**Tốc độ quét mã trên 20 đối tượng**

|  |  |
| --- | --- |
| **Thời gian (giây)** | **Số đối tượng** |
| 1 | 3 |
| 2 | 5 |
| 3 | 7 |
| 4 | 3 |
| 5 | 2 |

Bảng 4. 2 Thống kê thời gian quét

Hình 4. 1 Thống kê thời gian quét

Tốc độ quét trung bình xấp xỉ : 2,8 giây/người

## 4.2 Kết quả

Sau một thời gian nghiên cứu và thực hiện, nhóm đã sử dụng ngôn ngữ lập trình Java và sự hỗ trợ của một vài thư viện để xử lý dữ liệu được import vào từ file Excel cũng như việc export ra file Excel từ dữ liệu đã thu thập, ghi nhận được để thuận tiện cho việc thống kê, xử lý sau này cho người sử dụng.

Bên cạnh đó, sử dụng SQLite để xây dựng cơ sở dữ liệu nhằm phục vụ cho việc lưu trữ dữ liệu ghi nhận được vào thiết bị di động thông minh. Và để tạo giao diện thân thiện cho người sử dụng nhóm đã sử dụng ngôn ngữ lập trình Android nhằm phục vụ cho mục đích này.

Và cuối cùng, nhóm đã nghiên cứu và xây dựng thành công ứng dụng ghi ghận và quản lý vi phạm của sinh viên thông qua Barcode API chạy trên hệ điều hành Android, có thể chạy tốt và áp dụng được trong thực tế.

## 4.3 Kết luận

### 4.3.1 Ưu điểm

Ưu điểm lớn nhất của đề tài này là chúng ta có thể áp dụng thực tiễn tại nhà trường. Giúp các cán bộ, công nhân viên có thể dễ dàng ghi nhận lỗi vi phạm của sinh viên trong trường. Hạn chế tối đa sai sót trong lúc ghi nhận cũng như khó khăn trong việc tra cứu thông tin.

Ứng dụng dung lượng bé , không yêu cầu máy cấu hình quá hiện đại, có thể cài đặt ứng dụng cho bất cứ điện thoại thông minh nào có sử dụng hệ điều hành Android phiên bản 4.4 trở lên, kích hoạt nhanh chóng với giao diện thân thiện, dễ sử dụng.

Là ứng dụng gốc nên trong quá trình sử dụng không bị bên thứ 3 can thiệp, sửa đổi, đánh cắp thông tin, dữ liệu quan trọng (nhờ hệ thống đăng nhập khi muốn sử dụng các chức năng của ứng dụng).

Đề tài có thể mở rộng tùy vào nhu cầu của mỗi cơ quan, công ty, cá nhân.

### 4.3.2 Nhược điểm

Vì chương trình sử dụng bộ nhớ của thiết bị di dộng thông minh nên trong quá trình người dùng sử dụng, nếu có một cở sở dữ liệu lớn và nhiều thông tin thì khi import vào ứng dụng sẽ hao tốn bộ nhớ máy, với cơ sở dữ liệu lý tưởng là 1000 thông tin cá nhân. Ngoài ra việc sử dụng lâu dài ứng dụng cũng làm tăng dung lượng bộ nhớ của thiết bị di động thông minh trong quá trình ghi nhận, xử lý.

Ứng dụng chỉ chạy được trên thiết bị di động thông minh có cài đặt hệ điều hành Android 5.1 trở lên và hỗ trợ camera 5 chấm trở lên . Ngoài ra những hệ điều hành khác chưa thể cài đặt và sử dụng.

Và khi chèn file thông tin vào ứng dụng thì ứng dụng chỉ xử lý được một file được thêm vào một phiên làm việc; và đối với file có lượng lớn thông tin sinh viên thì tốc độ đọc file khá chậm. Ngoài ra, ứng dụng không thể hiển thị thông tin của những sinh viên không có trong file thông tin đã được nhập vào trước đó. Tuy nhiên việc ghi nhận vi phạm của sinh viên hoàn toàn không bị ảnh hưởng.

## 4.4. Hướng phát triển

Vì thời gian và kinh phí có hạn, nên đề tài chỉ mới thực hiện trên thiết bị di động thông minh chưa cần kết nối internet. Khi đã hoàn thiện đề tài, nhóm đang mong muốn sẽ phát triển đề tài này để tiện lợi hơn nữa, tự động hóa hơn nữa.

Cụ thể là, nhóm đang muốn mở rộng đề tài này thay vì chỉ ghi nhận lỗi vi phạm của sinh viên thì thay vào đó là điểm danh, kiểm kê hàng hóa, kiểm tra thông tin cá nhân khi ra vào cơ quan, kiểm tra thẻ giả, hoặc đăng ký tham gia một hoạt động bất kì, kiểm tra thi hộ trong thi cử bằng hình thức so sánh thông tin với cơ sở dữ liệu … chỉ bằng cái quẹt thẻ chưa đến 2 giây, trên bất cứ đối tượng nào có mã vạch Barcode.

Ngoài ra để giải quyết vấn đề bộ nhớ cho thiết bị di động thông minh, nhóm sẽ thay thế SQLite thành Server và áp dụng cho các thiết bị di động thông minh qua kết nối internet. Bằng cách đó, không chỉ vấn đề bộ nhớ thiết bị được giải quyết triệt để mà còn mở rộng hơn về cơ sở dữ liệu (không còn giới hạn ở 1000 sinh viên như trước nữa mà là nắm giữ cơ sở dữ liệu về toàn bộ sinh viên theo học ở trường), giúp việc quản lý, kiểm tra trở nên chặt chẽ và hiệu quả hơn bao giờ hết.

# TÀI LIỆU THAM KHẢO

[1] Rick Boyer, Kyle Mew. 2016. Android Application Development Cookbook, 2nd Edition

[2] Integrate ZXing in Android Studio

<https://stackoverflow.com/questions/18543668/integrate-zxing-in-android-studio>

<https://github.com/pethoalpar/ZxingExample>

Ngày truy cập (15/3/2018)

[3] How To Read Excel File From Assets Folder Android Studio

<https://www.youtube.com/watch?v=9iTRAwCazw8>

Ngày truy cập (25/3/2018)

[4] (24 Video dạy học lập trình android căn bản miễn phí by ThangCoder.Com)

<https://www.youtube.com/playlist?list=PLTCWqd1NHCAErzFPuCSnRvLN36rfC6q7r>

Ngày truy cập (6/3/2018)

[5] (14 Video hướng dẫn lập trình android nâng cao by Thangcoder.com)

<https://www.youtube.com/playlist?list=PLTCWqd1NHCAEXQmXaG8iy2pe9bTqRgQ4O>

Ngày truy cập (8/3/2018)

[6] Mã vạch

<https://vi.wikipedia.org/wiki/M%C3%A3_v%E1%BA%A1ch>

[7] Giao diện lập trình ứng dụng

<https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_l%E1%BA%ADp_tr%C3%ACnh_%E1%BB%A9ng_d%E1%BB%A5ng>

[8] SQLite

<http://vietjack.com/sqlite/sqlite_la_gi.jsp>

[9] Java

<https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)>

[10] Đặc điểm Java

<https://anphanhv.wordpress.com/2015/03/21/tong-quan-ngon-ngu-java/>

[11] Android

<https://vi.wikipedia.org/wiki/Android_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)>