TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO GIỮA KỲ**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG**

**TÌM HIỂU FIREBASE FIRESTORE VÀ ỨNG DỤNG QUẢN LÝ THÔNG TIN HỌC SINH**

*Người hướng dẫn*: **GV VÕ VĂN THÀNH**

*Người thực hiện*: **TRẦN TẤN THÀNH – 52100110**

Khoá**: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO GIỮA KỲ**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG**

**TÌM HIỂU FIREBASE FIRESTORE VÀ ỨNG DỤNG QUẢN LÝ THÔNG TIN HỌC SINH**

*Người hướng dẫn*: **GV VÕ VĂN THÀNH**

*Người thực hiện*: **TRẦN TẤN THÀNH – 52100110**

Khoá**: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Em chân thành cảm ơn Thầy Võ Văn Thành, người thầy tận tâm và kiến thức sâu rộng trong môn học Phát Triển Ứng Dụng Di Động.

Sự nhiệt huyết và tận tụy của Thầy không chỉ truyền đạt kiến thức mà còn truyền cả niềm đam mê với lĩnh vực này. Những hướng dẫn chi tiết, sự kiên nhẫn và sự cống hiến không ngừng nghỉ của Thầy đã đồng hành cùng em qua từng bước tiến trên con đường học tập.

Nhận được sự hỗ trợ và định hướng từ Thầy, em đã có cơ hội học hỏi và trải nghiệm những kiến thức thực tiễn, đồng thời phát triển khả năng sáng tạo và xây dựng ứng dụng di động một cách thành thạo. Em xin gửi lời cảm ơn sâu sắc đến Thầy Võ Văn Thành vì sự dẫn dắt và tạo điều kiện cho chúng em có một hành trình học tập ý nghĩa và đầy ý tưởng mới.

*TP. Hồ Chí Minh, ngày 30 tháng 11 năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

Thành

Trần Tấn Thành

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của GV. Võ Văn Thành. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 30 tháng 11 năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Thành*

*Trần Tấn Thành*

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vi](#_Toc152206431)

[CHƯƠNG 1. GIỚI THIỆU VỀ FIREBASE FIRESTORE 1](#_Toc152206432)

[1.1 Firebase 1](#_Toc152206433)

[1.2 Realtime Database 1](#_Toc152206434)

[1.3 Cloud Firestore 2](#_Toc152206435)

[1.4 Authentication 2](#_Toc152206436)

[1.5 Cloud Storage 3](#_Toc152206437)

[CHƯƠNG 2. QUẢN LÝ THÔNG TIN HỌC SINH 4](#_Toc152206438)

[2.1 Đặc tả yêu cầu 4](#_Toc152206439)

[2.2 Sơ đồ thực thể - quan hệ (ERD) mức vật lý 4](#_Toc152206440)

[2.3 Hiện thực ứng dụng 6](#_Toc152206441)

[2.3.1 Chức năng đăng nhập 6](#_Toc152206442)

[2.3.2 Chức năng đổi ảnh đại diện và cập nhật thông tin người dùng 7](#_Toc152206443)

[2.3.3 Chức năng xem danh sách người dùng trên hệ thống 9](#_Toc152206444)

[2.3.4 Chức năng thêm một người dùng mới 11](#_Toc152206445)

[2.3.5 Chức năng xóa người dùng 12](#_Toc152206446)

[2.3.6 Chức năng xem lịch sử đăng nhập của người dùng 12](#_Toc152206447)

[2.3.7 Chức năng xem danh sách học sinh 13](#_Toc152206448)

[2.3.8 Chức năng thêm một học sinh mới 14](#_Toc152206449)

[2.3.9 Chức năng xóa một học sinh 14](#_Toc152206450)

[2.3.10 Chức năng cập nhật thông tin học sinh 15](#_Toc152206451)

[2.3.11 Chức năng sắp xếp danh sách học sinh 16](#_Toc152206452)

[2.3.12 Chức năng tìm kiếm học sinh 18](#_Toc152206453)

[2.3.13 Chức năng xem danh sách chứng chỉ 19](#_Toc152206454)

[2.3.14 Chức năng thêm chứng chỉ mới cho học sinh 19](#_Toc152206455)

[2.3.15 Chức năng xóa chứng chỉ 20](#_Toc152206456)

[2.3.16 Chức năng thêm danh sách học sinh từ File 20](#_Toc152206457)

[2.3.17 Chức năng xuất danh sách học sinh ra File 23](#_Toc152206458)

[2.3.18 Chức năng thêm danh sách chứng chỉ từ File 24](#_Toc152206459)

[2.3.19 Chức năng xuất danh sách chứng chỉ ra File 25](#_Toc152206460)

[CHƯƠNG 3. DEMO 26](#_Toc152206461)

[3.1 Chức năng đăng nhập 26](#_Toc152206462)

[*3.2* Chức năng đổi ảnh đại diện và cập nhật thông tin người dùng 27](#_Toc152206463)

[3.3 Chức năng xem danh sách người dùng trên hệ thống 28](#_Toc152206464)

[3.4 Chức năng thêm một người dùng mới 29](#_Toc152206465)

[3.5 Chức năng xóa người dùng 30](#_Toc152206466)

[3.6 Chức năng xem lịch sử đăng nhập của người dùng 31](#_Toc152206467)

[3.7 Demo chức năng xem danh sách học sinh 32](#_Toc152206468)

[3.8 Demo chức năng thêm một học sinh mới 33](#_Toc152206469)

[3.9 Demo chức năng xóa một học sinh 34](#_Toc152206470)

[3.10 Demo chức năng cập nhật thông tin học sinh 35](#_Toc152206471)

[3.11 Demo chức năng sắp xếp danh sách học sinh 36](#_Toc152206472)

[3.12 Demo chức năng tìm kiếm học sinh 37](#_Toc152206473)

[3.13 Demo chức năng xem danh sách chứng chỉ 38](#_Toc152206474)

[3.14 Demo chức năng thêm chứng chỉ 39](#_Toc152206475)

[3.15 Demo chức năng xóa chứng chỉ 39](#_Toc152206476)

[TÀI LIỆU THAM KHẢO 41](#_Toc152206477)

# DANH MỤC HÌNH VẼ

[Hình 1 Sơ đồ thực thể - quan hệ (ERD) mức vật lý 5](#_Toc152206478)

[Hình 2 Đoạn code thực hiện chức năng đăng nhập (1) 7](#_Toc152206479)

[Hình 3 Đoạn code thực hiện chức năng đăng nhập (2) 7](#_Toc152206480)

[Hình 4 Đoạn code cập nhật thông tin người dùng (1) 8](#_Toc152206481)

[Hình 5 Đoạn code cập nhật thông tin người dùng (2) 9](#_Toc152206482)

[Hình 6 Đoạn code lấy danh sách người dùng trong hệ thống 10](#_Toc152206483)

[Hình 7 Đoạn code bắt sự kiện thay đổi trên collection 10](#_Toc152206484)

[Hình 8 Đoạn code thêm người dùng mới (1) 11](#_Toc152206485)

[Hình 9 Đoạn code thêm người dùng mới (2) 11](#_Toc152206486)

[Hình 10 Đoạn code xóa người dùng 12](#_Toc152206487)

[Hình 11 Đoạn code xem lịch sử đăng nhập (1) 13](#_Toc152206488)

[Hình 12 Đoạn code xem lịch sử đăng nhập (2) 13](#_Toc152206489)

[Hình 13 Đoạn code lấy danh sách học sinh 14](#_Toc152206490)

[Hình 14 Đoạn code thêm học sinh mới 14](#_Toc152206491)

[Hình 15 Đoạn code xóa một học sinh 15](#_Toc152206492)

[Hình 16 Đoạn code cập nhật thông tin học sinh 15](#_Toc152206493)

[Hình 17 Đoạn code hỗ trợ chức năng sắp xếp 16](#_Toc152206494)

[Hình 18 Đoạn code sắp xếp theo tên 17](#_Toc152206495)

[Hình 19 Đoạn code sắp xếp theo ngày sinh 17](#_Toc152206496)

[Hình 20 Đoạn code sắp xếp theo lớp 18](#_Toc152206497)

[Hình 21 Đoạn code hỗ trợ tìm kiếm học sinh 18](#_Toc152206498)

[Hình 22 Đoạn code tìm kiếm học sinh 19](#_Toc152206499)

[Hình 23 Đoạn code lấy danh sách chứng chỉ 19](#_Toc152206500)

[Hình 24 Đoạn code thêm chứng chỉ mới cho học sinh 20](#_Toc152206501)

[Hình 25 Đoạn code xóa chứng chỉ 20](#_Toc152206502)

[Hình 26 Đoạn code cấp quyền truy cập bộ nhớ 20](#_Toc152206503)

[Hình 27 Đoạn code gọi Intent để chọn File (1) 21](#_Toc152206504)

[Hình 28 Đoạn code gọi Intent để chọn File (2) 21](#_Toc152206505)

[Hình 29 Đoạn code thêm học sinh từ File (1) 22](#_Toc152206506)

[Hình 30 Đoạn code thêm học sinh từ File (2) 22](#_Toc152206507)

[Hình 31 Đoạn code xuất danh sách học sinh ra File (1) 23](#_Toc152206508)

[Hình 32 Đoạn code xuất danh sách học sinh ra File (2) 23](#_Toc152206509)

[Hình 33 Đoạn code thêm danh sách chứng chỉ từ File (1) 24](#_Toc152206510)

[Hình 34 Đoạn code thêm danh sách chứng chỉ từ File (2) 25](#_Toc152206511)

[Hình 35 Đoạn code xử lý xuất danh sách chứng chỉ ra File (1) 26](#_Toc152206512)

[Hình 36 Đoạn code xử lý xuất danh sách chứng chỉ ra File (2) 26](#_Toc152206513)

[Hình 37 Demo chức năng đăng nhập 27](#_Toc152206514)

[Hình 38 Demo chức năng đổi ảnh đại diện và cập nhật thông tin 28](#_Toc152206515)

[Hình 39 Demo chức năng xem danh sách người dùng 28](#_Toc152206516)

[Hình 40 Demo chức năng thêm người dùng mới 29](#_Toc152206517)

[Hình 41 Demo chức năng xóa người dùng 30](#_Toc152206518)

[Hình 42 Demo chức năng xem lịch sử đăng nhập 31](#_Toc152206519)

[Hình 43 Demo chức năng xem danh sách học sinh 32](#_Toc152206520)

[Hình 44 Demo chức năng thêm một học sinh mới 33](#_Toc152206521)

[Hình 45 Demo chức năng xóa học sinh 34](#_Toc152206522)

[Hình 46 Demo chức năng cập nhật thông tin học sinh 35](#_Toc152206523)

[Hình 47 Demo chức năng sắp xếp học sinh 36](#_Toc152206524)

[Hình 48 Demo chức năng tìm kiếm học sinh 37](#_Toc152206525)

[Hình 49 Demo chức năng xem danh sách chứng chỉ 38](#_Toc152206526)

[Hình 50 Demo chức năng thêm chứng chỉ 39](#_Toc152206527)

[Hình 51 Demo chức năng xóa chứng chỉ 40](#_Toc152206528)

# GIỚI THIỆU VỀ FIREBASE FIRESTORE

## Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web của Google, cung cấp các dịch vụ và công cụ giúp nhà phát triển xây dựng ứng dụng một cách nhanh chóng và hiệu quả. Nền tảng này được thiết kế để giúp giảm thiểu khả năng phức tạp trong việc quản lý cơ sở dữ liệu, xác thực người dùng, phân tích dữ liệu, và triển khai ứng dụng.

Firebase cung cấp các API thuận tiện, mạnh mẽ và tương thích trên nhiều nền tảng, giúp quản lý và sử dụng cơ sở dữ liệu trở nên đơn giản. Với Firebase, việc tương tác với cơ sở dữ liệu không còn phức tạp như trước; chúng ta có thể gọi các API này một cách dễ dàng và phần phức tạp của việc quản lý máy chủ đã được Firebase xử lý.

Một số dịch vụ phổ biến của Firebase như: Cloud Firestore, Authentication, Hosting, Cloud Storage, Realtime Database, …

## Realtime Database

Cơ sở dữ liệu thời gian thực (Realtime Database) của Firebase là một cơ sở dữ liệu NoSQL dựa trên cấu trúc JSON, được thiết kế để lưu trữ và đồng bộ dữ liệu thời gian thực giữa các thiết bị và máy chủ. Đây là một phần quan trọng trong hệ thống Firebase, cho phép các ứng dụng cập nhật dữ liệu và nhận thông báo ngay lập tức khi có sự thay đổi.

Cấu trúc dữ liệu: Realtime Database lưu trữ dữ liệu theo cấu trúc JSON, cho phép tổ chức dữ liệu thành các cặp key-value và các cấu trúc phức tạp hơn.

Sự đồng bộ thời gian thực: Dữ liệu được đồng bộ hóa ngay lập tức trên tất cả các thiết bị kết nối khi có thay đổi, giúp các ứng dụng hiển thị thông tin mới mà không cần phải làm mới trang.

Event Listener: Firebase cung cấp các event listener cho phép ứng dụng theo dõi và phản ứng khi dữ liệu thay đổi. Điều này cho phép các ứng dụng thực hiện các hành động phụ thuộc vào dữ liệu mà không cần phải kiểm tra thường xuyên.

Realtime database chấp nhận nhiều kiểu dữ liệu: String, Long, Double, Boolean, Map<String, Object>, List<Object> để lưu trữ. Chúng ta cũng có thể sử dụng custom java objects để lưu trữ dữ liệu.

## Cloud Firestore

Cloud Firestore là một cơ sở dữ liệu NoSQL linh hoạt và mạnh mẽ của Firebase. Nó cung cấp một cách tiếp cận tài liệu (document) và bộ sưu tập (collection) cho phép lưu trữ dữ liệu dễ dàng và phản hồi nhanh chóng. Đây là một bộ phận quan trọng trong hệ thống Firebase, được thiết kế để cung cấp tính linh hoạt cao và khả năng mở rộng tốt cho mobile, web và server.

Cũng giống như Firebase realtime database Cloud Firestore giúp cho việc đồng bộ dữ liệu giữa các ứng dụng phía client một các nhanh chóng (Realtime) và hộ trợ lưu offline data trong ứng dụng của bạn.

Cấu trúc dữ liệu linh hoạt: Firestore lưu trữ dữ liệu theo cấu trúc tài liệu và bộ sưu tập (collections-documents), cho phép tổ chức dữ liệu một cách hiệu quả hơn, tương tự như các bảng và hàng trong cơ sở dữ liệu quan hệ.

Đồng bộ thời gian thực: Dữ liệu được đồng bộ hóa ngay lập tức trên tất cả các thiết bị kết nối khi có thay đổi, cho phép các ứng dụng hiển thị thông tin mới mà không cần làm mới trang.

Đồng bộ hoá ngoại tuyến: Tính năng này của Firestore có nhiệm vụ thực hiện bộ nhớ đệm dữ liệu đang được dùng và cho phép ứng dụng có thể đọc, ghi, truy vấn và lắng nghe dữ liệu ngay cả khi thiết bị đang ngoại tuyến.

Truy vấn mạnh mẽ: Firestore hỗ trợ các truy vấn linh hoạt để truy xuất dữ liệu, cho phép lọc, sắp xếp và giới hạn dữ liệu trả về.

## Authentication

Nền tảng đa dạng: Firebase Authentication hỗ trợ xác thực trên nhiều nền tảng, bao gồm email/password, số điện thoại, mạng xã hội như Google, Facebook, Twitter, GitHub và nhiều hơn nữa.

Xác thực mạnh mẽ và an toàn: Cung cấp các phương thức xác thực an toàn để ngăn chặn truy cập trái phép hoặc không được ủy quyền vào ứng dụng.

Quản lý người dùng linh hoạt: Cho phép quản lý thông tin người dùng, thiết lập quyền truy cập và tùy chỉnh trải nghiệm người dùng.

Xác thực đa cấp độ: Firebase cung cấp các cấp độ xác thực khác nhau để phù hợp với yêu cầu cụ thể của ứng dụng.

## Cloud Storage

Cloud Storage trong Firebase cung cấp một dịch vụ lưu trữ đám mây mạnh mẽ, cho phép bạn lưu trữ và quản lý các tệp như hình ảnh, video, tệp âm thanh và các tài nguyên khác của ứng dụng của mình.

Lưu trữ tệp đa dạng: Cho phép lưu trữ nhiều loại tệp khác nhau với các định dạng và kích thước khác nhau.

An toàn và bảo mật: Dữ liệu được lưu trữ trên cơ sở hạ tầng đám mây an toàn và được bảo vệ bởi các biện pháp bảo mật nghiêm ngặt của Google.

Đa khu vực và sao lưu tự động: Dữ liệu được lưu trữ tại các trung tâm dữ liệu khắp thế giới, đồng thời có khả năng sao lưu tự động để đảm bảo an toàn và sẵn sàng.

Quản lý tệp từ Firebase Console và API: Firebase cung cấp giao diện người dùng trực quan qua Firebase Console và các API để quản lý, tải lên, tải xuống và xử lý tệp.

# QUẢN LÝ THÔNG TIN HỌC SINH

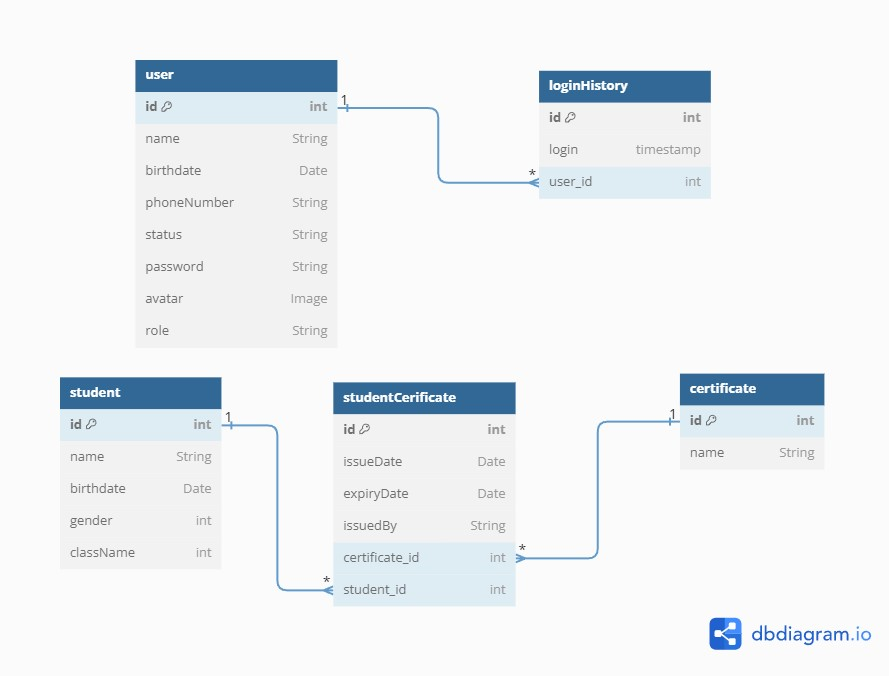
## Đặc tả yêu cầu

Ứng dụng cần xây dựng là một nền tảng quản lý người dùng và sinh viên, với khả năng quản lý tài khoản người dùng, thông tin cá nhân, và danh sách sinh viên một cách linh hoạt và tiện lợi. Người dùng sẽ có khả năng đăng nhập, thay đổi hình đại diện, và quản lý thông tin cá nhân của họ. Chức năng quản lý người dùng bao gồm việc xem, thêm mới, xóa, và chỉnh sửa thông tin của người dùng, cũng như xem lịch sử đăng nhập của họ.

Đối với quản lý sinh viên, ứng dụng sẽ cung cấp khả năng xem, thêm mới, xóa, cập nhật thông tin sinh viên, và thực hiện các chức năng sắp xếp và tìm kiếm dựa trên các tiêu chí khác nhau. Đi kèm với việc quản lý sinh viên là khả năng truy cập vào trang chi tiết của mỗi sinh viên, hiển thị thông tin đầy đủ và quản lý danh sách chứng chỉ cá nhân của họ, bao gồm xem, thêm mới, xóa, và cập nhật thông tin chứng chỉ.

Ứng dụng cũng hỗ trợ nhập và xuất dữ liệu từ file, cho phép người dùng nhập danh sách sinh viên hoặc danh sách chứng chỉ từ file, cũng như xuất danh sách sinh viên và chứng chỉ sang định dạng Excel hoặc CSV. Để quản lý quyền hạn, hệ thống cung cấp ba vai trò người dùng: admin, quản lý, và nhân viên. Admin có quyền thực hiện tất cả các chức năng, quản lý có thể quản lý sinh viên và chứng chỉ, trong khi nhân viên chỉ có thể xem nội dung và cập nhật hình đại diện cá nhân của họ. Tài khoản người dùng bình thường không được phép tự tạo tài khoản, chỉ admin mới có thể tạo mới các tài khoản khác.

## Sơ đồ thực thể - quan hệ (ERD) mức vật lý



Hình 1 Sơ đồ thực thể - quan hệ (ERD) mức vật lý

Bảng user sẽ lưu thông tin cơ bản của người dùng trong hệ thống như tên (name), ngày sinh (birthdate), số điện thoại (phoneNumber), trạng thái (status), mật khẩu (password), avatar và role (quyền hạn trong hệ thống).

Mỗi user sẽ có nhiều bản ghi lịch sử đăng nhập (login history), bảng lịch sử đang nhập chứa thông tin về thời gian đăng nhập và user\_id là khóa ngoại trỏ đến user.

Mỗi học sinh sẽ có các thông tin cơ bản như tên, ngày sinh, giới tính, lớp học hiện tại, mỗi học sinh sẽ có 1 hoặc nhiều chứng chỉ, thông tin về chứng chỉ bao gồm ngày thi, ngày hết hạn, đơn vị cấp.

## Hiện thực ứng dụng

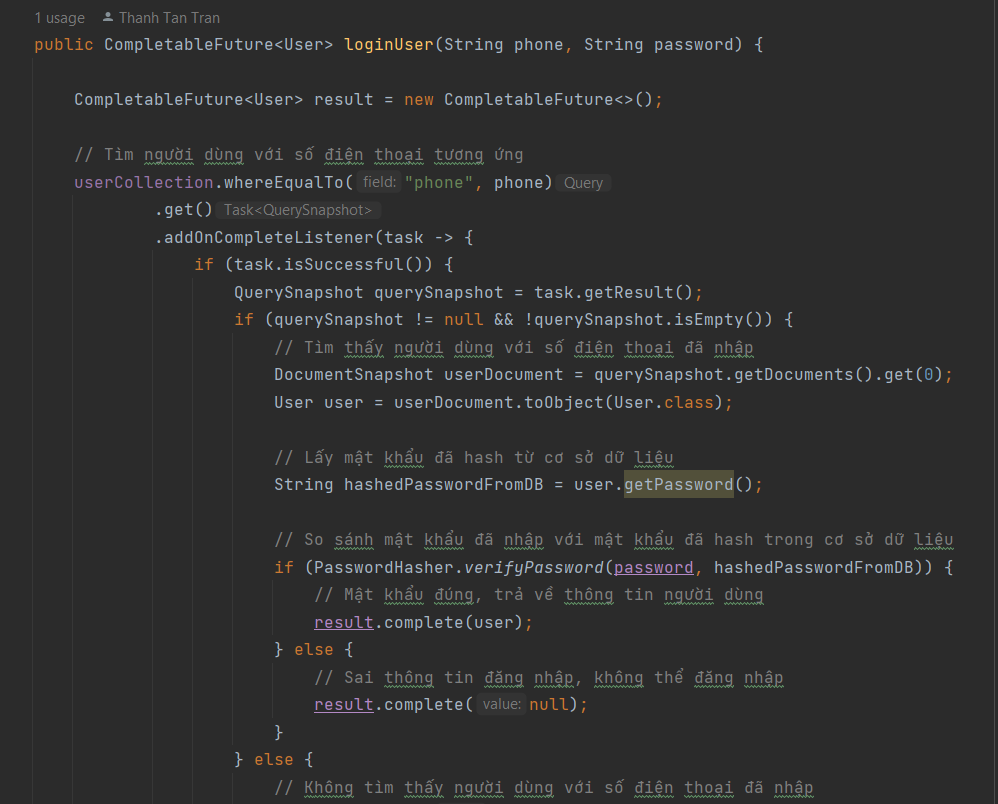
Em chọn sử dụng Cloud Firestore để làm backend lưu trữ dữ liệu thay vì Realtime Database bởi vì Cloud Firestore nhanh hơn, cấu trúc lưu trữ rõ ràng dễ đọc hơn so với Realtime Database.

### Chức năng đăng nhập

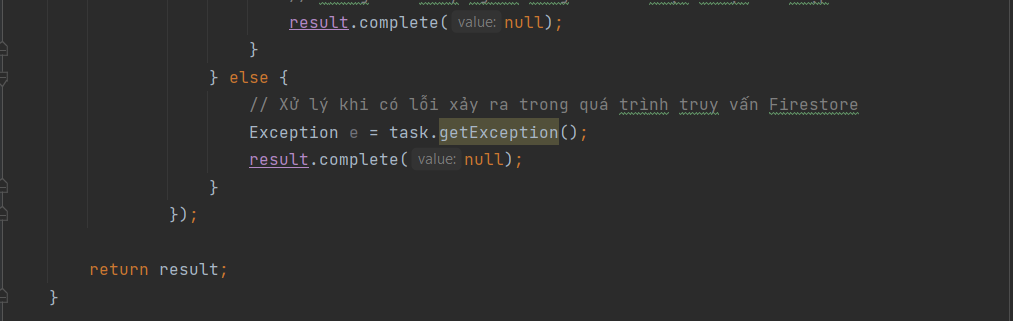
Để thực hiện chức năng đăng nhập, em chọn cách làm thủ công thay vì ứng dụng Authentication của Firebase.

Logic để xác thực: Đầu tiên sẽ tìm trên firestore xem phone đã tồn tại hay chưa, nếu phone đã tồn tại thì trả về null để báo rằng đăng nhập không thành công, ếu tìm thấy phone thì tiếp tục thực hiện kiểm tra mật khẩu, mật khẩu nhập vào phải trùng khớp với mật khẩu đã được lưu trên firestore, mật khẩu này đã được mã hóa.

Em sẽ dùng Completable để làm kiểu dữ liệu trả về cho hàm này, Completable là một lớp được thiết kế để xử lý các tác vụ bất đồng bộ. Đoạn code thực hiện chức năng đăng nhập như sau:



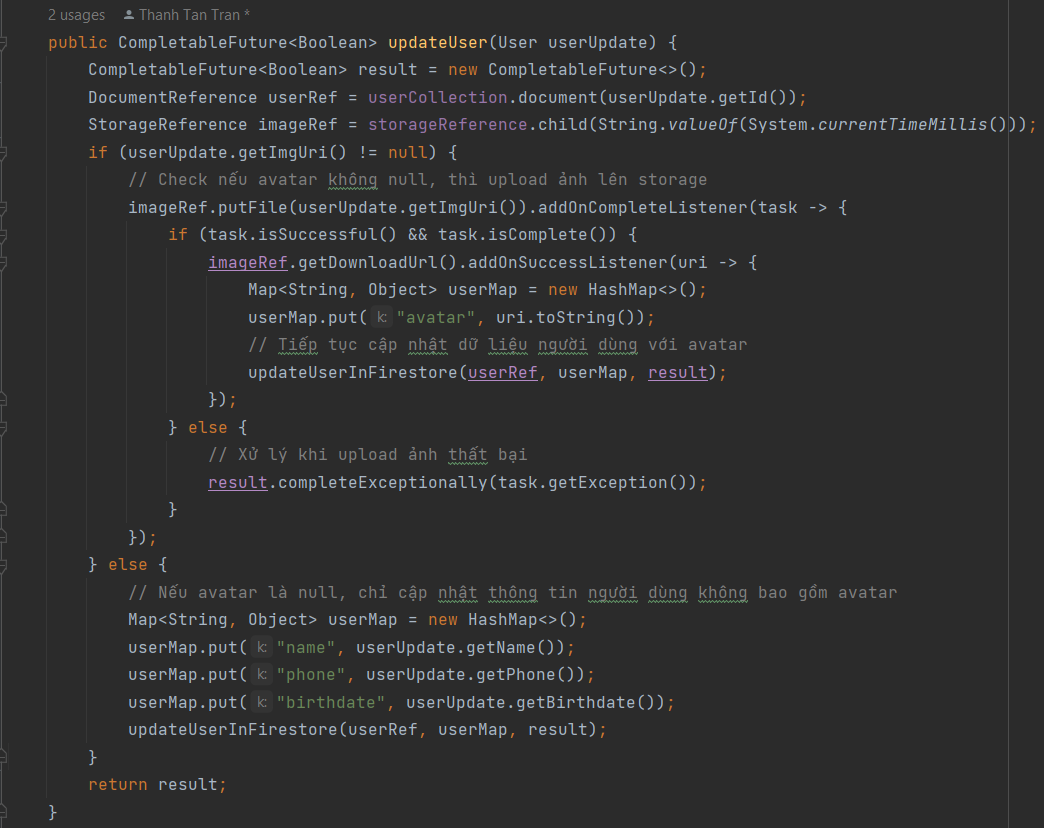
Hình 2 Đoạn code thực hiện chức năng đăng nhập (1)



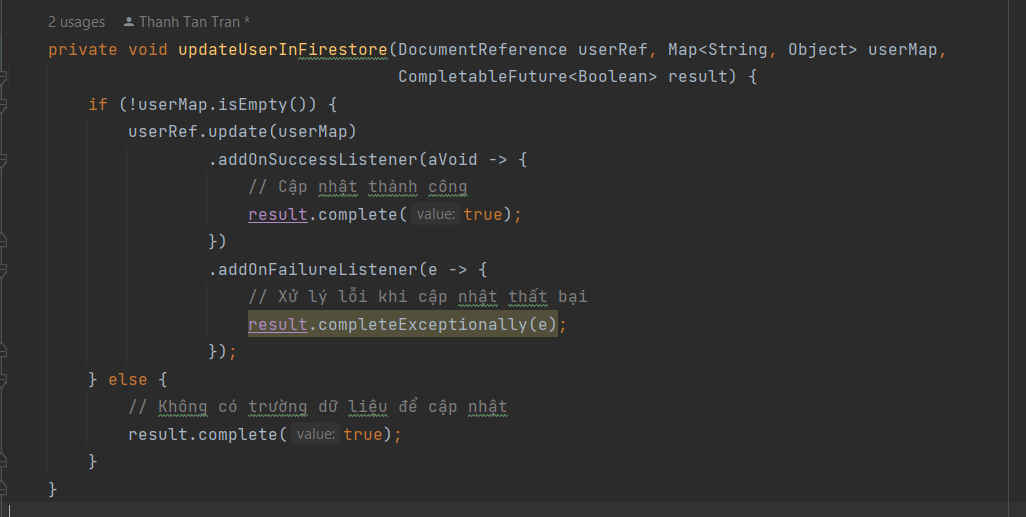
Hình 3 Đoạn code thực hiện chức năng đăng nhập (2)

### Chức năng đổi ảnh đại diện và cập nhật thông tin người dùng

Em thực hiện chức năng cập nhật thông tin người dùng và thay đổi ảnh đại diện chung vào một hàm, nếu dữ liệu gửi lên server gồm có Uri của ảnh mới, thì thực hiện lưu vào Firebase Storage sau đó Firebase Storage sẽ trả một callback về là một URI trỏ đến ảnh vừa thêm vào Firebase Storage, việc tiếp theo cần thực hiện đó là lưu URI này vào bản ghi User trên Firestore.



Hình 4 Đoạn code cập nhật thông tin người dùng (1)



Hình 5 Đoạn code cập nhật thông tin người dùng (2)

### Chức năng xem danh sách người dùng trên hệ thống

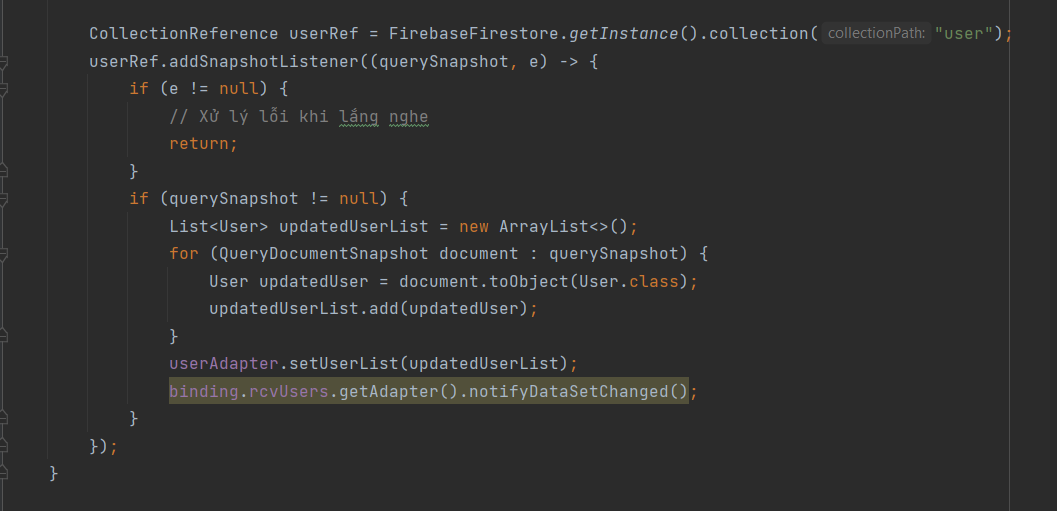
Để hiển thị được danh sách người dùng trên hệ thống thì cần phải có một Adapter đảm nhiệm vai trò chuyển các Object User sang thành các View tương ứng. Em xin phép không chụp hình về code Adapter ở đây do khá dài, Adapter cho User có tên là UserAdapter (thầy có thể xem qua trong thư mục adapter).

Sau khi đã có adapter thì điều quan trọng kế tiếp đó là lấy dữ liệu để hiển thị. Em sẽ gọi API của Firestore để lấy toàn bộ bản ghi, sau đó chuyển các bản ghi này thành Java Object bằng cách gọi method toObject, sau khi chuyển thành Java Object, em sẽ thêm Object này vào List và trả về kết quả khi đã hoàn thành lấy dữ liệu. Hàm lấy dữ liệu sẽ có dạng sau:



Hình 6 Đoạn code lấy danh sách người dùng trong hệ thống

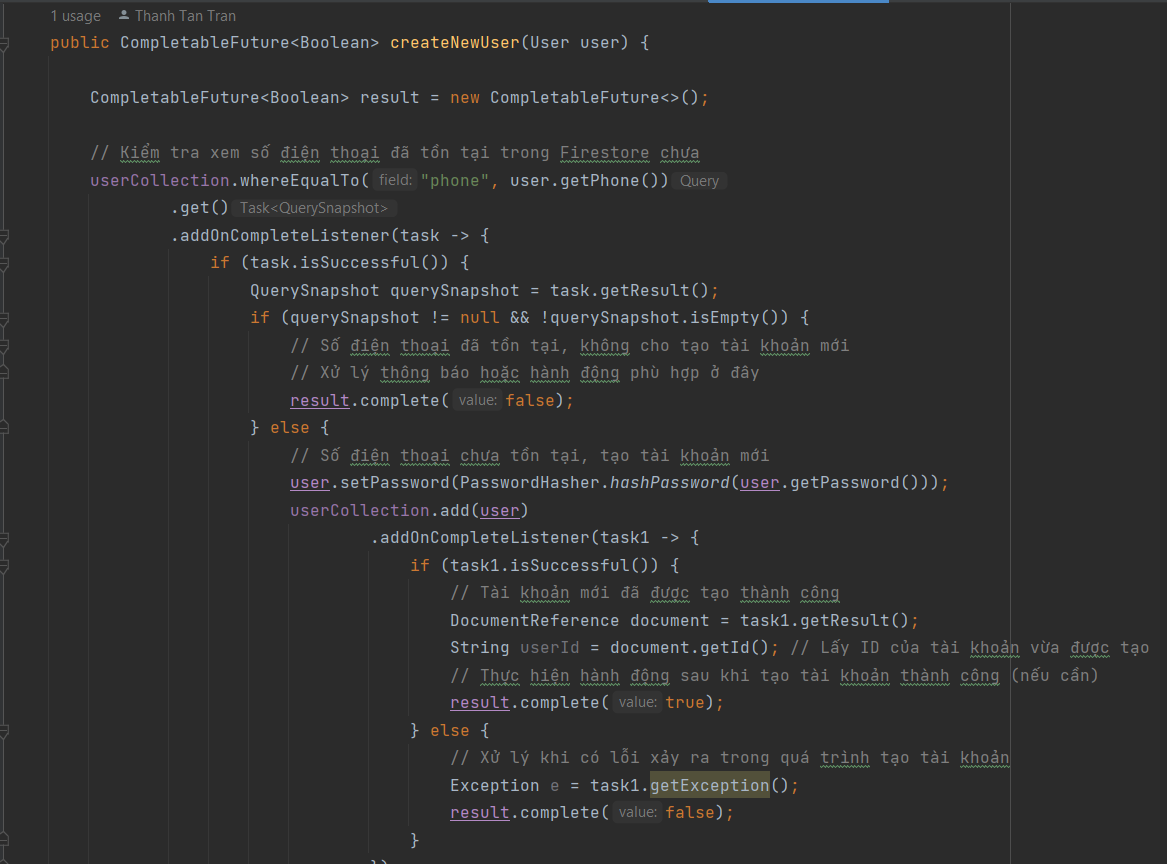
Firestore cũng cung cấp một API để theo dõi sự thay đổi từ Collection để cập nhật lại giao diện, do đó ở file Activity, em sẽ bắt sự kiện này để thực hiện render lại giao diện khi có thay đổi mới:



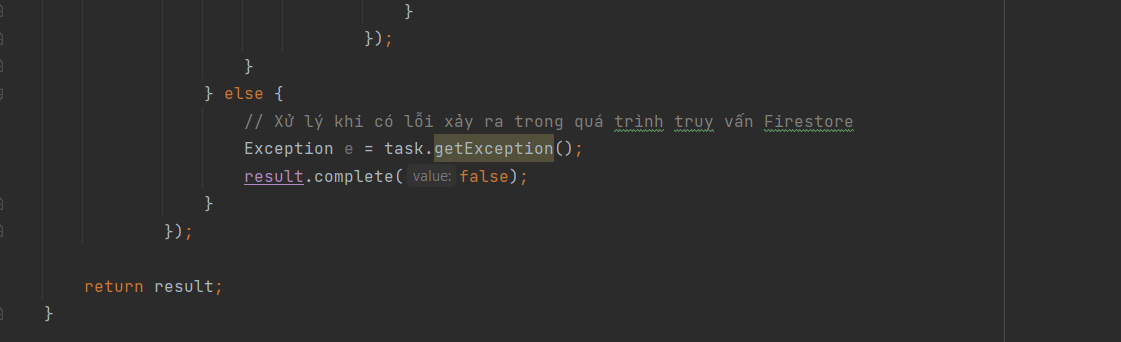
Hình 7 Đoạn code bắt sự kiện thay đổi trên collection

### Chức năng thêm một người dùng mới

Để thêm một người dùng mới thì sau khi lấy giá trị được nhập từ Admin, tất cả thông tin này sẽ được lưu vào Java Object là User mà em đã định nghĩa sẵn, sau đó gọi hàm để thêm. Nếu số điện thoại của người dùng chưa tồn tại thì hoàn toàn có thê thêm, nếu đã tồn tại không cho phép thêm.



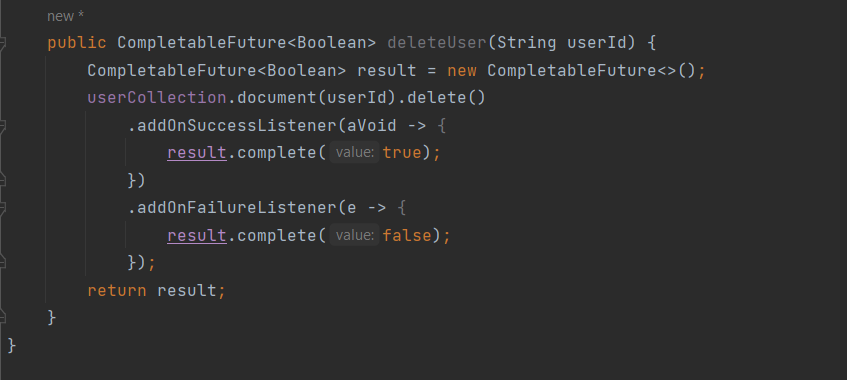
Hình 8 Đoạn code thêm người dùng mới (1)



Hình 9 Đoạn code thêm người dùng mới (2)

### Chức năng xóa người dùng

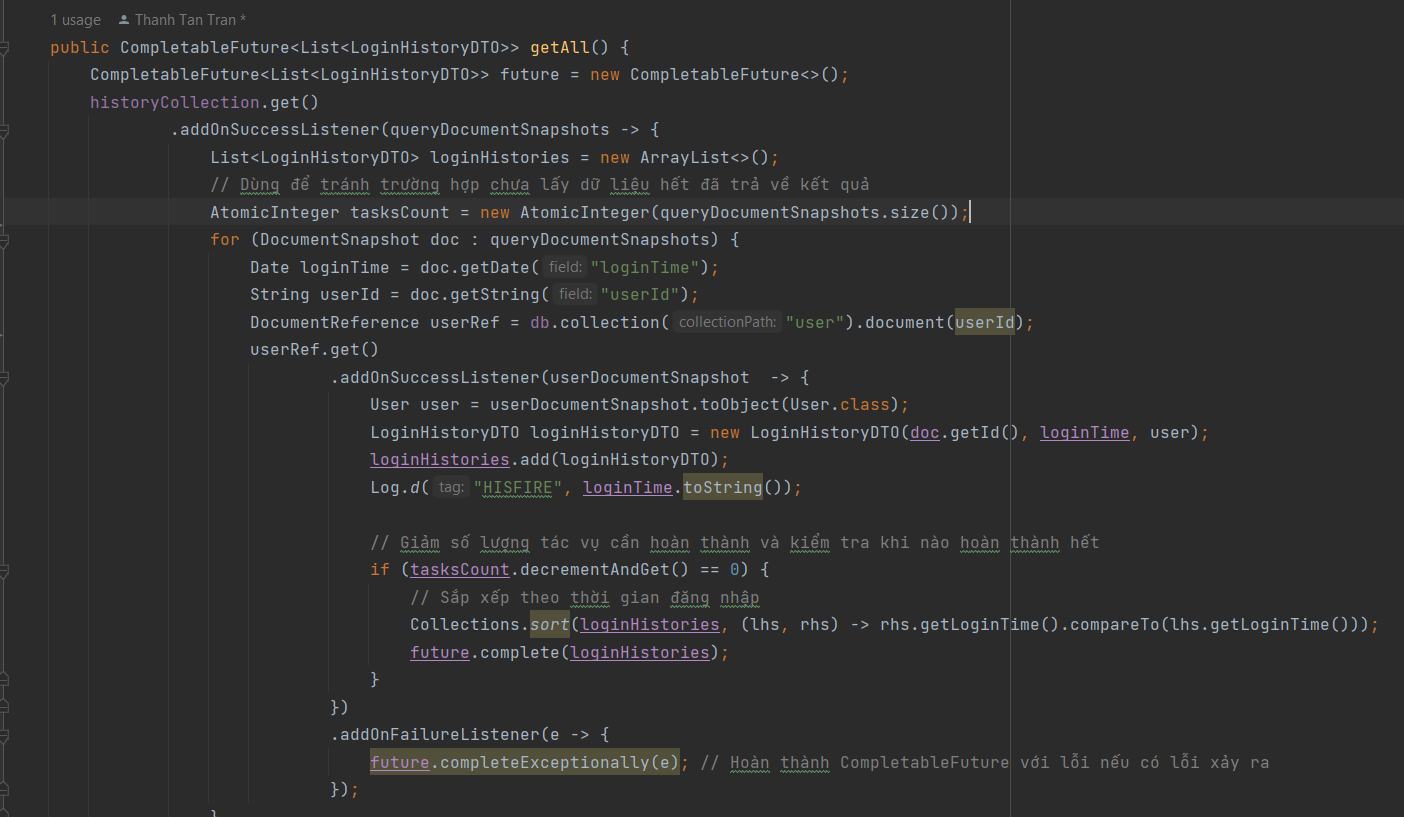
Để thực hiện chức năng này thì em sẽ tạo một hàm để nhận tham số đầu vào là id của user, sau đó thực hiện gọi API xóa của Firestore.



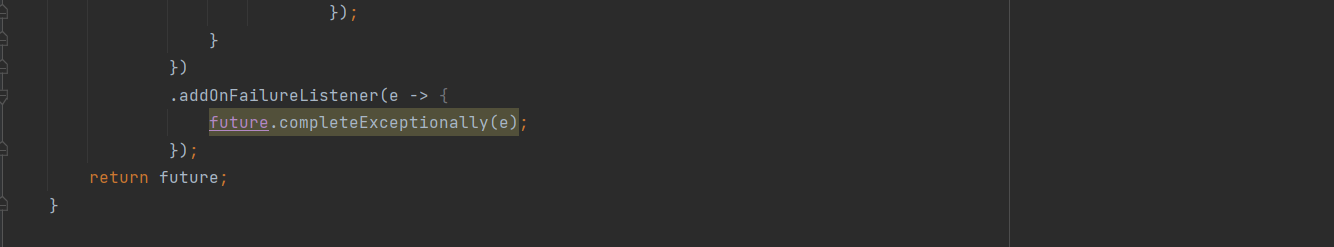
Hình 10 Đoạn code xóa người dùng

### Chức năng xem lịch sử đăng nhập của người dùng

Tương tự với xem danh sách người dùng, cần có một adapter để chuyển Object thành View, tên của adapter là LoginHistoryAdapter (thầy có thể xem trong thư mục adapter). Sau khi có adapter, việc tiếp theo là lấy danh sách các lịch sử.



Hình 11 Đoạn code xem lịch sử đăng nhập (1)



Hình 12 Đoạn code xem lịch sử đăng nhập (2)

### Chức năng xem danh sách học sinh

Tương tự với xem danh sách người dùng và lịch sử đăng nhập, tên adapter là StudentAdapter (thầy có thể xem trong thư mục adapter).

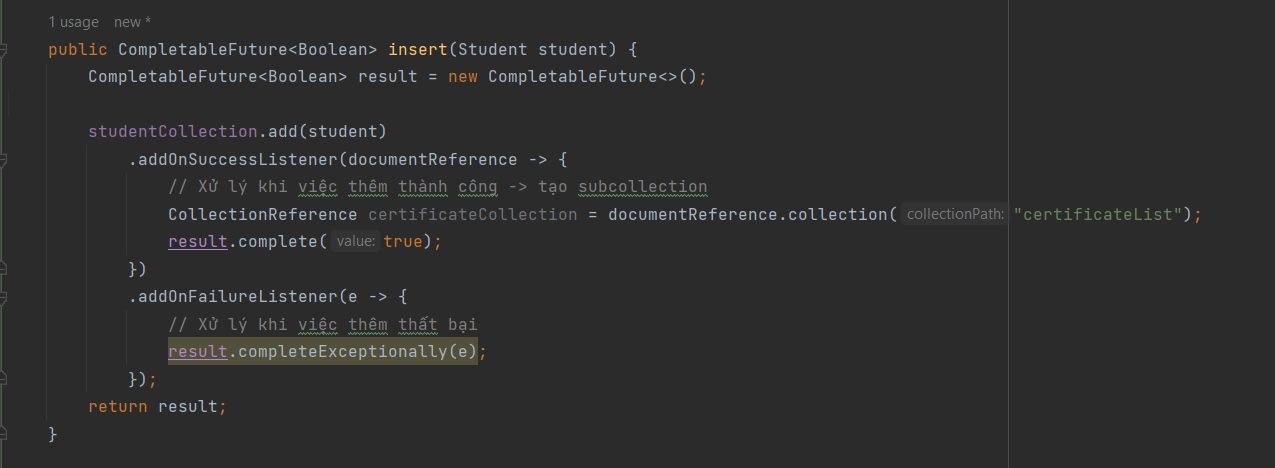
Thực hiện lấy danh sách học sinh từ Firestore:



Hình 13 Đoạn code lấy danh sách học sinh

### Chức năng thêm một học sinh mới

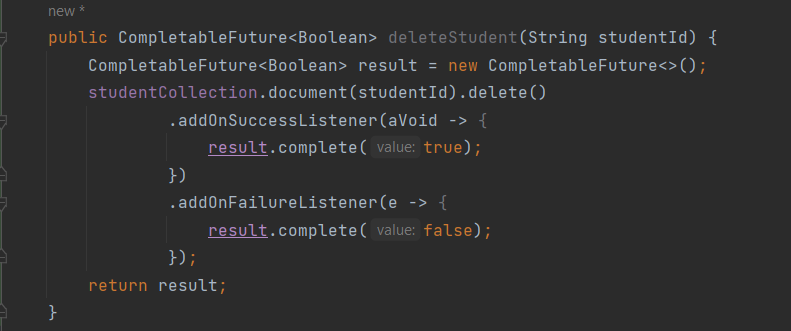
Để thêm một học sinh mới thì sau khi nhận được dữ liệu nhập vào từ người dùng thì gọi hàm thêm học sinh mới:



Hình 14 Đoạn code thêm học sinh mới

### Chức năng xóa một học sinh

Để xóa một học sinh thì cần phải truyền tham số là studentId, sau đó gọi API của Firestore để xóa học sinh.



Hình 15 Đoạn code xóa một học sinh

### Chức năng cập nhật thông tin học sinh

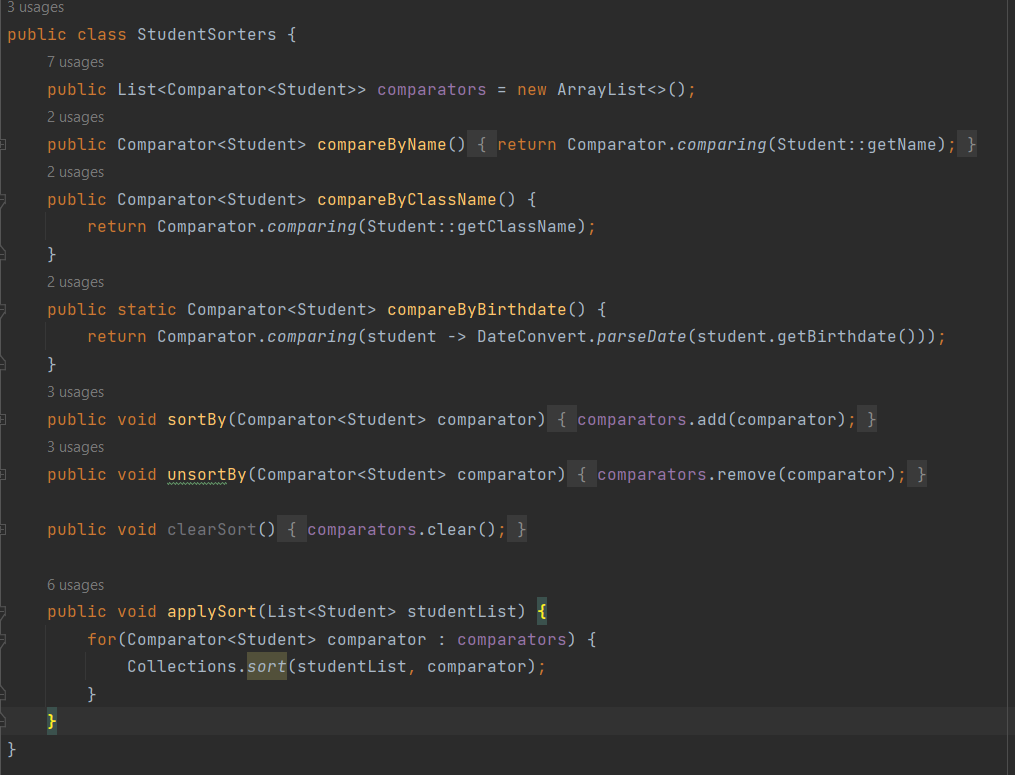
Để thực hiện chức năng này, sau khi người dùng nhập đầy đủ các thông tin cần cập nhật thì gọi đến hàm update này, hàm này sẽ gọi API để cập nhật lại thông tin của học sinh.



Hình 16 Đoạn code cập nhật thông tin học sinh

### Chức năng sắp xếp danh sách học sinh

Ở chức năng này, em thực hiện sắp xếp dựa trên danh sách đã có, em cung cấp xếp theo tên, theo lớp, và theo ngày sinh do đó để tiện em sẽ định nghĩa ra một lớp để hỗ trợ cho việc này:



Hình 17 Đoạn code hỗ trợ chức năng sắp xếp

Sau khi sắp xếp, cập nhật lại giao diện, dưới đây là đoạn code sắp xếp và cập nhật lại giao diện:



Hình 18 Đoạn code sắp xếp theo tên



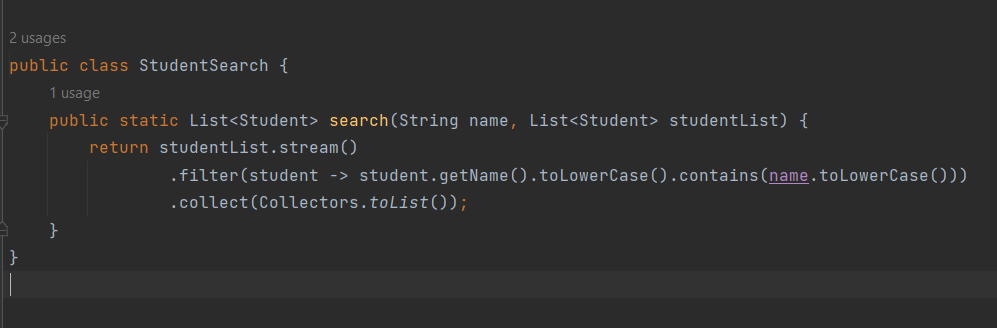
Hình 19 Đoạn code sắp xếp theo ngày sinh



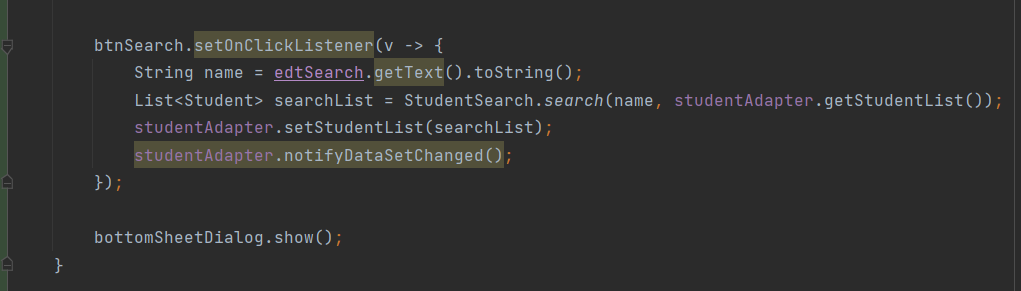
Hình 20 Đoạn code sắp xếp theo lớp

### Chức năng tìm kiếm học sinh

Ở chức năng này học sinh sẽ được tìm kiếm dựa trên tên, em cũng tìm dựa trên danh sách đã được lấy trước đó, kết quả trả về sẽ là danh sách các học sinh có tên gần giống với từ khóa nhập vào. Em định nghĩa ra một lớp riêng để hỗ trợ việc tìm kiếm này:



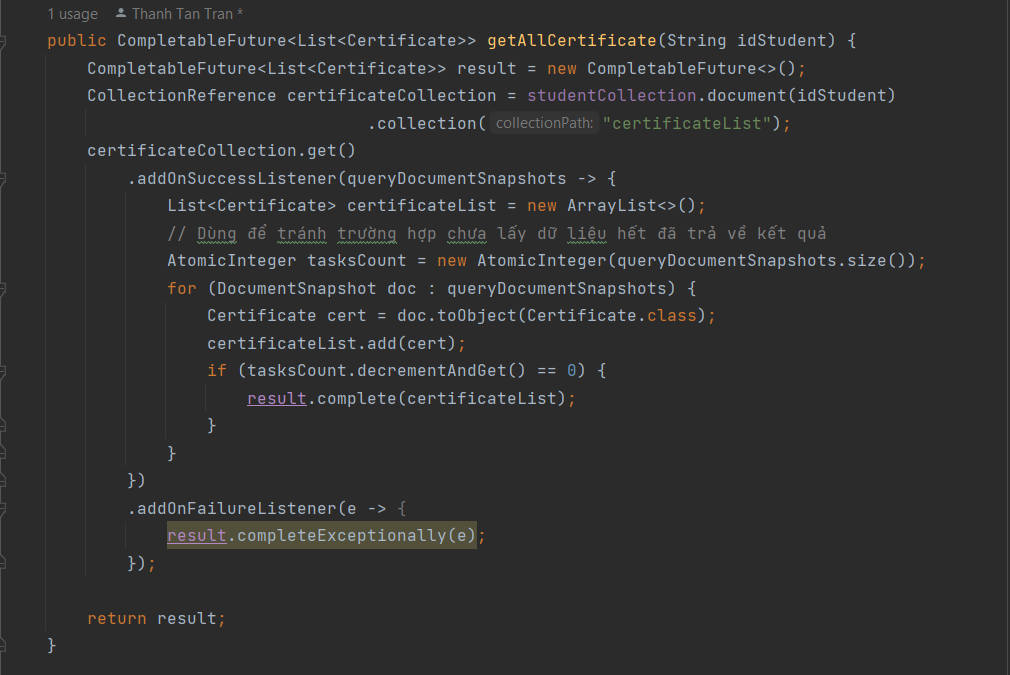
Hình 21 Đoạn code hỗ trợ tìm kiếm học sinh



Hình 22 Đoạn code tìm kiếm học sinh

### Chức năng xem danh sách chứng chỉ

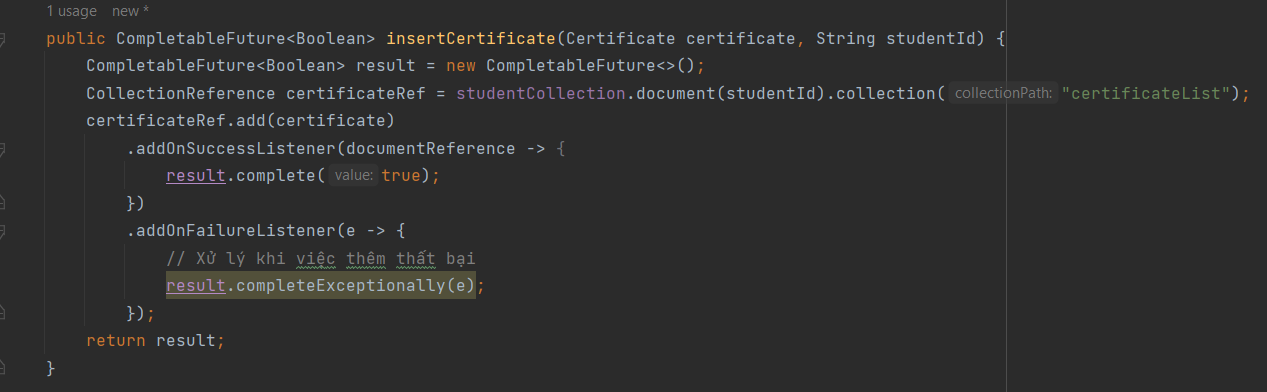
Tương tự với danh sách học sinh, cần có một adapter, adapter này có tên là CertificateAdapter. Sau đó gọi hàm để lấy danh sách các chứng chỉ theo studentId.



Hình 23 Đoạn code lấy danh sách chứng chỉ

### Chức năng thêm chứng chỉ mới cho học sinh

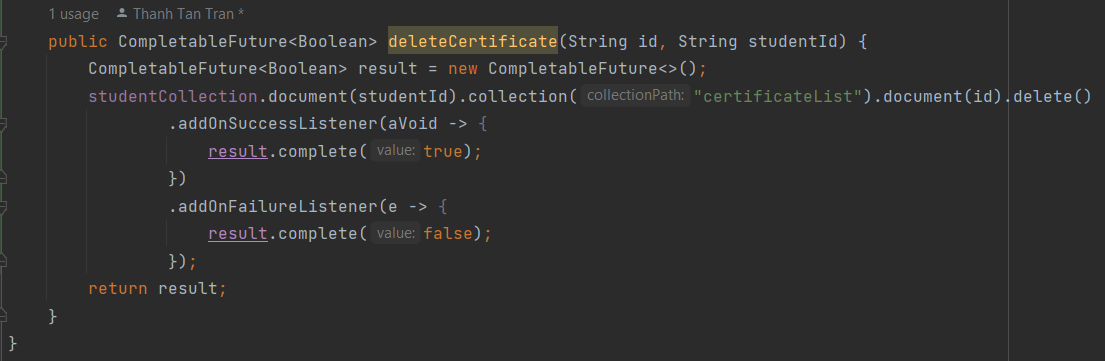
Để thực hiện chức năng này sau khi người dùng nhập các thông tin của chứng chỉ thì gửi thông tin này cùng với studentId thông qua hàm phía dưới.



Hình 24 Đoạn code thêm chứng chỉ mới cho học sinh

### Chức năng xóa chứng chỉ

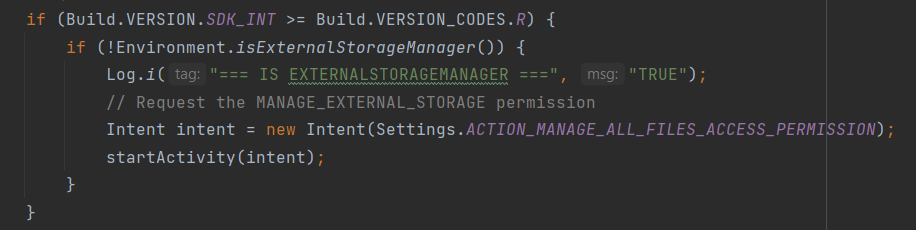
Hàm thực hiện xóa chứng chỉ, tham số truyền vào là id của chứng chỉ và studentId vì chứng chỉ thuộc về một học sinh nên phải truyền studentId để xóa.



Hình 25 Đoạn code xóa chứng chỉ

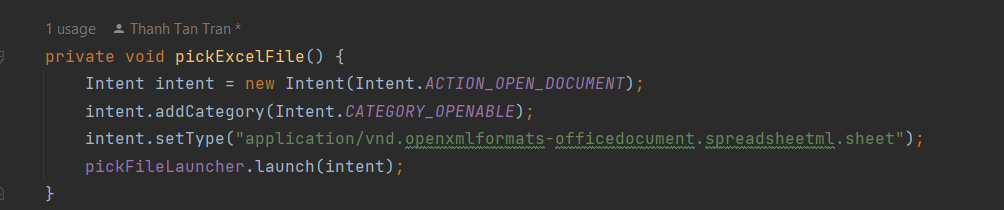
### Chức năng thêm danh sách học sinh từ File

Để thực hiện thêm từ File, đầu tiên phải thực hiện nhiệm vụ cấp quyền truy cập vào bộ nhớ để có thể đọc và ghi File:

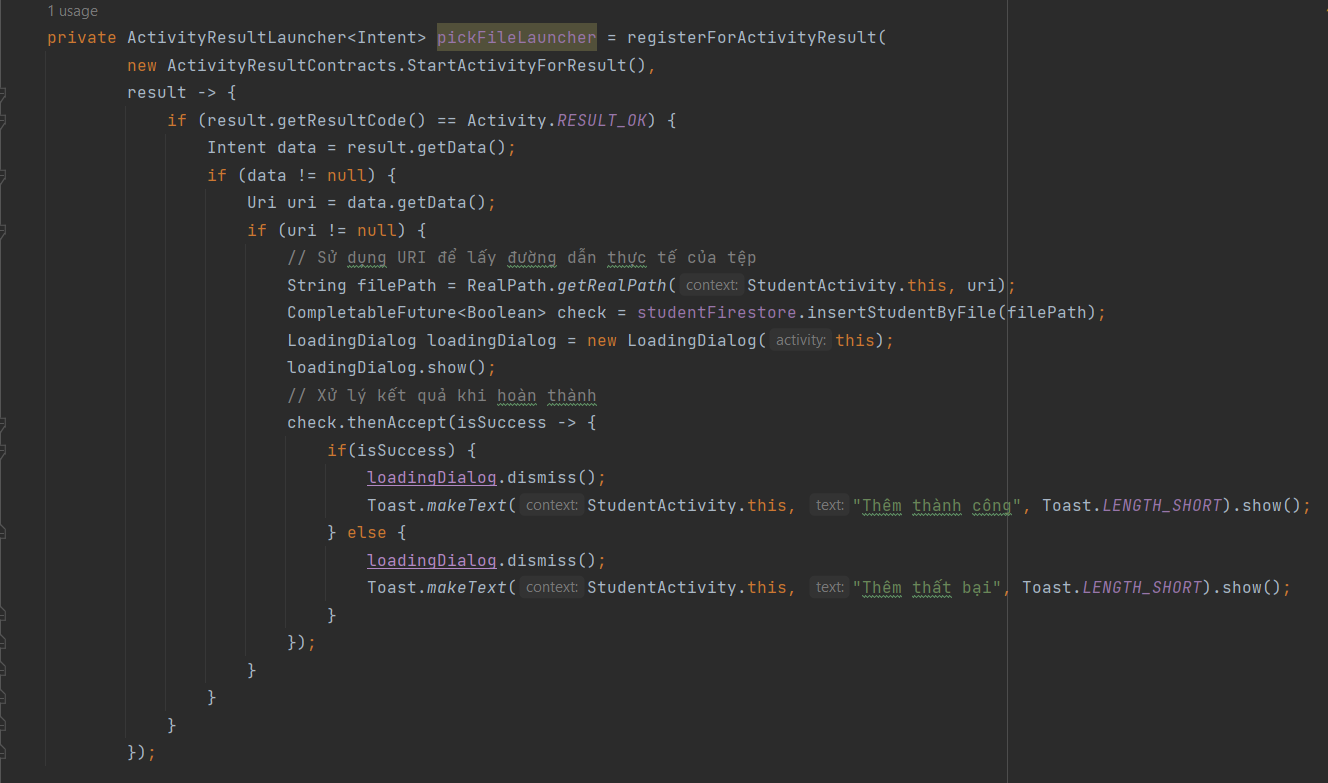


Hình 26 Đoạn code cấp quyền truy cập bộ nhớ

Hàm để mở một Intent cho phép người dùng chọn File Excel:

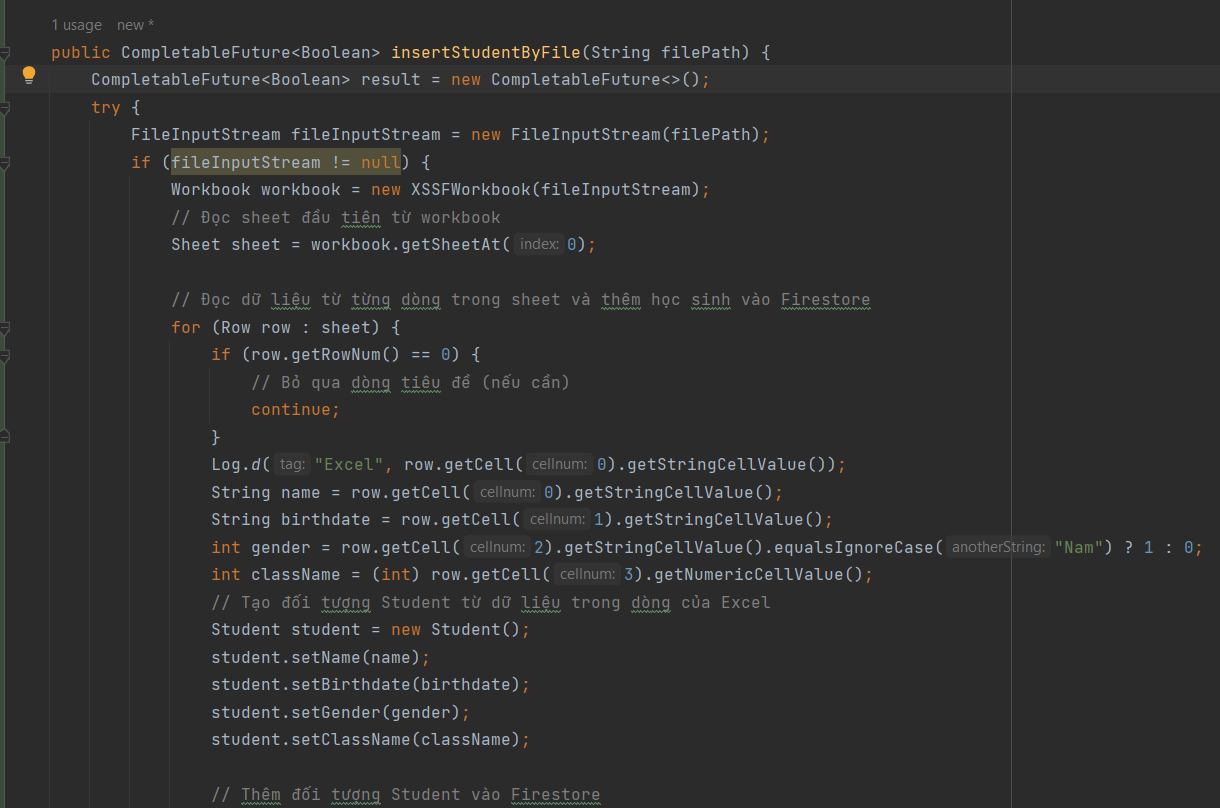


Hình 27 Đoạn code gọi Intent để chọn File (1)

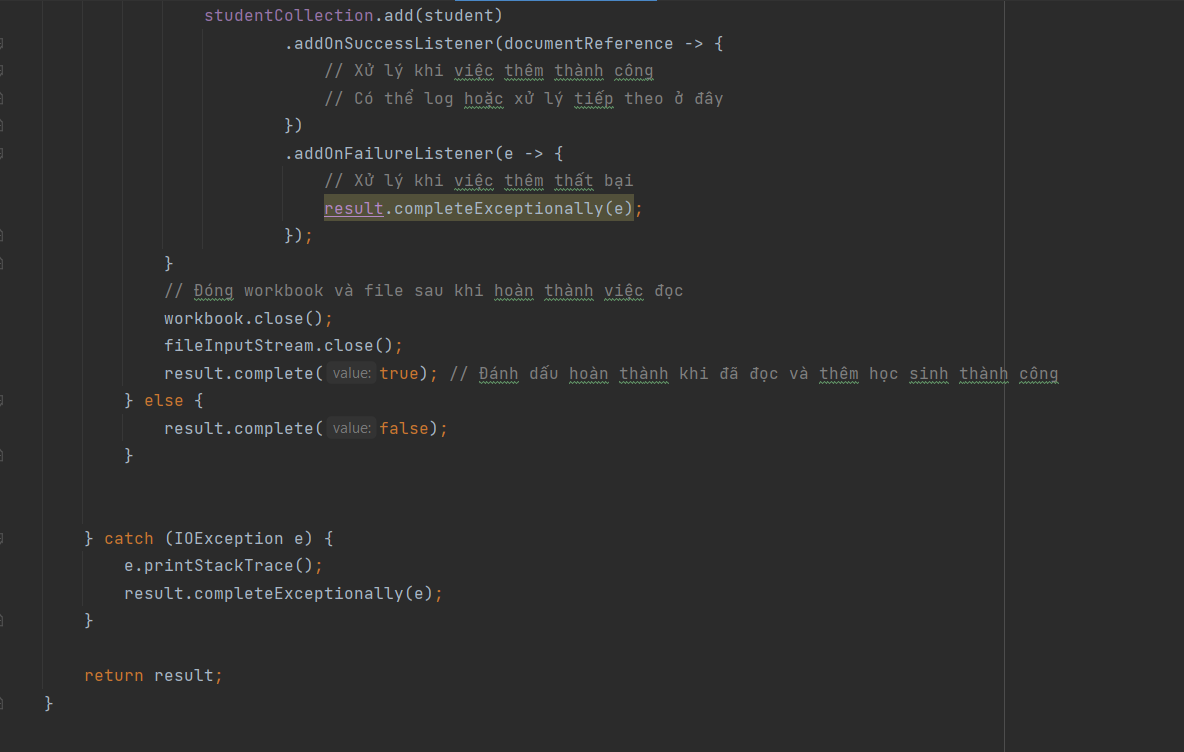


Hình 28 Đoạn code gọi Intent để chọn File (2)

Khi đã chọn File thành công, một callback sẽ được gọi lại, kết quả trả về sẽ nhận một URI, sau đó em lấy URI này để đổi thành một đường dẫn tuyệt đối trỏ đến File vừa chọn, nhiệm vụ cuối cùng là đọc file này và insert từng dòng dữ liệu bên trong nó. Để xử lý nhiệm vụ này, em có hàm sau:



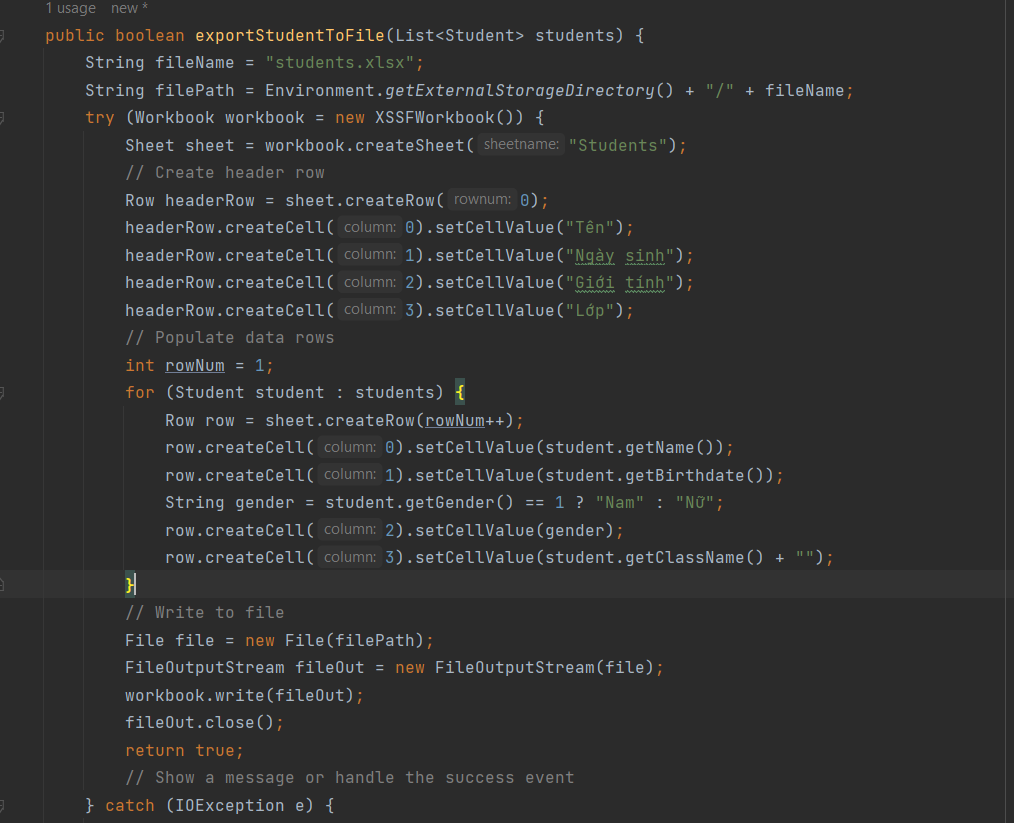
Hình 29 Đoạn code thêm học sinh từ File (1)



Hình 30 Đoạn code thêm học sinh từ File (2)

### Chức năng xuất danh sách học sinh ra File

Chức năng xuất danh sách học sinh ra File không phức tạp như thêm từ File, ở xuất File em chỉ tạo ra một File Excel sau đó với mỗi học sinh trong danh sách học sinh truyền vào từ hàm em sẽ tạo một hàng trong excel.



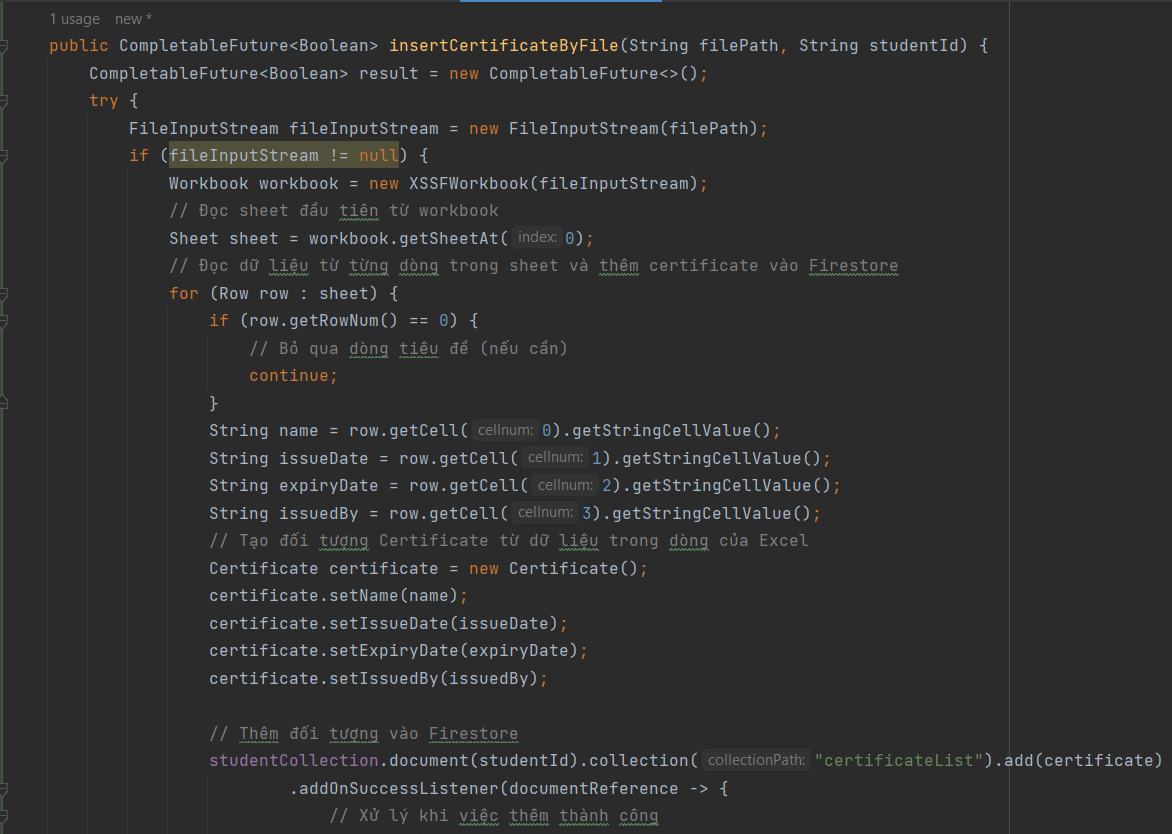
Hình 31 Đoạn code xuất danh sách học sinh ra File (1)



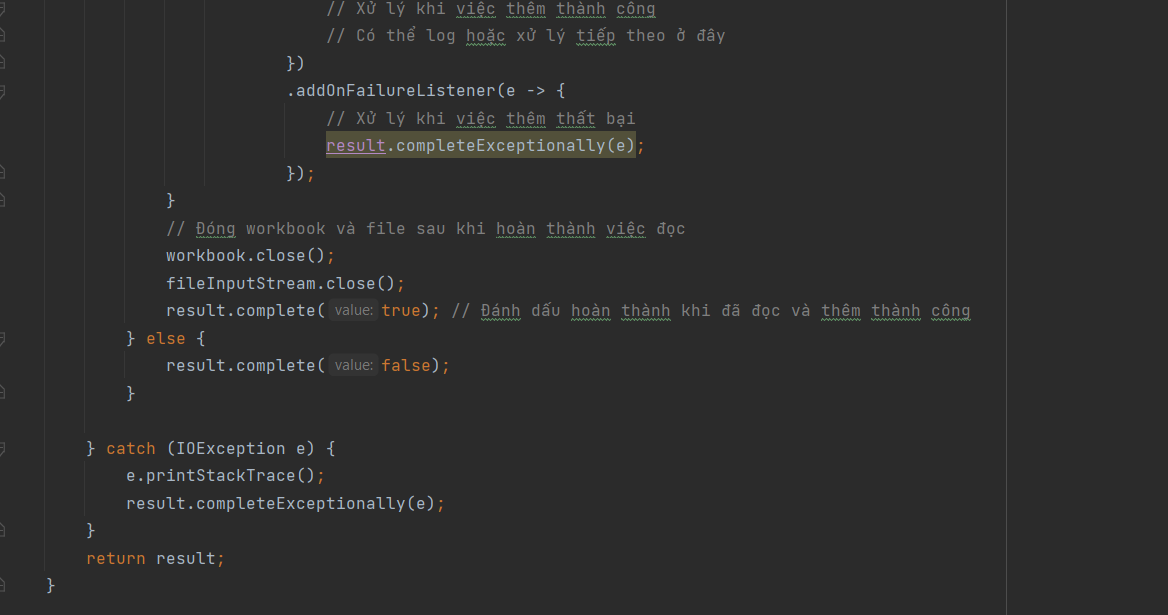
Hình 32 Đoạn code xuất danh sách học sinh ra File (2)

### Chức năng thêm danh sách chứng chỉ từ File

Tương tự với thêm danh sách học sinh, thêm danh sách chứng chỉ cũng tương tự ở phần chọn File Excel, chỉ khác ở phần xử lý insert.



Hình 33 Đoạn code thêm danh sách chứng chỉ từ File (1)

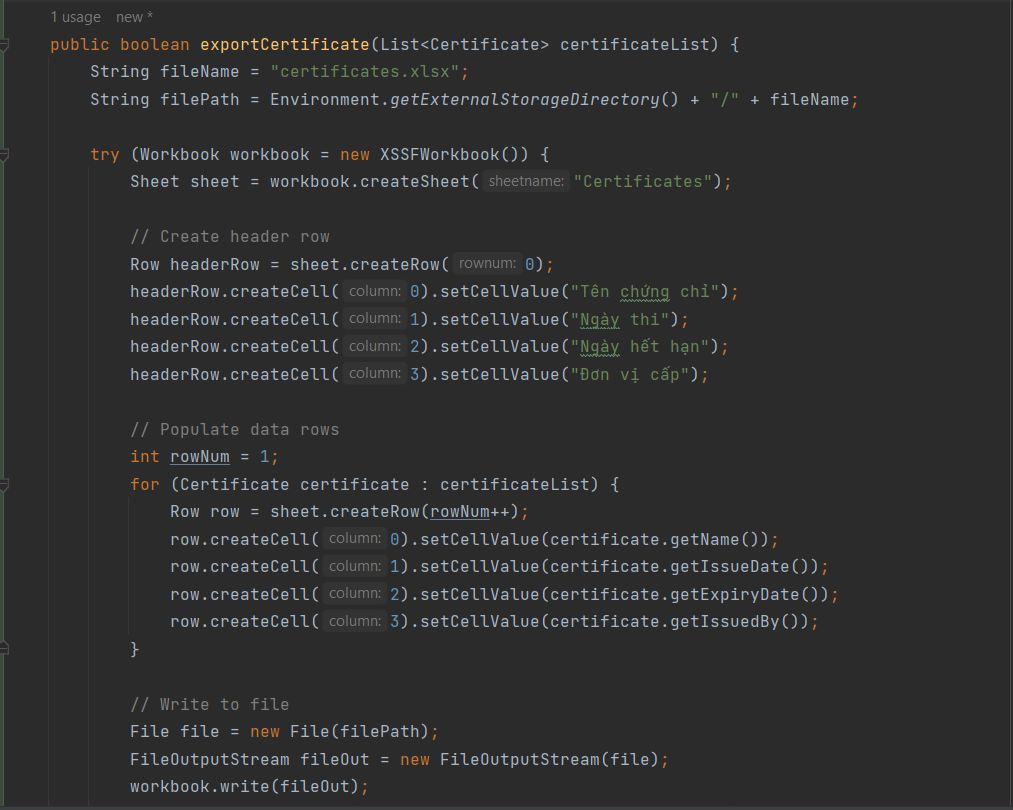


Hình 34 Đoạn code thêm danh sách chứng chỉ từ File (2)

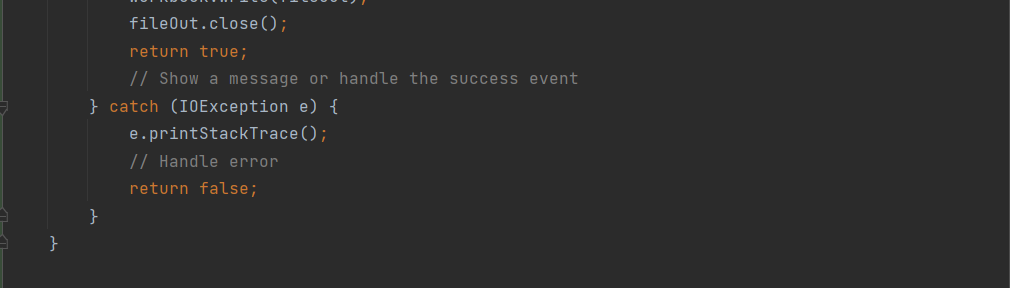
### Chức năng xuất danh sách chứng chỉ ra File

Cũng tương tự như với chức năng xuất danh sách học sinh ra File, chức năng này cũng tương tự.

Đoạn code xử lý như sau:



Hình 35 Đoạn code xử lý xuất danh sách chứng chỉ ra File (1)

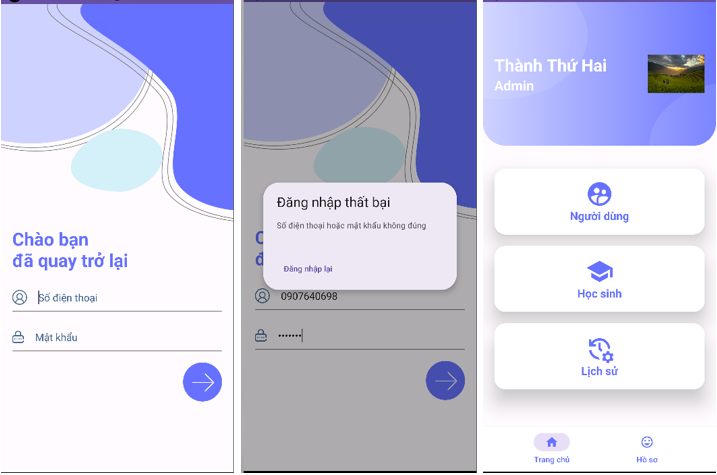


Hình 36 Đoạn code xử lý xuất danh sách chứng chỉ ra File (2)

# DEMO

## Chức năng đăng nhập

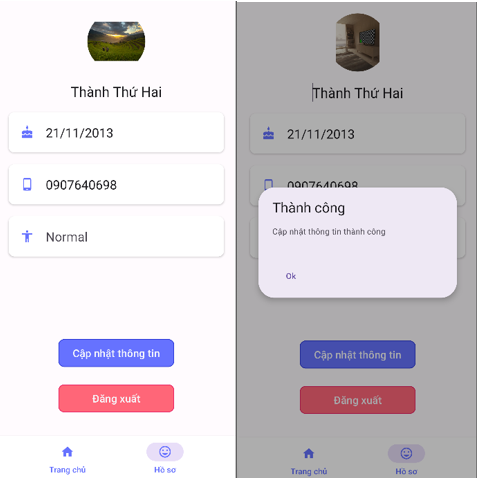
Giao diện đăng nhập, em sẽ tiến hành đăng nhập thất bại và thành công theo thứ tự để kiểm thử.



Hình 37 Demo chức năng đăng nhập

## Chức năng đổi ảnh đại diện và cập nhật thông tin người dùng

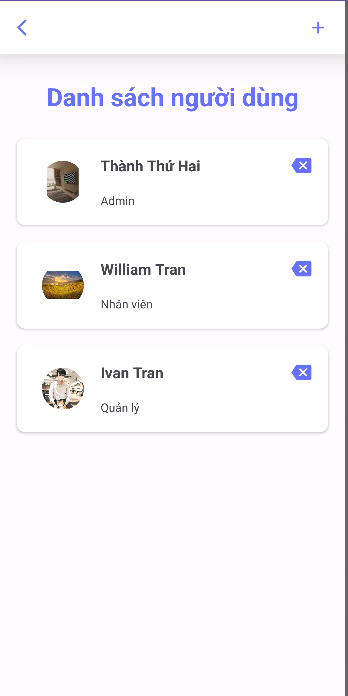
Giao diện trang thông tin cá nhân, em sẽ thực hiện thay đổi ảnh đại diện và tên để kiểm thử:



Hình 38 Demo chức năng đổi ảnh đại diện và cập nhật thông tin

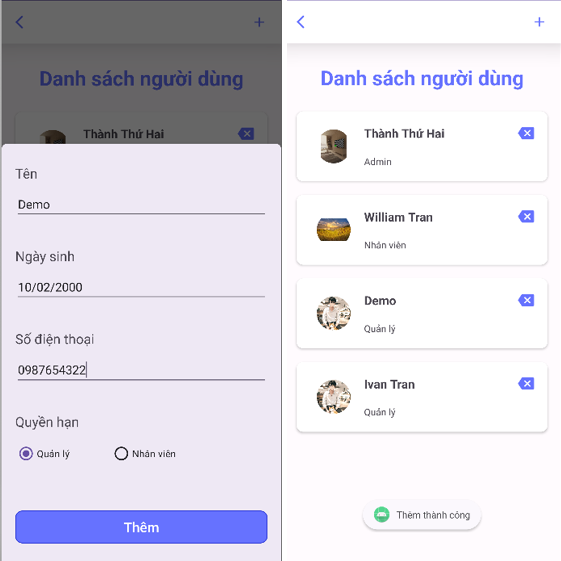
## Chức năng xem danh sách người dùng trên hệ thống

Nếu là Admin thì sẽ truy cập được quyền này, giao diện như sau:



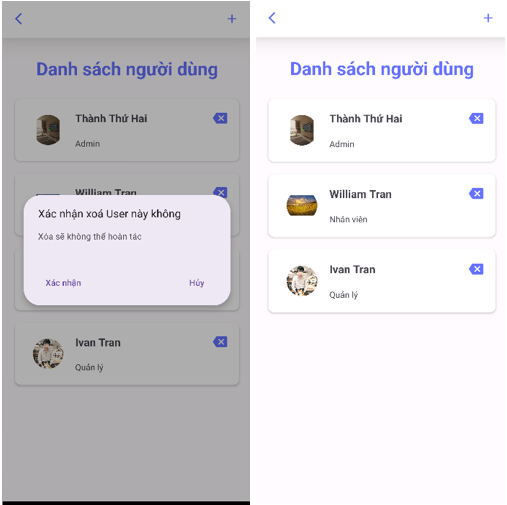
Hình 39 Demo chức năng xem danh sách người dùng

## Chức năng thêm một người dùng mới



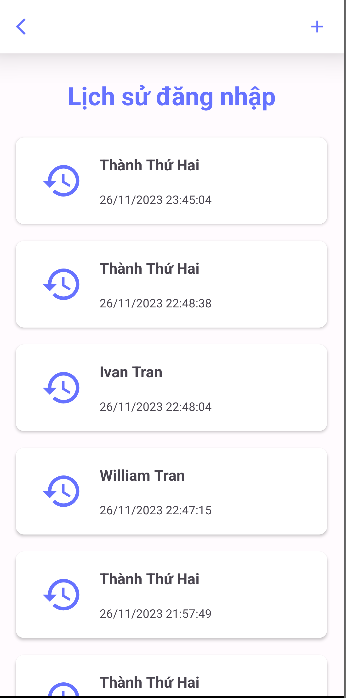
Hình 40 Demo chức năng thêm người dùng mới

## Chức năng xóa người dùng



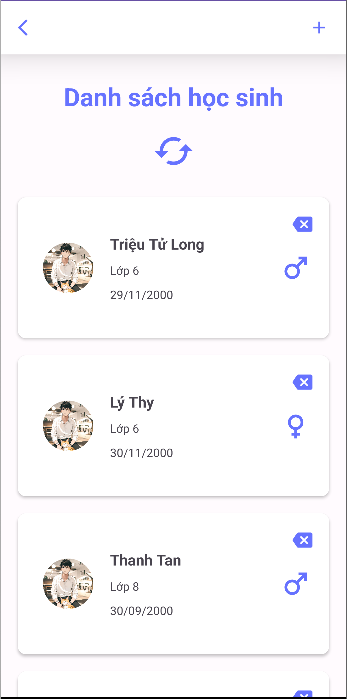
Hình 41 Demo chức năng xóa người dùng

## Chức năng xem lịch sử đăng nhập của người dùng



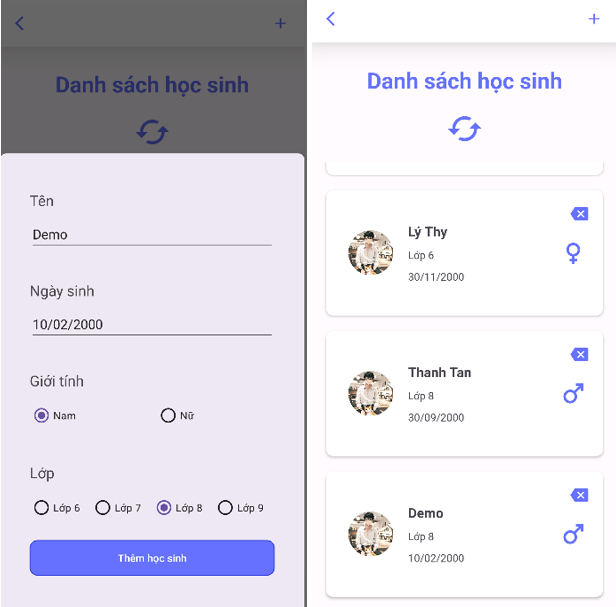
Hình 42 Demo chức năng xem lịch sử đăng nhập

## Demo chức năng xem danh sách học sinh



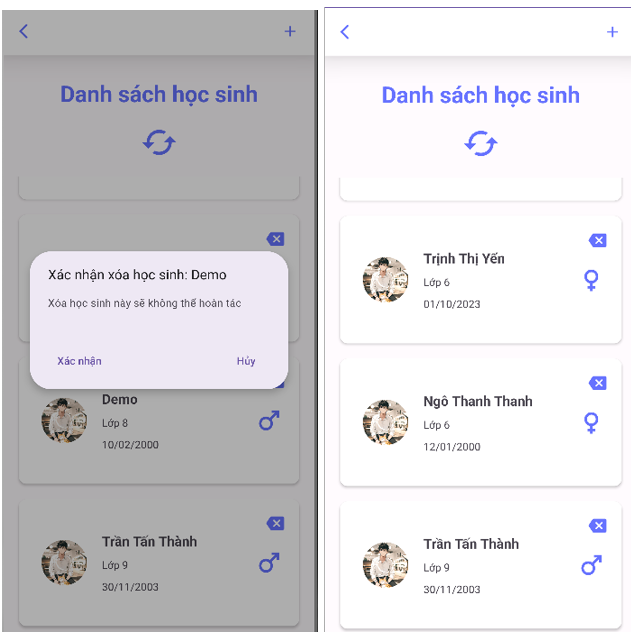
Hình 43 Demo chức năng xem danh sách học sinh

## Demo chức năng thêm một học sinh mới



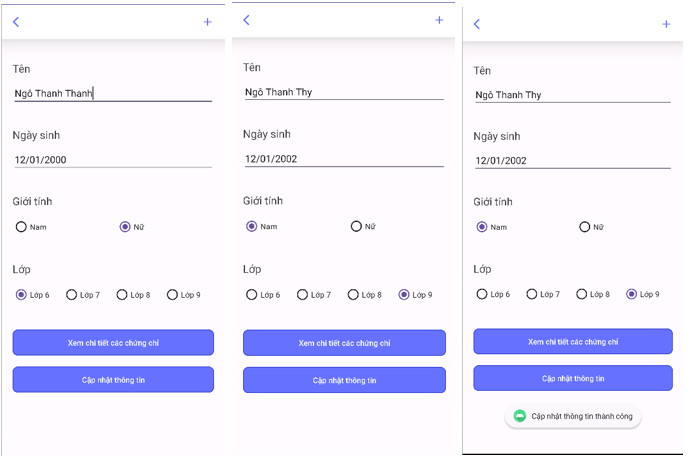
Hình 44 Demo chức năng thêm một học sinh mới

## Demo chức năng xóa một học sinh



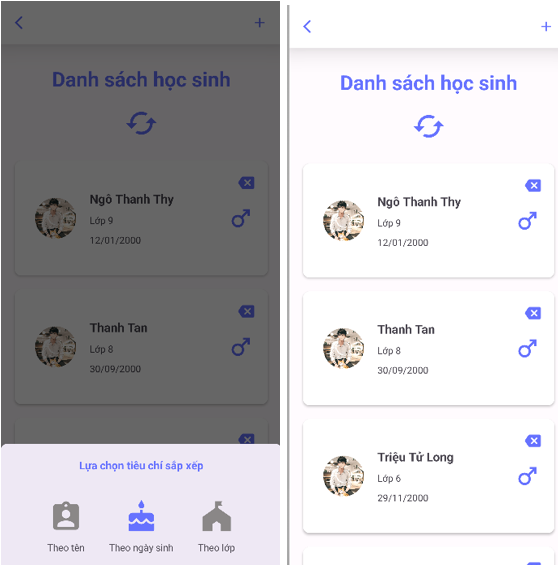
Hình 45 Demo chức năng xóa học sinh

## Demo chức năng cập nhật thông tin học sinh



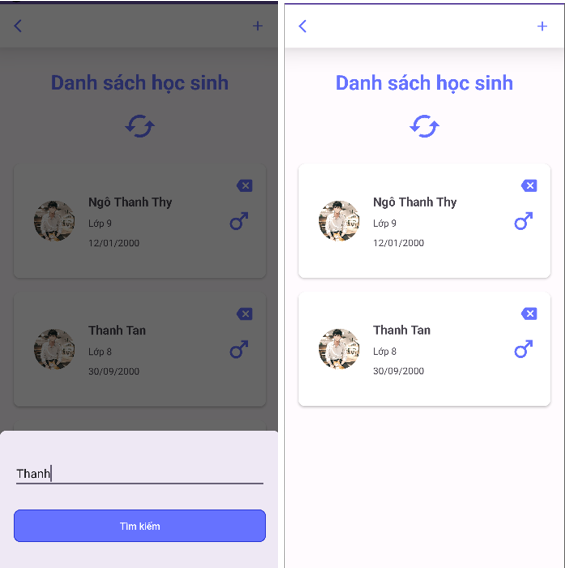
Hình 46 Demo chức năng cập nhật thông tin học sinh

## Demo chức năng sắp xếp danh sách học sinh



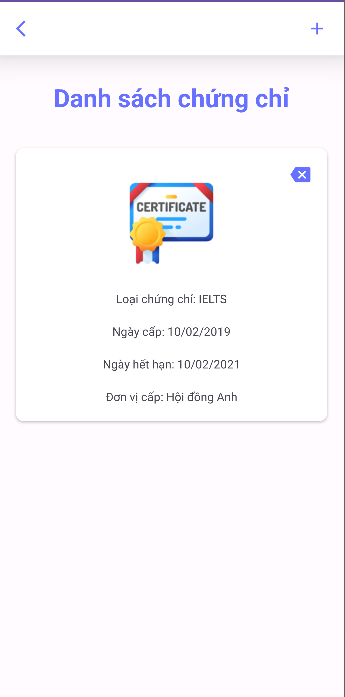
Hình 47 Demo chức năng sắp xếp học sinh

## Demo chức năng tìm kiếm học sinh



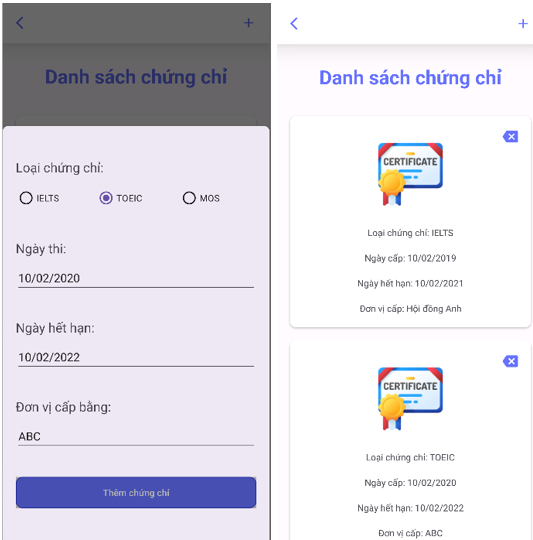
Hình 48 Demo chức năng tìm kiếm học sinh

## Demo chức năng xem danh sách chứng chỉ



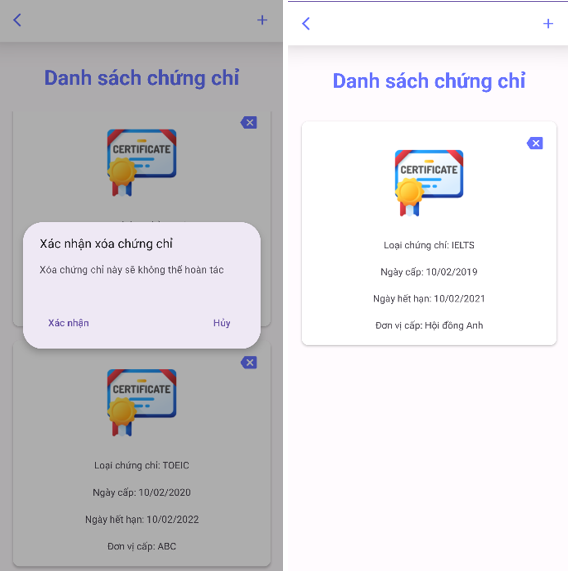
Hình 49 Demo chức năng xem danh sách chứng chỉ

## Demo chức năng thêm chứng chỉ



Hình 50 Demo chức năng thêm chứng chỉ

## Demo chức năng xóa chứng chỉ



Hình 51 Demo chức năng xóa chứng chỉ

# TÀI LIỆU THAM KHẢO

Tiếng Anh

Firebase Firestore (25/11/2023), Firestore Documentation, website: <https://firebase.google.com/docs/firestore>

Firebase Storage (25/11/2023), Firebase Storage Documentation, website: <https://firebase.google.com/docs/storage>