

BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC THỦY LỢI



BÀI TẬP THỰC HÀNH SỐ 4
PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG
NỘI DUNG BỔ SUNG: ỨNG DỤNG VỚI CSDL

STT	Mã sinh viên	Họ và tên	Lớp
1	2251172445	Vũ Quang Nhật	64KTPM2

Hà Nội, năm 2025

BÀI TẬP 1: SHARED PREFERENCE

Mục tiêu:

- Hiểu cách sử dụng Shared Preference để lưu trữ dữ liệu cục bộ trong ứng dụng Android.
- Thực hành lưu trữ và đọc dữ liệu từ Shared Preference.

Yêu cầu:

1. Tạo ứng dụng mới:

- Tạo một dự án Android mới bằng Kotlin.
- Thiết kế giao diện người dùng với hai trường nhập (EditText) cho tên người dùng và mật khẩu, và ba nút bấm: "Lưu", "Xóa", và "Hiển thị".

2. Sử dụng Shared Preference:

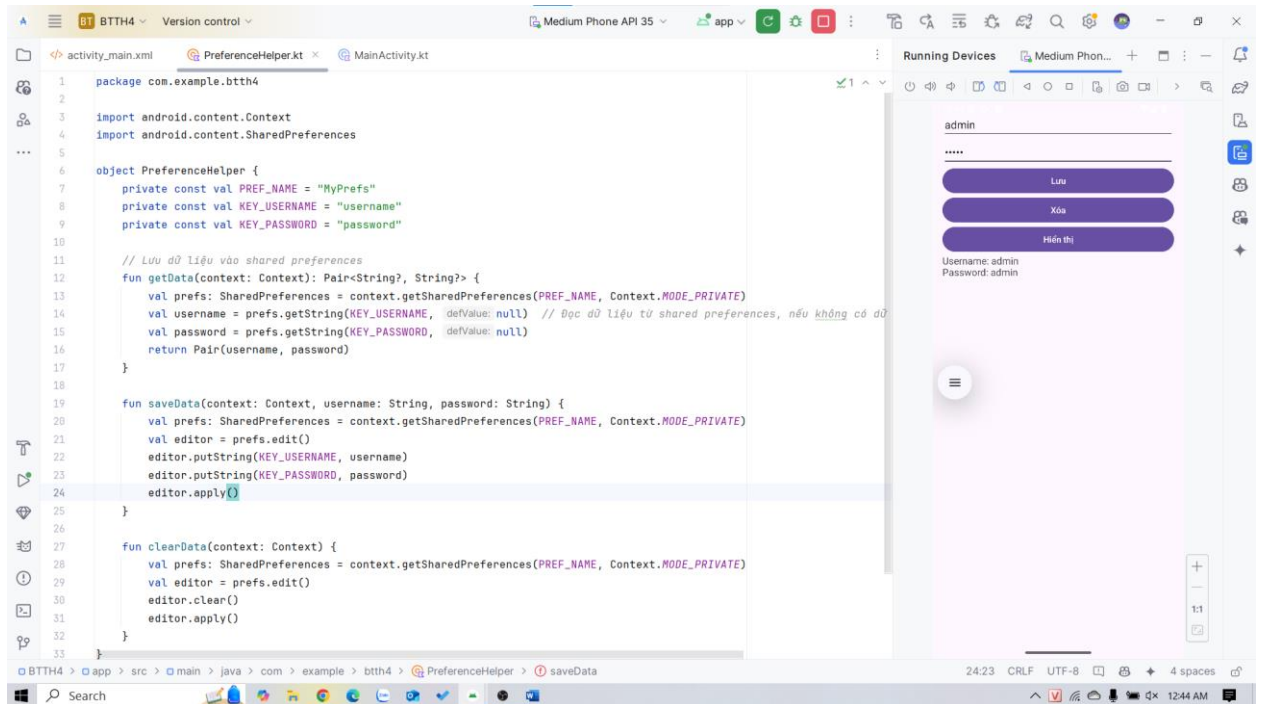
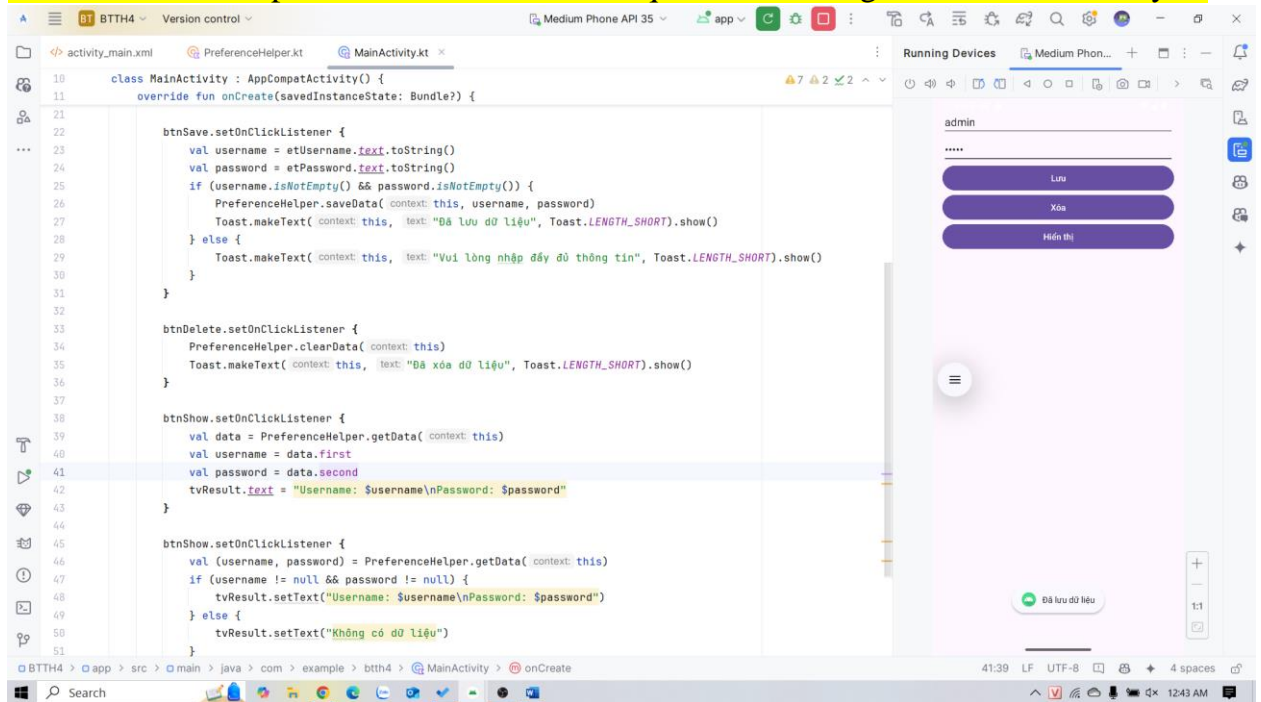
- Tạo một lớp helper **PreferenceHelper** để quản lý Shared Preference.
- Khi người dùng nhấn nút "Lưu", lưu tên người dùng và mật khẩu vào Shared Preference.
- Khi người dùng nhấn nút "Xóa", xóa dữ liệu đã lưu trong Shared Preference.
- Khi người dùng nhấn nút "Hiển thị", đọc dữ liệu từ Shared Preference và hiển thị lên màn hình.

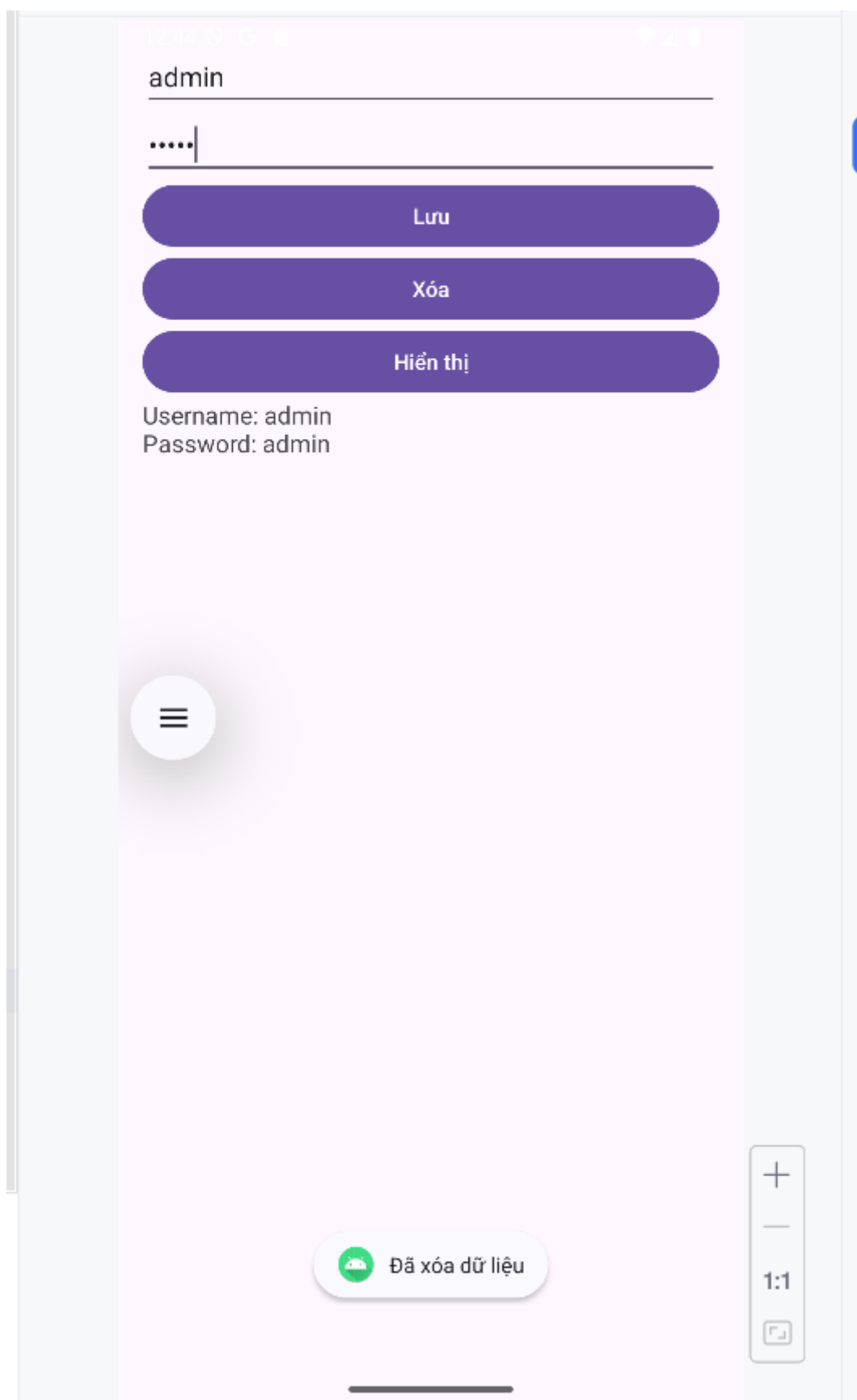
3. Thực hành:

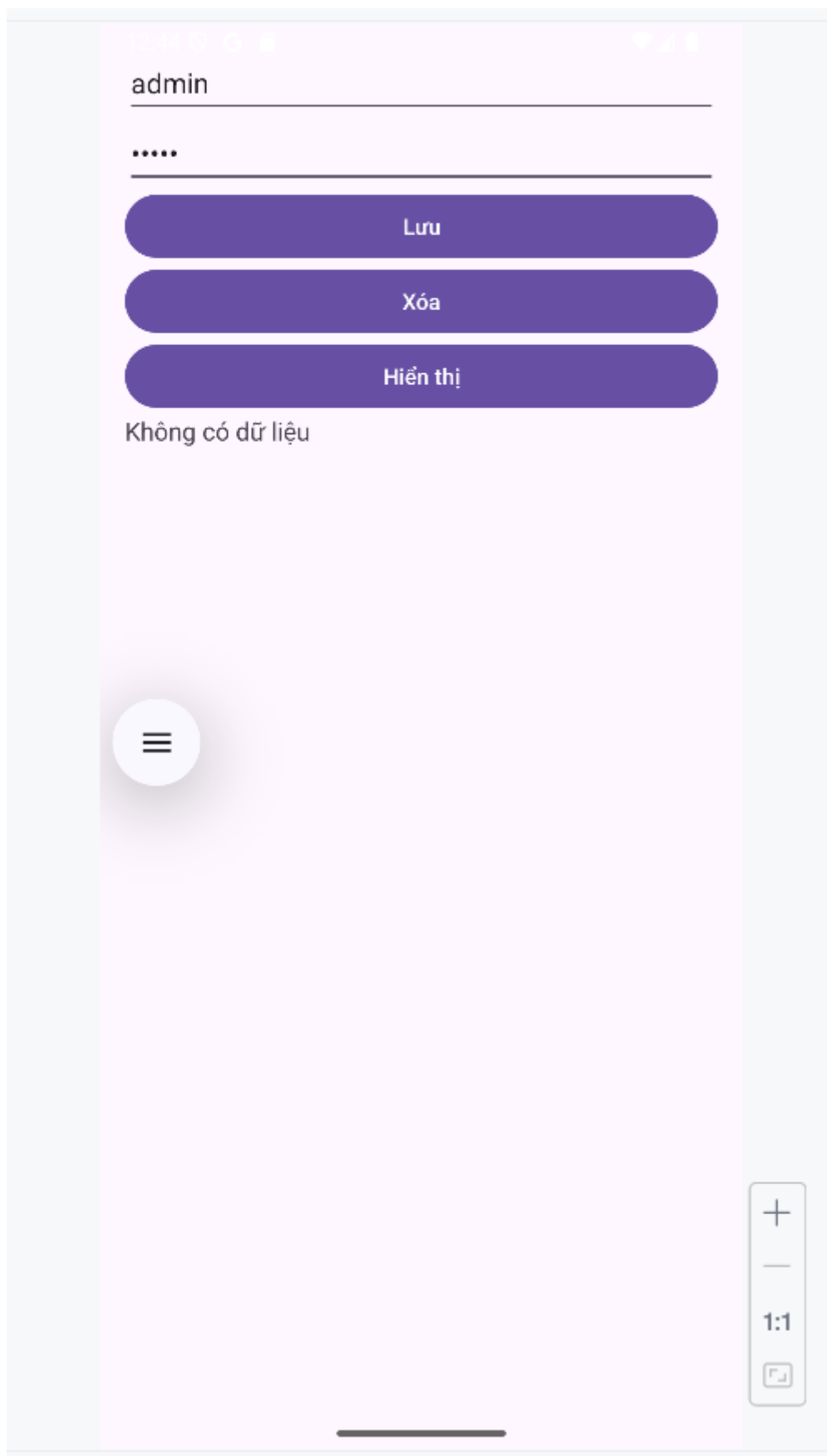
- Viết mã Kotlin để thực hiện các chức năng trên.
- Sử dụng `getSharedPreferences` để truy cập Shared Preference và `edit()` để lưu dữ liệu.
- Sử dụng `commit()` hoặc `apply()` để lưu thay đổi.

4. Kết quả

<<Sinh viên chụp Ảnh màn hình kết quả và mã nguồn chính tại đây>>







BÀI TẬP 2: SQLite

Mục tiêu:

- Hiểu cách sử dụng SQLite để lưu trữ dữ liệu trong ứng dụng Android.
- Thực hành tạo cơ sở dữ liệu SQLite, thêm, sửa, xóa dữ liệu.

Yêu cầu:

1. Tạo ứng dụng mới:

- Tạo một dự án Android mới bằng Kotlin.
- Thiết kế giao diện người dùng với hai trường nhập (EditText) cho tên và số điện thoại, và bốn nút bấm: "Thêm", "Sửa", "Xóa", và "Hiển thị".

2. Sử dụng SQLite:

- Tạo một lớp helper để quản lý cơ sở dữ liệu SQLite.
- Tạo bảng dữ liệu với hai cột: tên và số điện thoại.
- Viết các hàm để thêm, sửa, xóa dữ liệu từ cơ sở dữ liệu.
- Khi người dùng nhấn nút "Hiển thị", đọc dữ liệu từ cơ sở dữ liệu và hiển thị lên màn hình.

3. Thực hành:

- Viết mã Kotlin để thực hiện các chức năng trên.
- Sử dụng SQLiteOpenHelper để tạo và quản lý cơ sở dữ liệu.

4. Kết quả

<<Sinh viên chụp Ảnh màn hình kết quả và mã nguồn chính tại đây>>

nhaajt

p122

Thêm

Sửa

Xóa

Hiện thị

12: nhaajt - 0122

13: nhaajt - 0122

14: nhaajt - 0122

15: nhaajt - 0122

16: nhaajt - 0122

17: nhaajt - 0122

18: nhaajt - 0122

19: nhaajt - 0122

20: nhaajt - 0122

21: nhaajt - 0122

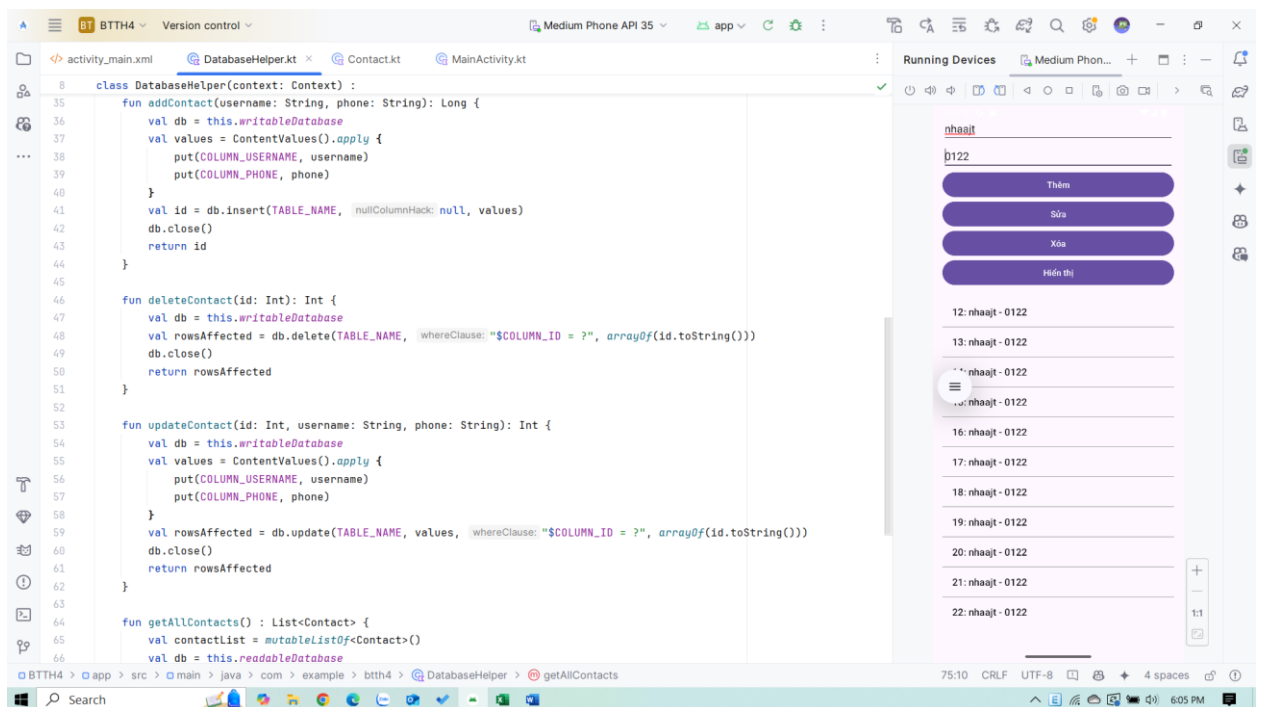
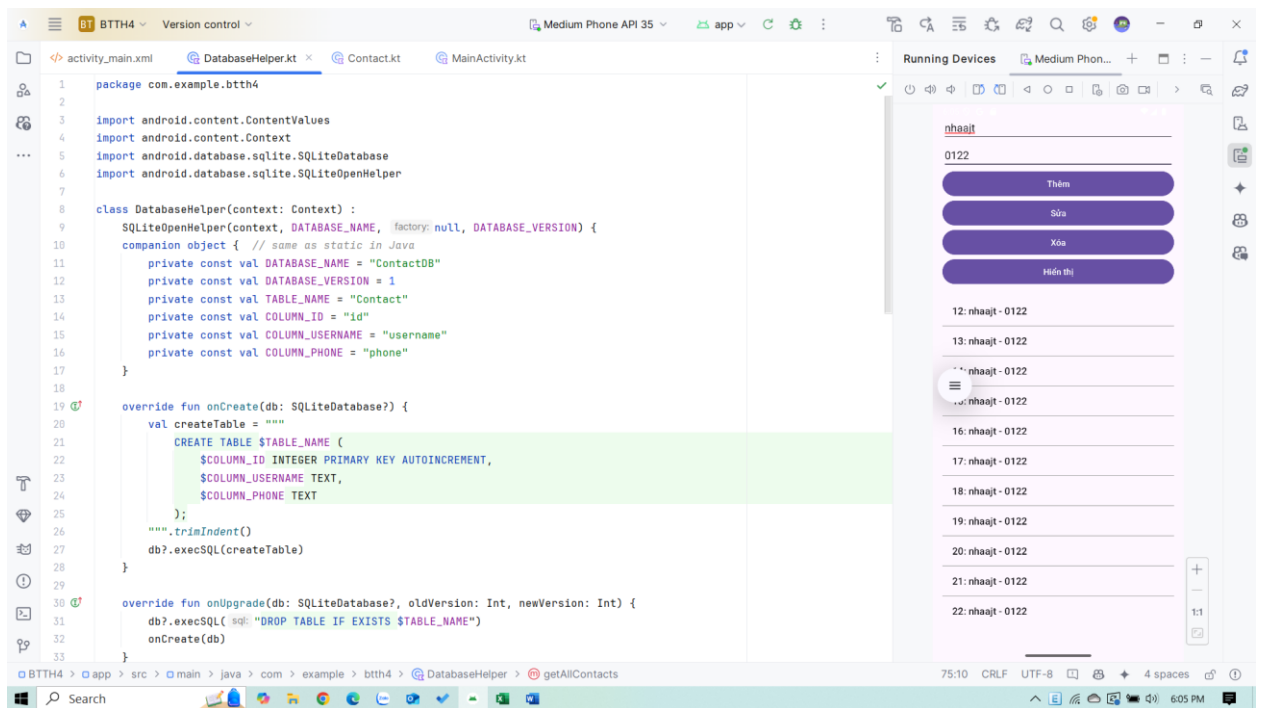
22: nhaajt - 0122

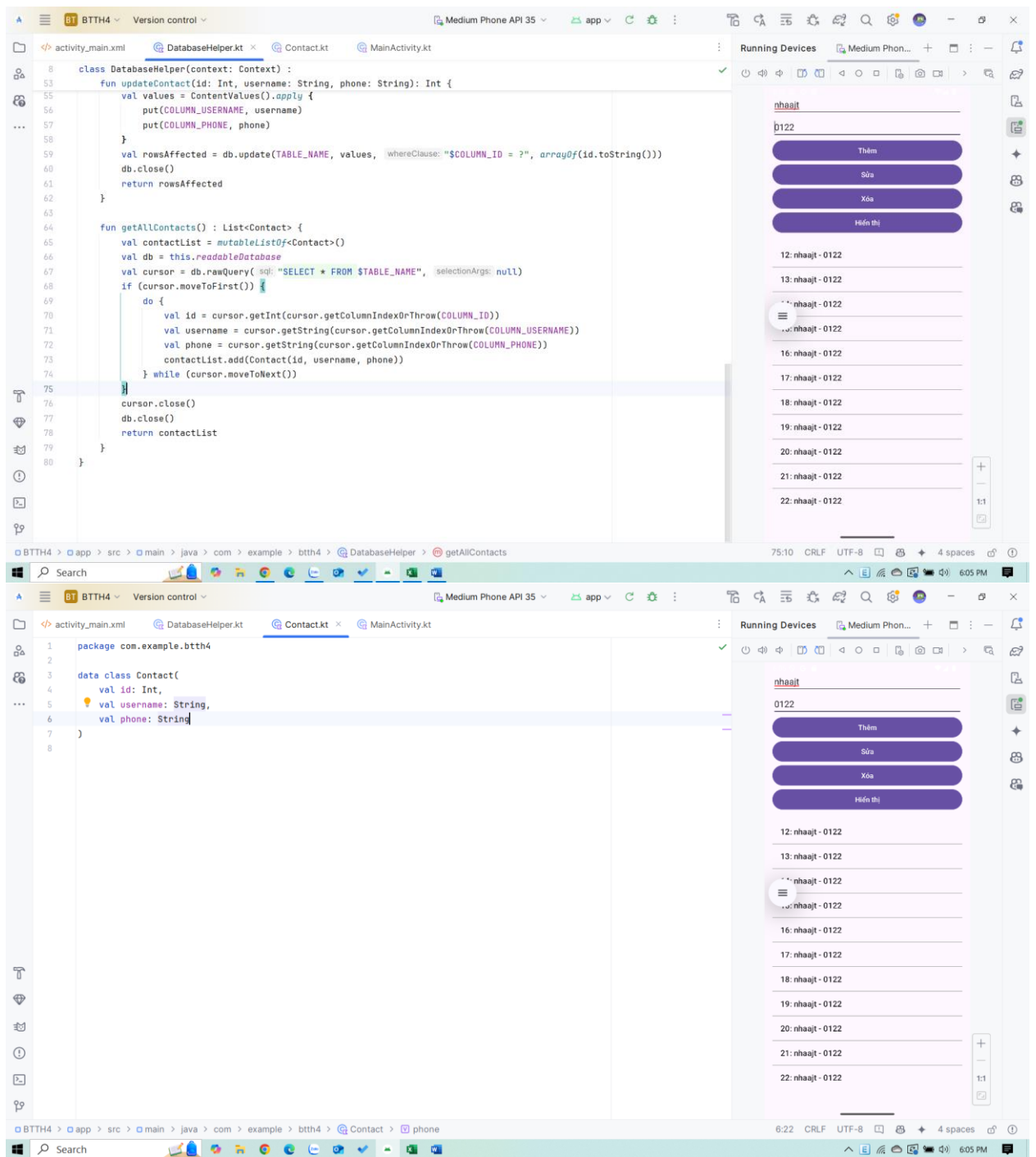
Đã lưu dữ liệu

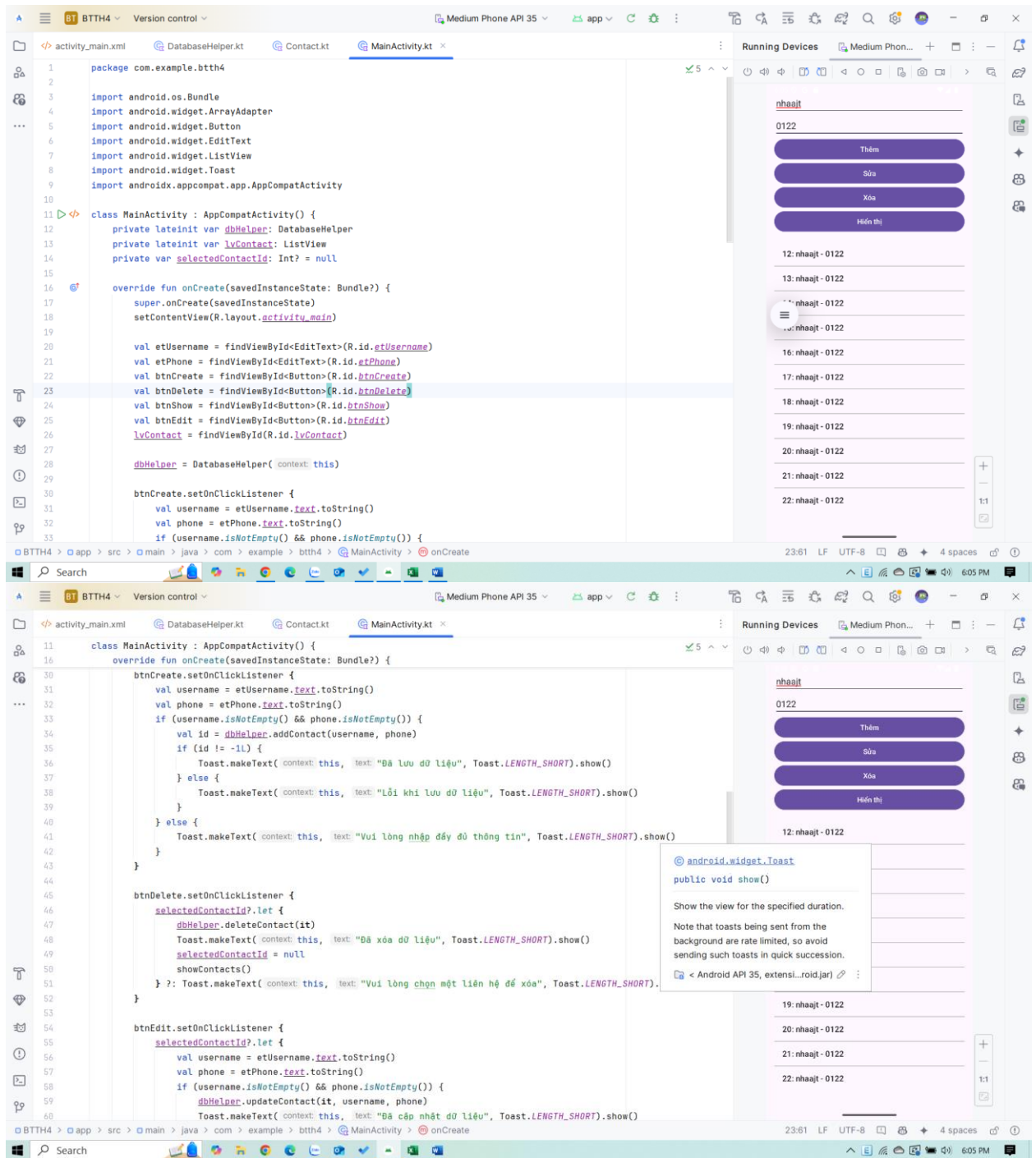
+

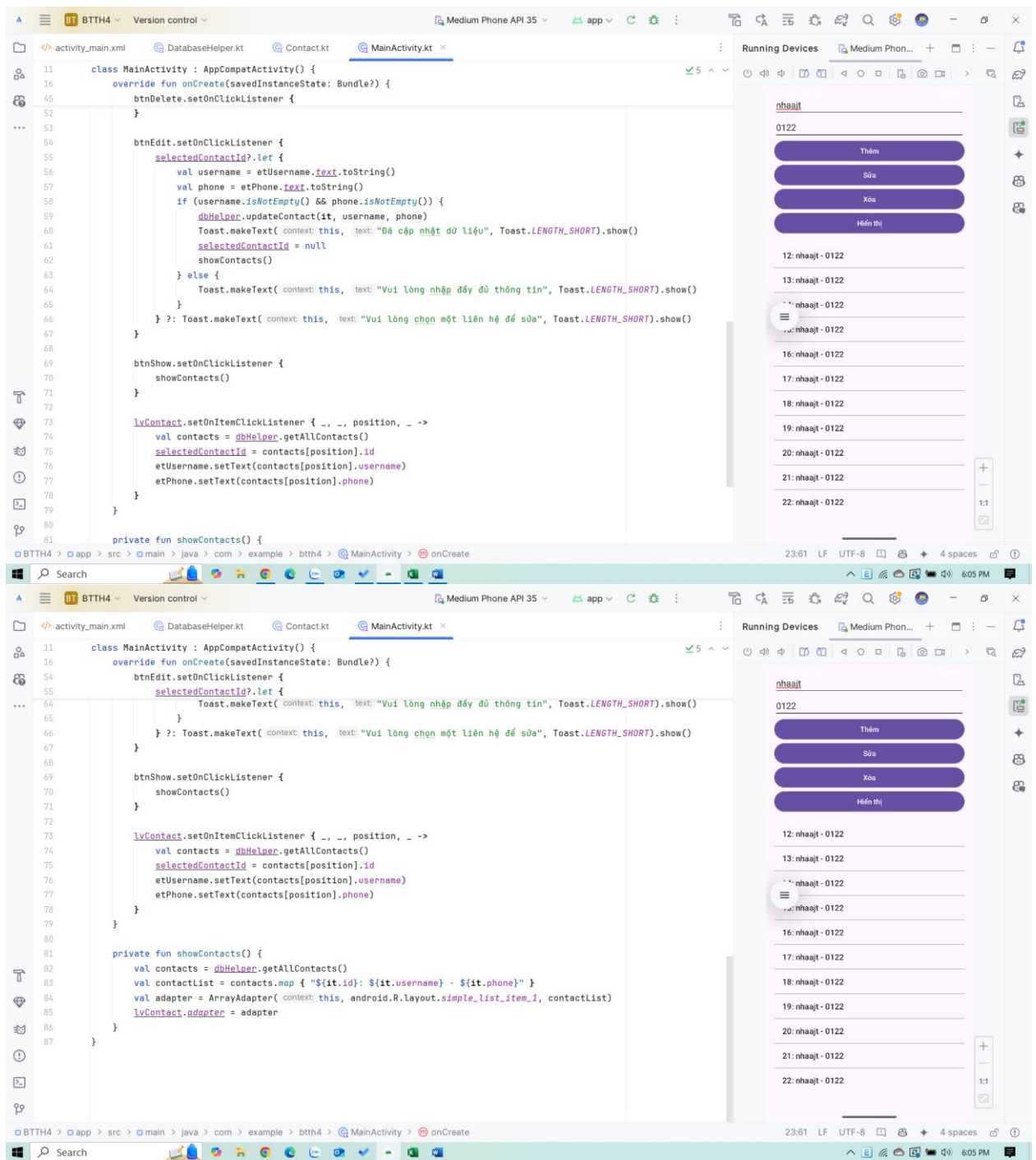
—

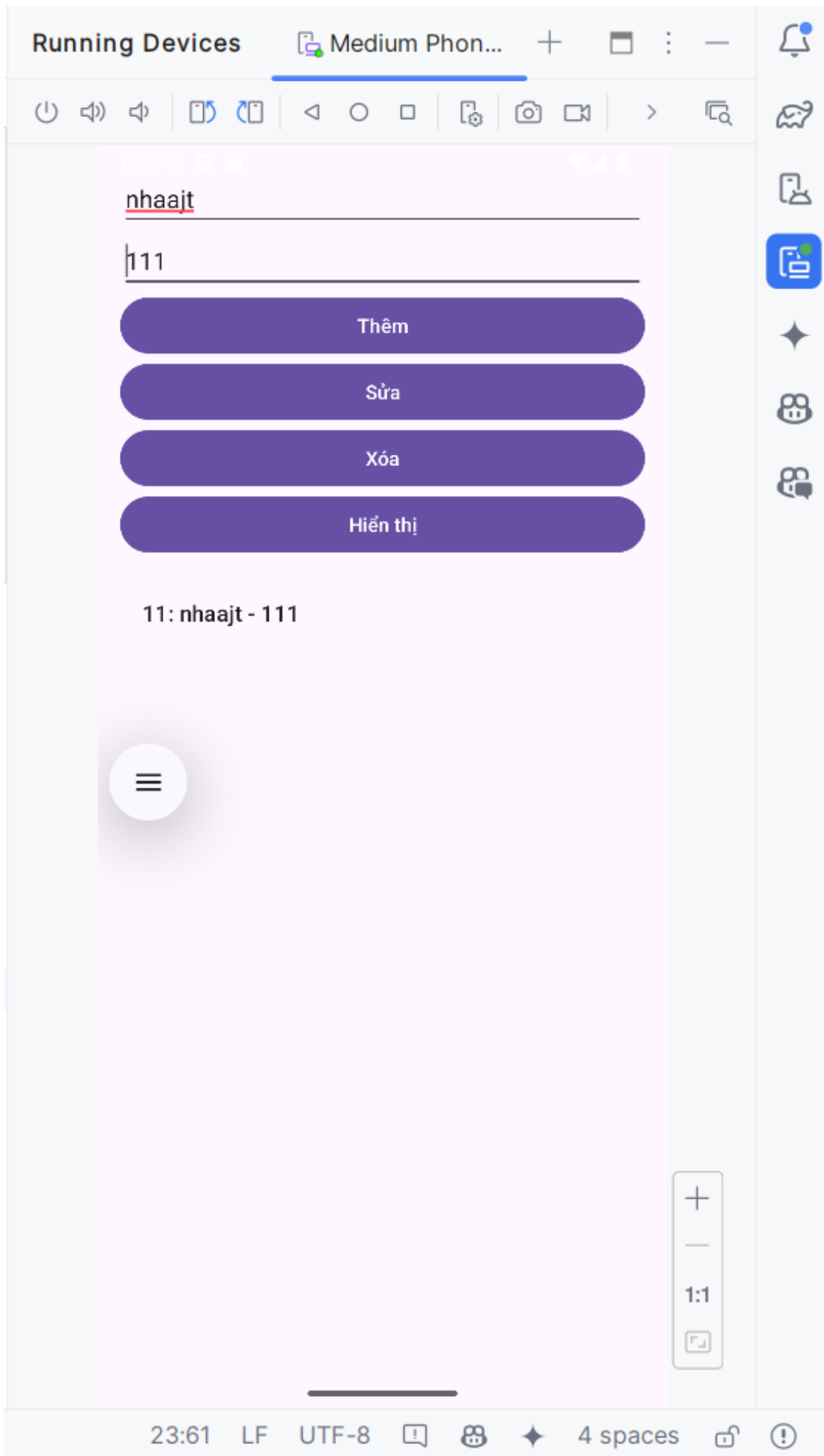
1:1

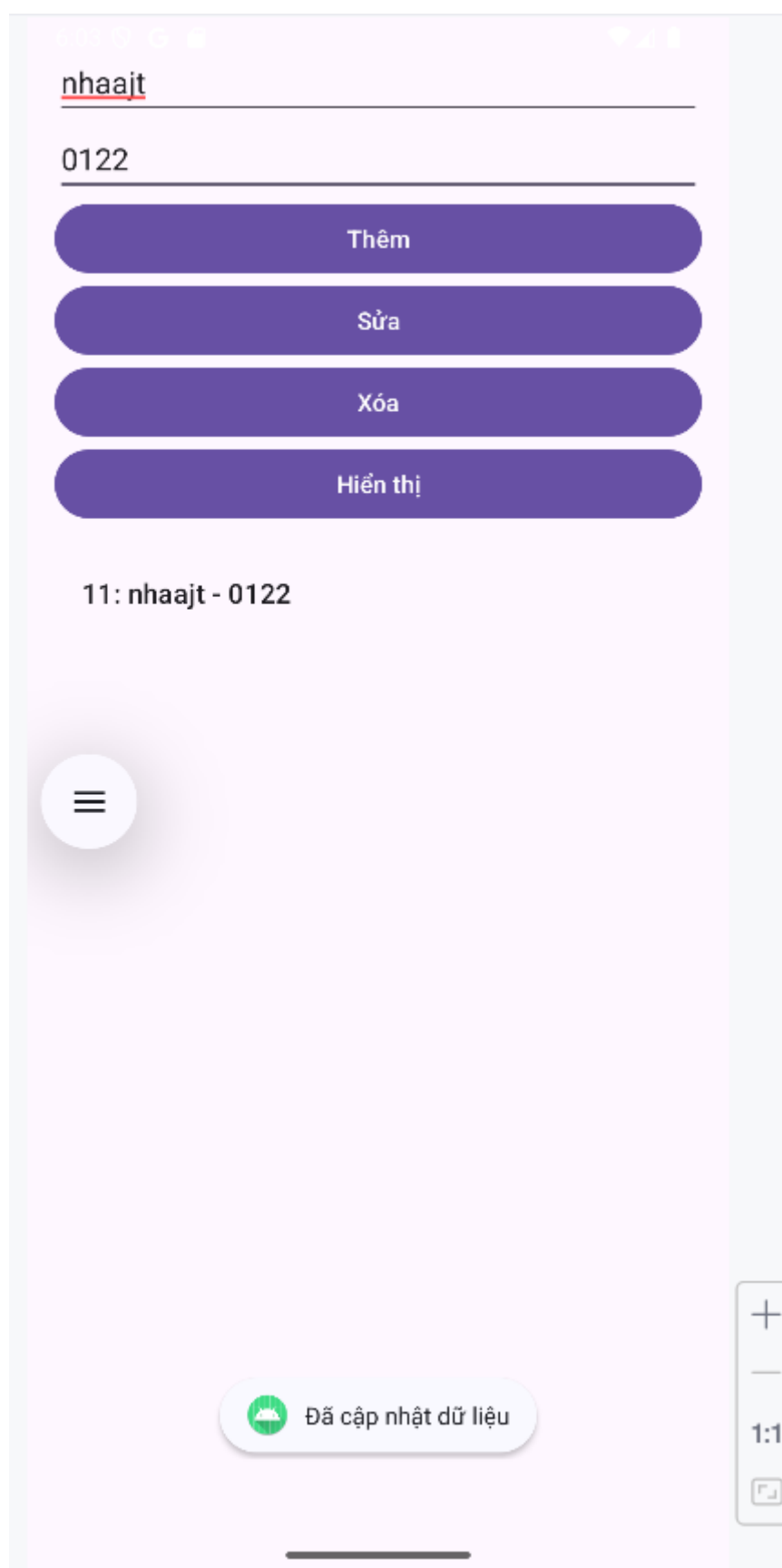


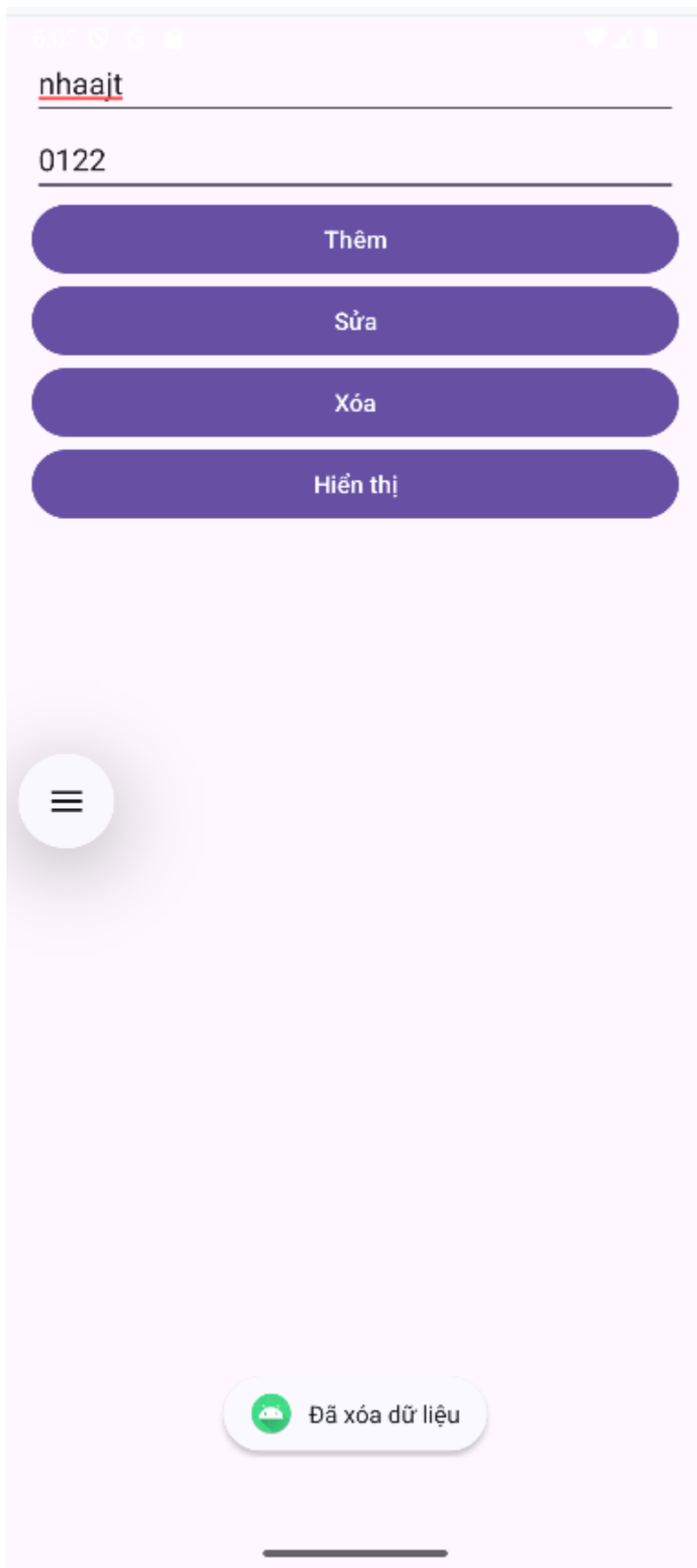












BÀI TẬP 3: HỆ SINH THÁI FIREBASE

Mục tiêu:

- Hiểu rõ về các dịch vụ chính của Firebase.

- Biết cách tích hợp Firebase vào dự án phát triển ứng dụng.

Yêu cầu:

1. Tìm hiểu các dịch vụ chính của Firebase:

- Firebase Authentication: Xác thực người dùng.
- Firebase Realtime Database và Cloud Firestore: Cơ sở dữ liệu thời gian thực và NoSQL.
- Firebase Cloud Functions: Chạy mã backend serverless.
- Firebase Cloud Messaging (FCM): Gửi thông báo đẩy.
- Firebase Storage: Lưu trữ tệp tin trên đám mây.
- Firebase Machine Learning (ML): Tích hợp trí tuệ nhân tạo vào ứng dụng.

2. Viết báo cáo:

- Giới thiệu tổng quan về Firebase và lịch sử phát triển.
- Mô tả chi tiết từng dịch vụ chính của Firebase.
- Thảo luận về lợi ích và ứng dụng của Firebase trong phát triển ứng dụng.

Nội dung báo cáo viết ở đây

1. Giới thiệu tổng quan về Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web được cung cấp bởi Google. Ban đầu, Firebase được ra mắt vào năm 2011 như một dịch vụ cung cấp cơ sở dữ liệu thời gian thực. Sau đó, vào năm 2014, Google mua lại Firebase và tích hợp nó vào hệ sinh thái của mình. Firebase hiện nay cung cấp một loạt dịch vụ hỗ trợ phát triển ứng dụng, bao gồm xác thực người dùng, cơ sở dữ liệu thời gian thực, lưu trữ tệp tin, thông báo đẩy, và tích hợp trí tuệ nhân tạo.

Với Firebase, các nhà phát triển có thể xây dựng các ứng dụng mạnh mẽ mà không cần phải lo lắng về việc thiết lập và quản lý các máy chủ backend. Firebase giúp giảm thiểu thời gian phát triển và tập trung vào trải nghiệm người dùng.

2. Các dịch vụ chính của Firebase

- **2.1. Firebase Authentication:** Firebase Authentication cung cấp các giải pháp xác thực người dùng an toàn và dễ dàng, bao gồm đăng nhập bằng email, số điện thoại, và đăng nhập thông qua các nền tảng xã hội như

Google, Facebook, Twitter, v.v. Dịch vụ này giúp quản lý người dùng hiệu quả, tích hợp chặt chẽ với các dịch vụ khác của Firebase.

- **2.2. Firebase Realtime Database và Cloud Firestore** Firebase: cung cấp hai giải pháp cơ sở dữ liệu chính:
 - Realtime Database: Cơ sở dữ liệu NoSQL thời gian thực, lưu trữ dữ liệu dưới dạng JSON và đồng bộ hóa tức thì tới tất cả các thiết bị kết nối.
 - Cloud Firestore: Cơ sở dữ liệu NoSQL linh hoạt, hỗ trợ truy vấn mạnh mẽ, đồng bộ hóa thời gian thực và khả năng mở rộng quy mô lớn.
- **2.3. Firebase Cloud Functions:** Cloud Functions là dịch vụ chạy mã backend không cần máy chủ (serverless). Các hàm được kích hoạt bởi các sự kiện từ Firebase hoặc các dịch vụ khác, giúp giảm tải công việc xử lý trên máy khách.
- **2.4. Firebase Cloud Messaging (FCM):** FCM cho phép gửi thông báo đẩy tới thiết bị của người dùng. Các thông báo có thể là tin nhắn văn bản, hình ảnh, hoặc các dữ liệu tùy chỉnh. FCM hỗ trợ giao tiếp đa nền tảng (Android, iOS, web).
- **2.5. Firebase Storage:** Firebase Storage dùng để lưu trữ các tệp tin (hình ảnh, video, âm thanh, v.v.) trên đám mây. Dịch vụ này đảm bảo an toàn dữ liệu với các quy tắc bảo mật mạnh mẽ.
- **2.6. Firebase Machine Learning (ML):** Dịch vụ ML của Firebase tích hợp các mô hình học máy vào ứng dụng mà không cần kiến thức chuyên sâu về trí tuệ nhân tạo. Các mô hình có thể được đào tạo trước hoặc tùy chỉnh theo yêu cầu.

3. Lợi ích và ứng dụng của Firebase Firebase mang lại nhiều lợi ích như:

- Phát triển nhanh chóng mà không cần quản lý backend.
- Tăng cường bảo mật và xác thực an toàn.
- Khả năng mở rộng cao, phù hợp với các ứng dụng di động và web.
- Tích hợp nhiều dịch vụ trong một nền tảng duy nhất.

Firebase đã được sử dụng rộng rãi trong các ứng dụng như hệ thống quản lý người dùng, nền tảng học trực tuyến, dịch vụ chia sẻ tệp tin, và các ứng dụng trò chuyện trực tiếp.

KẾT LUẬN

Firebase là một nền tảng mạnh mẽ giúp các nhà phát triển xây dựng ứng dụng di động và web một cách nhanh chóng và hiệu quả. Với các dịch vụ đa dạng và linh hoạt, Firebase đang ngày càng khẳng định vị thế của mình trong hệ sinh thái phát triển phần mềm.

3. Thực hành:

- Tạo một dự án Firebase mới trên Firebase Console.
- Đăng ký ứng dụng Android vào dự án Firebase.
- Sử dụng ít nhất hai dịch vụ của Firebase trong dự án (ví dụ: Authentication và Realtime Database).

Bài tập cụ thể: Tích hợp Firebase Authentication và Realtime Database

Yêu cầu:

1. Tạo ứng dụng mới:

- Tạo một dự án Android mới bằng Kotlin.
- Thiết kế giao diện người dùng với hai trường nhập (EditText) cho email và mật khẩu, và ba nút bấm: "Đăng ký", "Đăng nhập", và "Hiển thị dữ liệu".

2. Tích hợp Firebase Authentication:

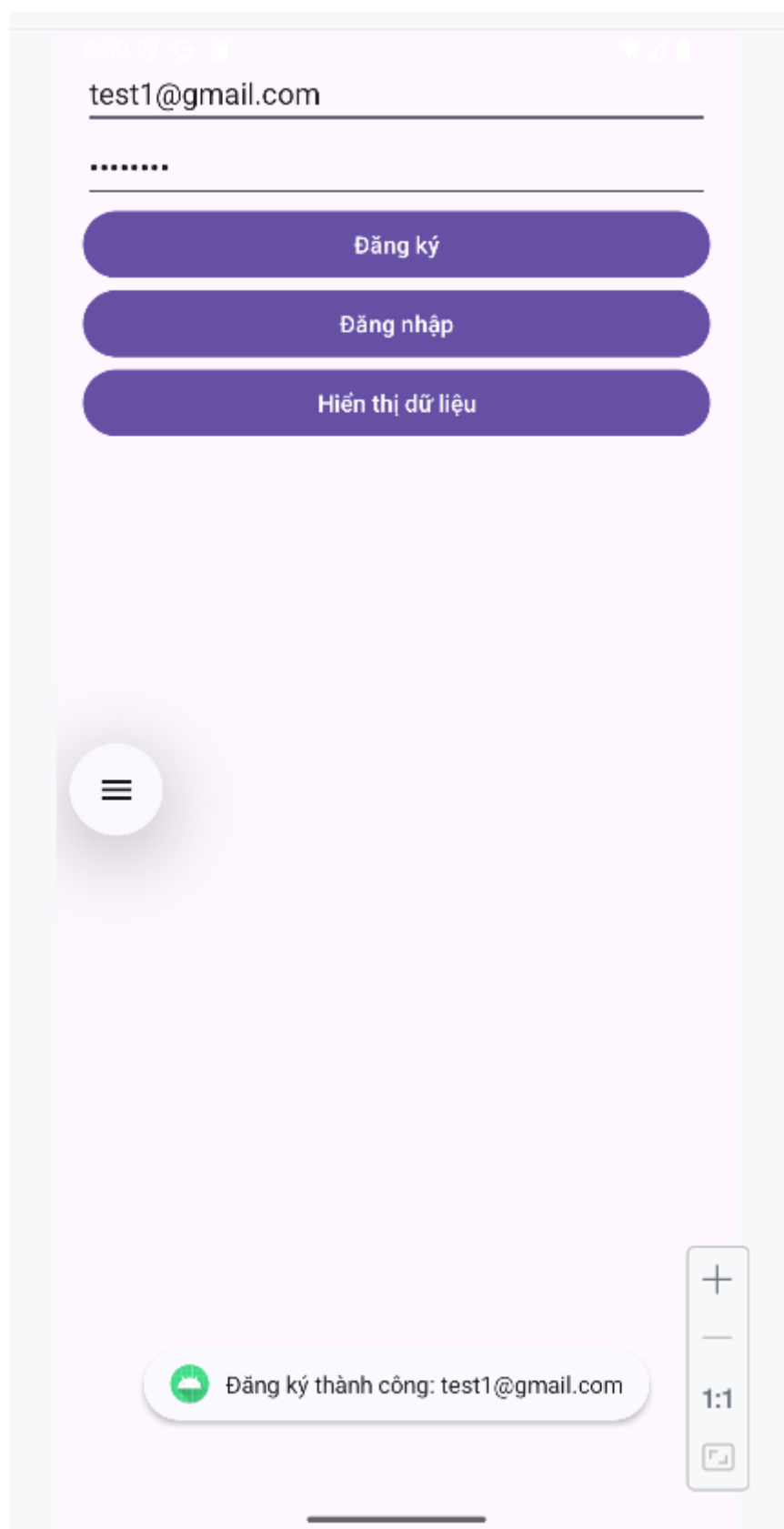
- Sử dụng Firebase Authentication để cho phép người dùng đăng ký và đăng nhập bằng email và mật khẩu.
- Viết mã để xử lý các sự kiện đăng ký và đăng nhập thành công hoặc thất bại.

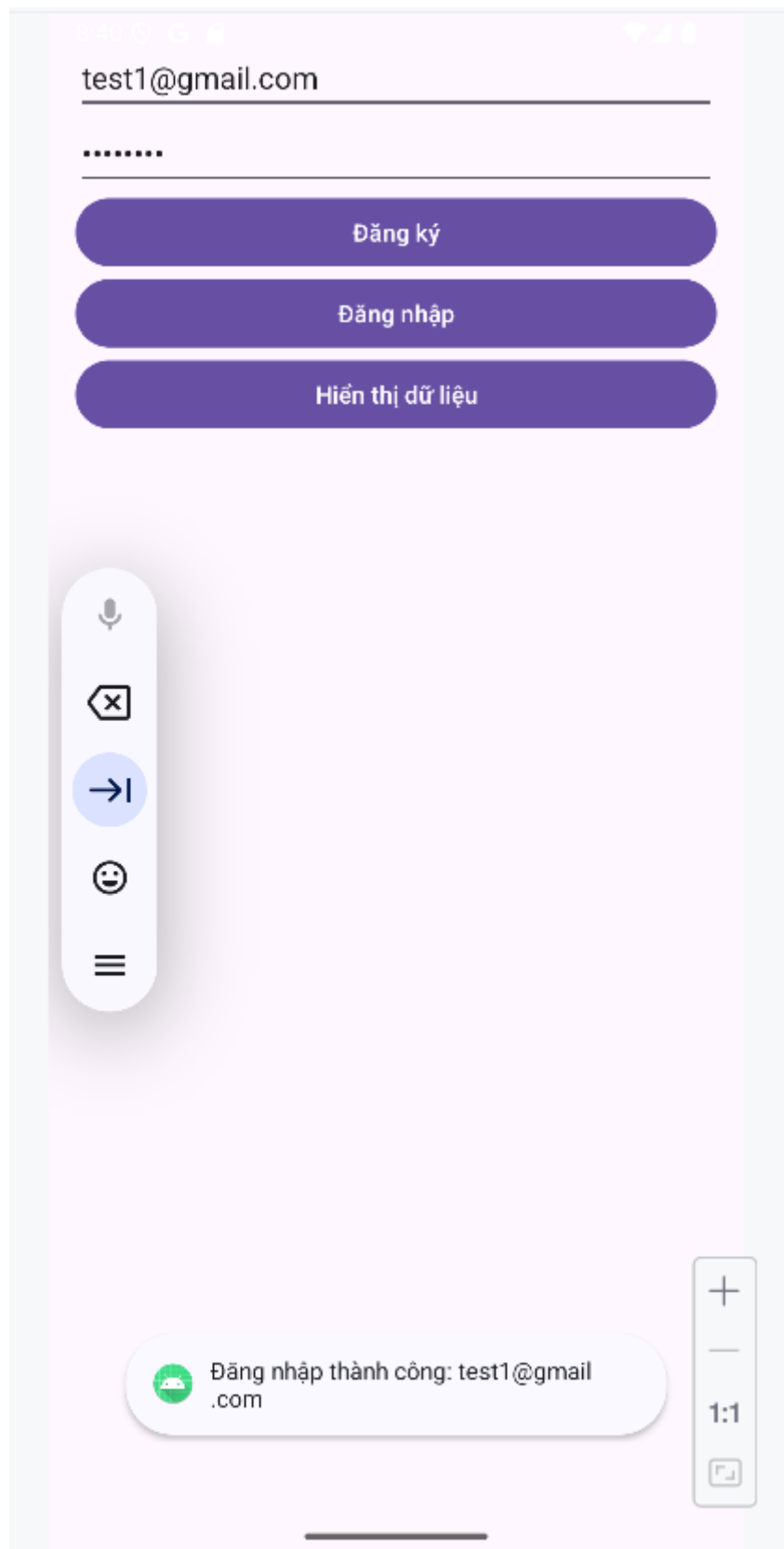
3. Tích hợp Firebase Realtime Database:

- Sau khi người dùng đăng nhập thành công, lưu trữ thông tin người dùng vào Firebase Realtime Database.
- Khi người dùng nhấn nút "Hiển thị dữ liệu", đọc dữ liệu từ Firebase Realtime Database và hiển thị lên màn hình.

4. Kết quả

<<Sinh viên chụp Ảnh màn hình kết quả và mã nguồn chính tại đây>>





test1@gmail.com

.....

Đăng ký

Đăng nhập

Hiển thị dữ liệu

Email: test1@gmail.com
Thời gian đăng ký: 1742780386785

+

—

1:1

