

BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC THỦY LỢI



BÀI TẬP LỚN

PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG

ĐỀ TÀI: ỨNG DỤNG YourClassroom

Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng

Sinh viên thực hiện:

STT	Mã sinh viên	Họ và tên	Lớp
1	2251061744	Lê Văn Đức	64CNTT1
2	2251061745	Nguyễn Trung Đức	64CNTT1
3	2251061735	Trịnh Đức Đại	64CNTT1
4	2251061782	Trần Văn Hòa	64CNTT1

Hà Nội, năm 2025

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC THỦY LỢI**



BÀI TẬP LỚN

**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG
ĐỀ TÀI: ỨNG DỤNG YourClassroom**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	2251061744	Lê Văn Đức	10/07/2004		
2	2251061745	Nguyễn Trung Đức	03/04/2004		
3	2251061735	Trịnh Đức Đại	14/03/2004		
4	2251061782	Trần Văn Hòa	26/12/2004		

CÁN BỘ CHẤM THI

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong bối cảnh giáo dục ngày càng được số hóa và nhu cầu học tập linh hoạt ngày càng tăng, việc ứng dụng công nghệ thông tin vào quản lý lớp học và hỗ trợ giảng dạy trở nên vô cùng cần thiết. Các nền tảng học tập trực tuyến và ứng dụng di động ngày càng được ưa chuộng nhờ tính tiện lợi, khả năng tương tác và hỗ trợ học tập hiệu quả.

Xuất phát từ thực tế đó, nhóm chúng em đã lựa chọn đề tài xây dựng ứng dụng hỗ trợ trong quá trình học tập **YourClassroom** trên nền tảng Android nhằm thiết kế và phát triển một ứng dụng học tập dành cho học sinh, sinh viên và giảng viên. Ứng dụng cho phép người dùng tạo, tham gia lớp học, theo dõi thông báo, quản lý bài tập và điểm số, góp phần nâng cao hiệu quả giảng dạy và học tập.

Dự án sử dụng các công nghệ như ngôn ngữ lập trình Java và Kotlin, hệ cơ sở dữ liệu Firebase Realtime Database, và công cụ thiết kế giao diện Figma để xây dựng một sản phẩm hoàn chỉnh, có tính thực tiễn và khả năng triển khai trong thực tế.

Trong quá trình thực hiện, nhóm đã nỗ lực tìm hiểu, học hỏi và áp dụng kiến thức đã học để hoàn thiện sản phẩm. Tuy nhiên, do thời gian và kiến thức còn hạn chế, chắc chắn vẫn còn tồn tại một số thiếu sót. Rất mong nhận được sự góp ý từ quý thầy cô để nhóm có thể cải thiện và hoàn thiện đề tài tốt hơn trong tương lai.

MỤC LỤC

LỜI NÓI ĐẦU.....	1
MỤC LỤC.....	2
DANH MỤC CÁC TỪ VIẾT TẮT	4
Chương 1. TỔNG QUAN VỀ ĐỀ TÀI	5
1.1. Giới thiệu về đề tài	5
1.2. Mục tiêu của đề tài	5
1.3. Phạm vi của đề tài	5
1.4. Phân chia nhiệm vụ	5
Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ	6
2.1. Kiến trúc hệ thống	6
2.2. Giới thiệu về Công nghệ phát triển	6
Chương 3. XÂY DỰNG ỨNG DỤNG.....	7
3.1. Thiết kế Figma	7
3.1.1. Thiết kế giao diện đăng nhập.....	7
3.1.2. Thiết kế giao diện đăng ký	8
3.1.3. Thiết kế giao diện Quên mật khẩu.....	9
3.1.3. Thiết kế giao diện danh sách lớp	10
3.1.4. Thiết kế giao diện Tham gia/Tạo lớp học	11
3.1.5. Thiết kế giao diện Bảng tin.....	13
3.1.6. Thiết kế giao diện Đăng tin	14
3.1.7. Thiết kế giao diện Xóa tin.....	14
3.1.8. Thiết kế giao diện Tạo bài tập.....	15
3.1.9. Thiết kế giao diện Nộp bài/Chấm điểm	16
3.2. Thiết kế CSDL.....	17
3.3. Giao diện ứng dụng	18
3.3.1. Màn hình Đăng nhập	18
3.3.2. Màn hình Đăng ký	19
3.3.3. Màn hình Quên mật khẩu	20
3.3.4. Màn hình Danh sách lớp	21
3.3.5. Màn hình Tạo lớp	22
3.3.6. Màn hình Tham gia lớp học	23
3.3.7. Màn hình Xóa lớp học.....	24
3.3.8. Màn hình Bảng tin	25
3.3.9. Màn hình Đăng tin (Teacher)	25
3.3.10. Màn hình Xóa bài đăng (Teacher).....	26

3.3.11. Màn hình Bài tập	27
3.3.12. Màn hình Tạo bài tập (Teacher)	27
3.3.13. Màn hình Chấm điểm (Teacher)	28
3.3.14. Màn hình Nộp bài (Student)	29
3.4. Code minh họa các chức năng cốt lõi	30
KẾT LUẬN	41
1. Kết quả đạt được	41
2. Nhược điểm	41
3. Hướng phát triển.....	41
TÀI LIỆU THAM KHẢO	43

DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	MVC	Model – View – Controller
2	SDK	Software Development Kit
3	UI	User Interface
4	MVVM	Model – View – View Model

Chương 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu về đề tài

Ứng dụng YourClassroom là một nền tảng học tập được phát triển trên hệ điều hành Android. Mục tiêu chính của ứng dụng này là cung cấp một môi trường học tập tương tác, nơi giáo viên và học sinh có thể kết nối và thực hiện các hoạt động học tập như giao bài tập, tải tài liệu, và quản lý tiến độ học tập. Ứng dụng sử dụng các công nghệ hiện đại như Java và Kotlin để đảm bảo hiệu suất và tính linh hoạt.

1.2. Mục tiêu của đề tài

1. **Tạo môi trường học tập số hiện đại:** Cung cấp một nền tảng mà giáo viên và học sinh có thể dễ dàng thực hiện các hoạt động học tập trực tuyến.
2. **Tăng cường khả năng tương tác:** Cho phép giáo viên giao bài tập, thu thập và đọc tệp đính kèm, và theo dõi tiến độ của học sinh.
3. **Bảo mật và hiệu quả:** Ứng dụng sử dụng cơ chế bảo mật, đảm bảo sự riêng tư của người dùng.

1.3. Phạm vi của đề tài

1. **Ứng dụng Android:** Được phát triển dành riêng cho các thiết bị Android với phiên bản tối thiểu là Android SDK 27.
2. **Chức năng chính:**
 - **Quản lý tài khoản:** Đăng ký, đăng nhập, quên mật khẩu.
 - **Quản lý lớp học:** Tạo lớp học, tham gia lớp học, xóa lớp học.
 - **Quản lý bảng tin:** Tạo thông báo cho lớp, thông báo bài tập.
 - **Quản lý bài tập:** Thêm bài tập, chấm điểm, nộp bài làm.
3. **Công nghệ sử dụng:**
 - **Ngôn ngữ lập trình:** Java và Kotlin.
 - **Firebase:** Lưu trữ và quản lý dữ liệu.
 - **Thư viện AndroidX:** Cho giao diện và quản lý trạng thái ứng dụng.

1.4. Phân chia nhiệm vụ

Họ và Tên	Nhiệm vụ
Lê Văn Đức	Quản lý bài tập
Nguyễn Trung Đức	Quản lý bảng tin
Trần Văn Hòa	Quản lý lớp học

Trình Đức Đại	Quản lý tài khoản
---------------	-------------------

Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ

2.1. Kiến trúc hệ thống

Ứng dụng được phát triển theo mô hình kiến trúc MVC (Model - View - Controller), giúp phân tách rõ ràng giữa logic xử lý các thành phần, giao diện người dùng và dữ liệu. Cụ thể:

- Model, Repository:

Chịu trách nhiệm quản lý dữ liệu và tương tác với Cơ sở dữ liệu thời gian thực Firebase Realtime Database.

- View:

Giao diện người dùng của ứng dụng được xây dựng bằng cách kết hợp giữa Jetpack Compose và XML với các Activity và Fragment trong Android. View chỉ đảm nhận vai trò hiển thị dữ liệu và xử lý các sự kiện tương tác từ người dùng

- Controller:

Là bộ xử lý luồng Activity/Fragment của lớp. Nhận dữ liệu từ model, xử lý logic, sau đó cập nhật lại cho view. Đồng thời nhận được thao tác của người dùng và phản hồi phù hợp.

2.2. Giới thiệu về Công nghệ phát triển

Ứng dụng được phát triển trên nền tảng Android, sử dụng các công cụ sau:

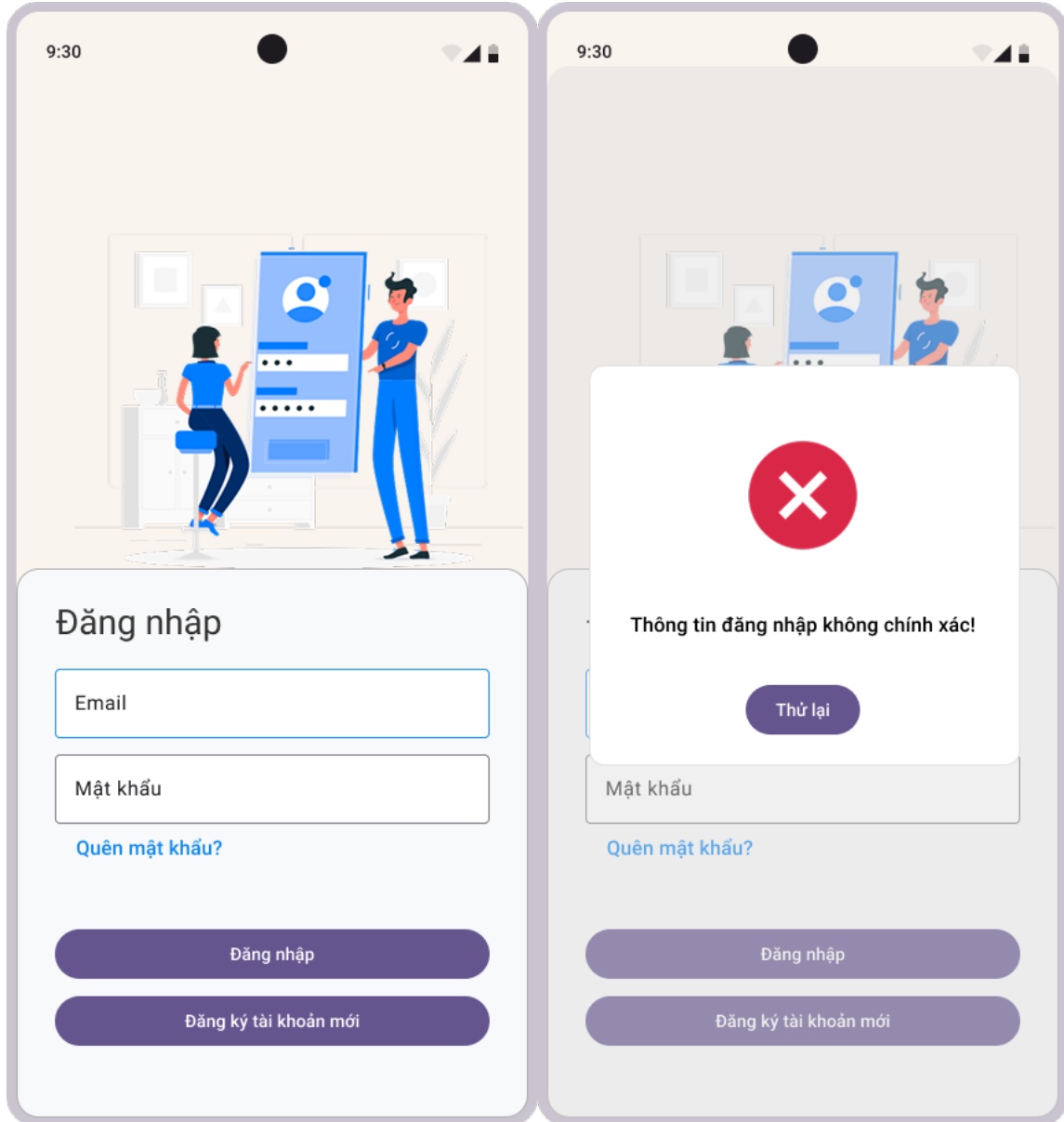
- Ngôn ngữ lập trình: Java và Kotlin.
- Cơ sở dữ liệu: Firebase Realtime Database.
- Firebase Authentication: xác thực người dùng bằng email/password.
- Android SDK: Bộ công cụ phát triển phần mềm được cung cấp bởi Google.
- Giao diện người dùng (UI): Thiết kế bằng Material Design kết hợp XML, Jetpack Compose.
- Glide/Picasso: tải và hiển thị hình ảnh.
- RecyclerView: hiển thị danh sách dữ liệu động.

Chương 3. XÂY DỰNG ỨNG DỤNG

3.1. Thiết kế Figma

Link Github: https://github.com/Kuku107/64CNTT1_Mobile

3.1.1. Thiết kế giao diện đăng nhập



3.1.2. Thiết kế giao diện đăng ký

The image displays two mobile app screens for a registration process. Both screens have a status bar at the top showing the time 9:30 and signal/battery icons.

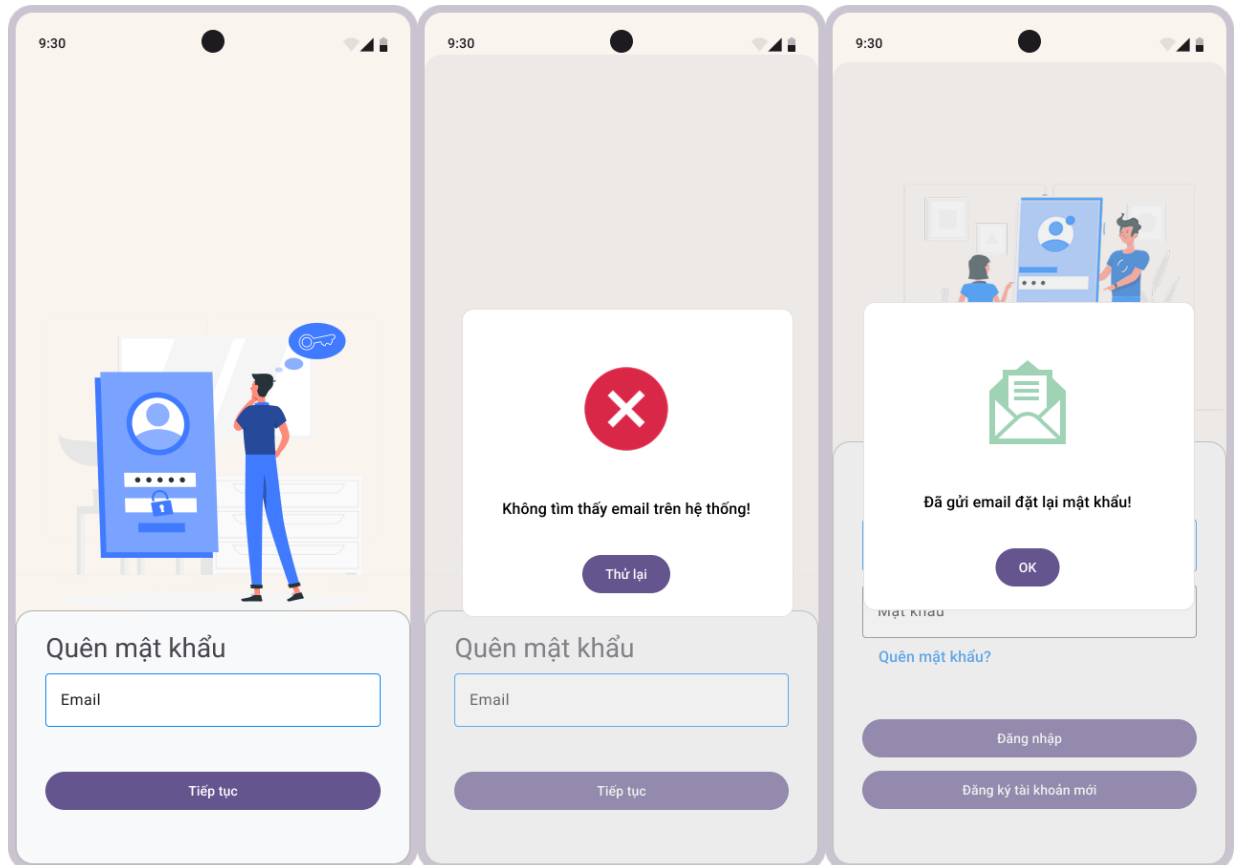
Left Screen (Registration Form):

- Header:** An illustration of a person interacting with a large screen displaying a login form and a smaller screen showing a 'YOUR CODE 2388'.
- Title:** Đăng ký
- Form Fields:**
 - Họ và tên
 - Email
 - Mật khẩu
 - Xác nhận mật khẩu
- Button:** Đăng ký

Right Screen (Success and Login Options):

- Header:** An illustration of two people at a computer.
- Success Message:** Đăng ký thành công! Vui lòng xác nhận email!
- Button:** OK
- Text:** Mật khẩu
- Link:** [Quên mật khẩu?](#)
- Buttons:**
 - Đăng nhập
 - Đăng ký tài khoản mới

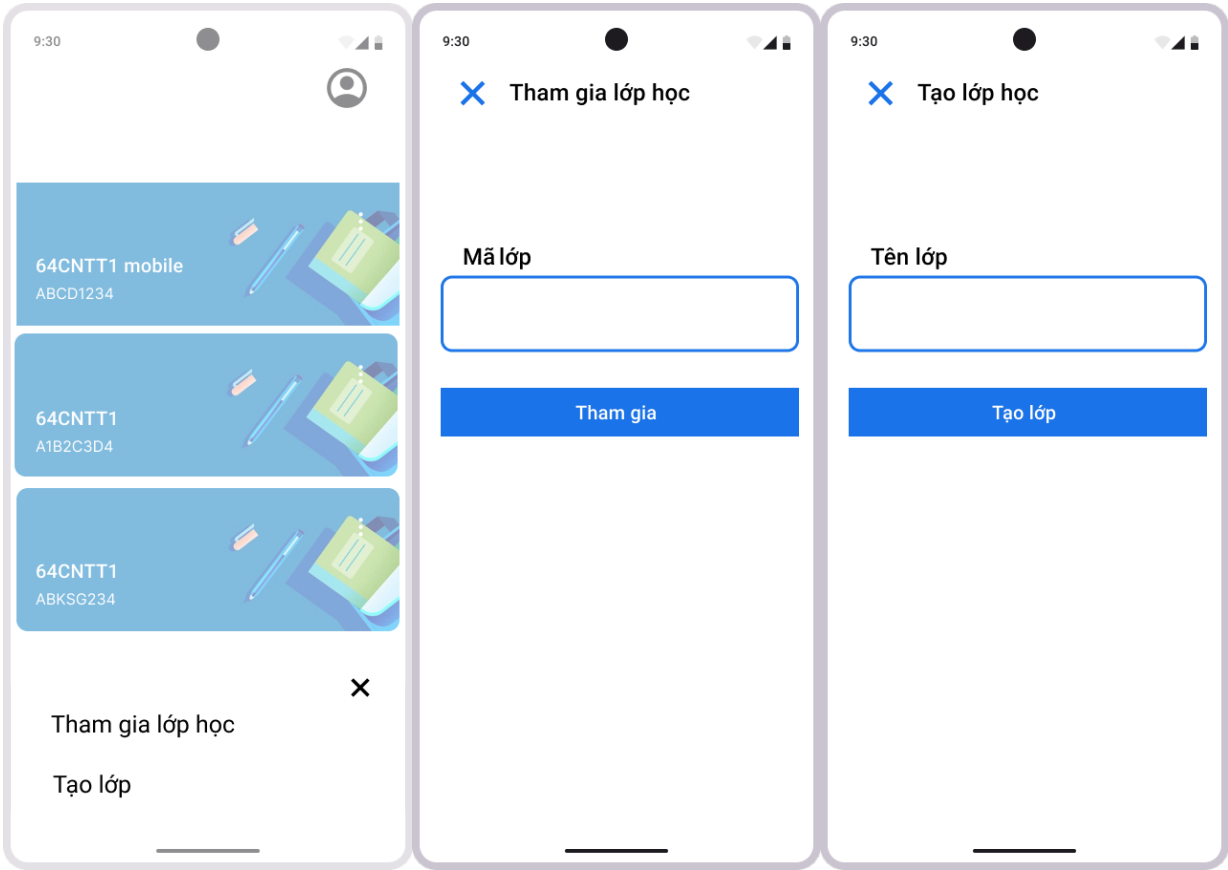
3.1.3. Thiết kế giao diện Quên mật khẩu



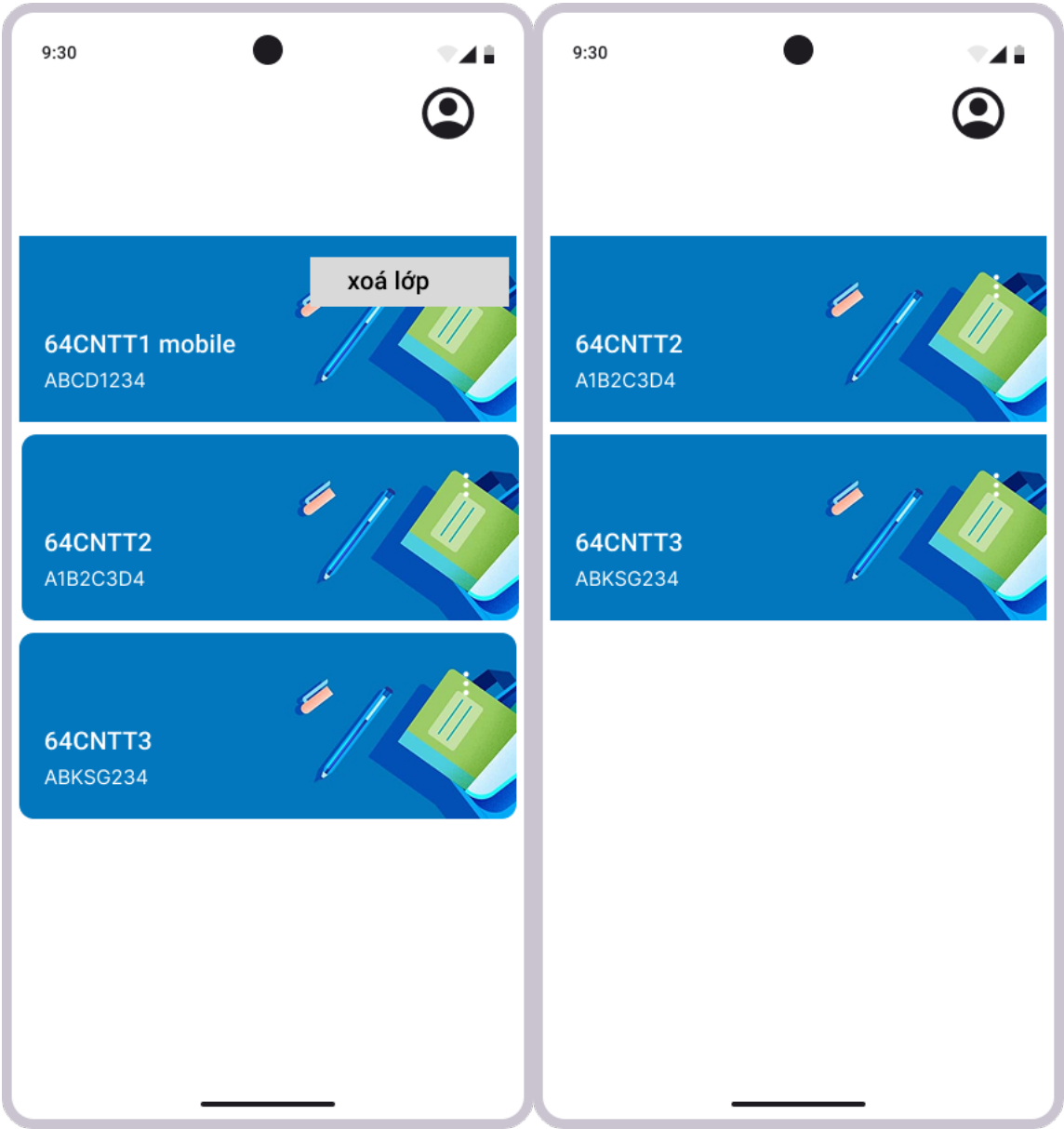
3.1.3. Thiết kế giao diện danh sách lớp



3.1.4. Thiết kế giao diện Tham gia/Tạo lớp học

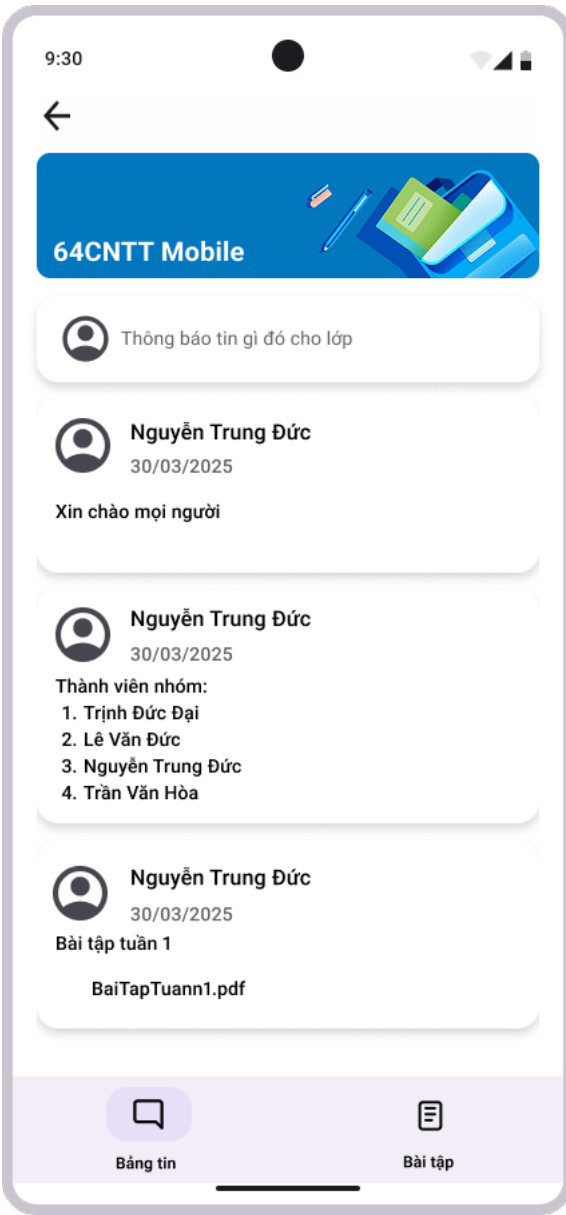


3.1.5. Thiết kế giao diện Xóa lớp học

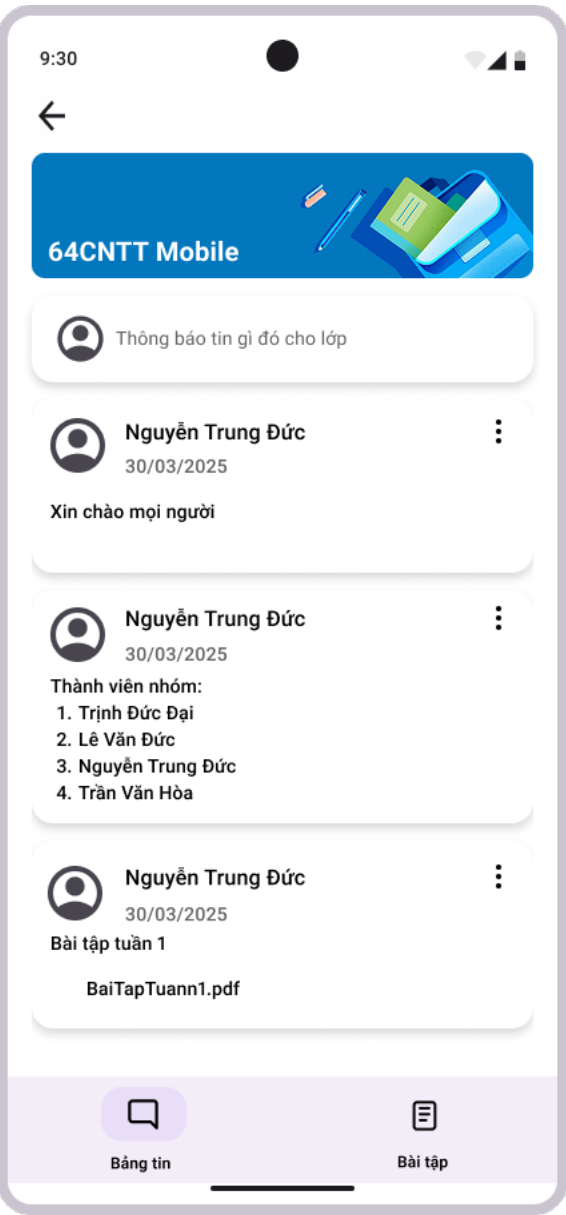


3.1.5. Thiết kế giao diện Bảng tin

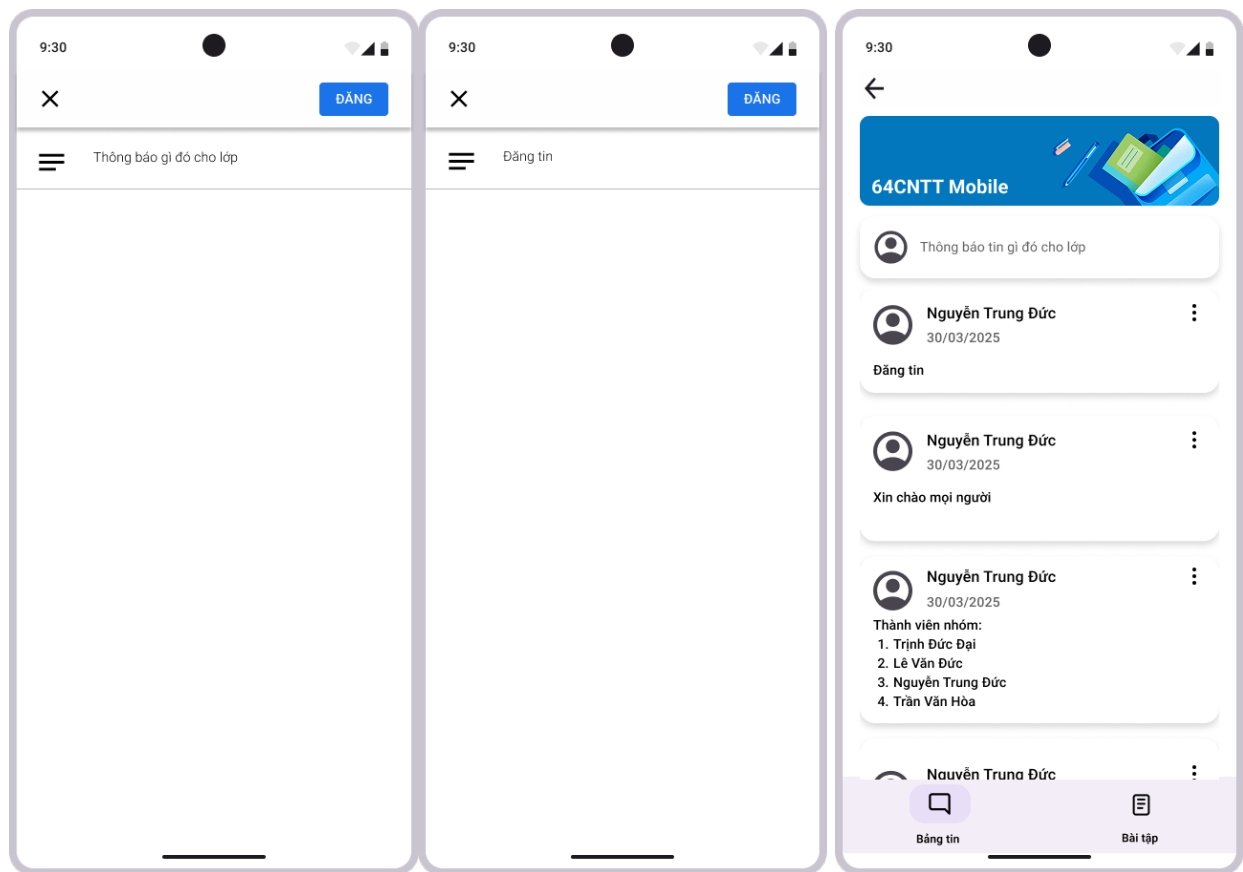
Đối với Student



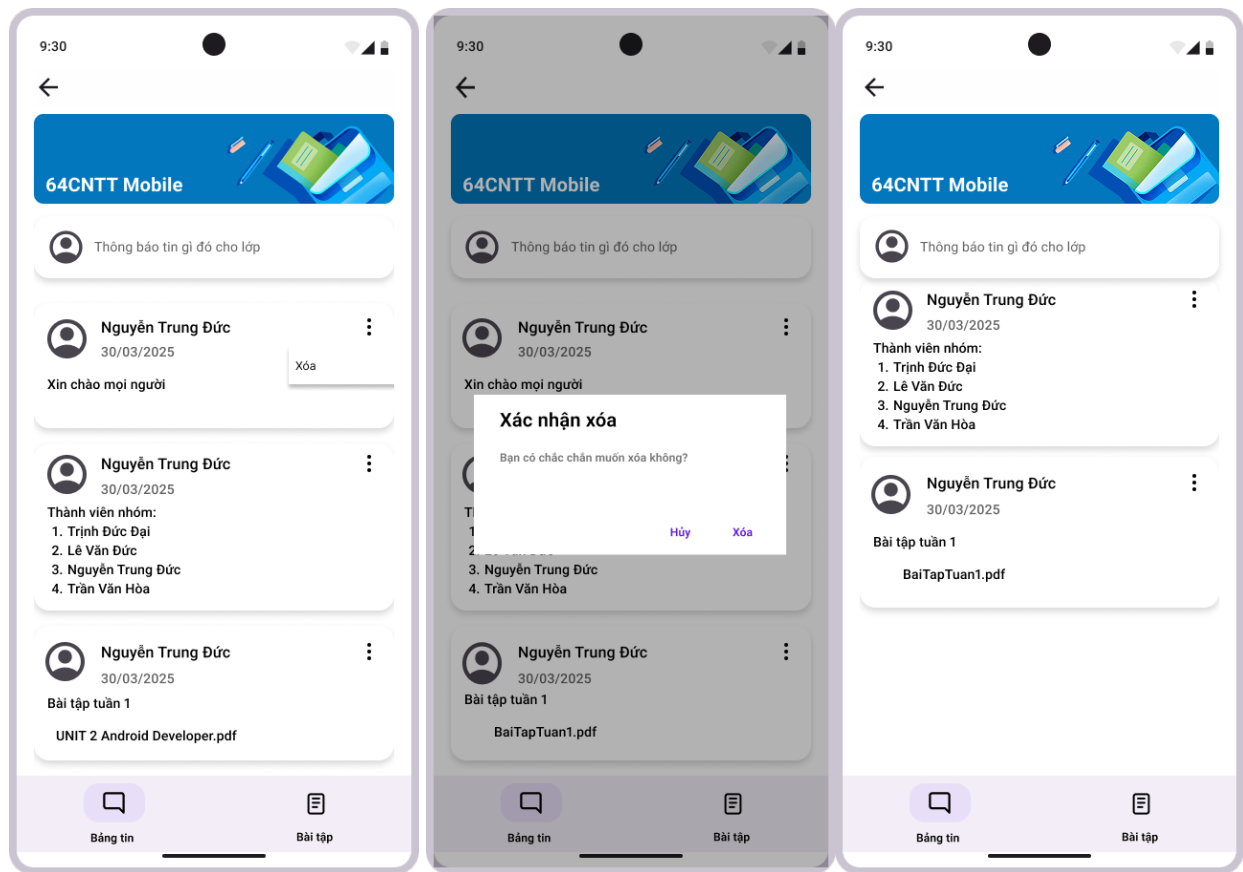
Đối với Teacher



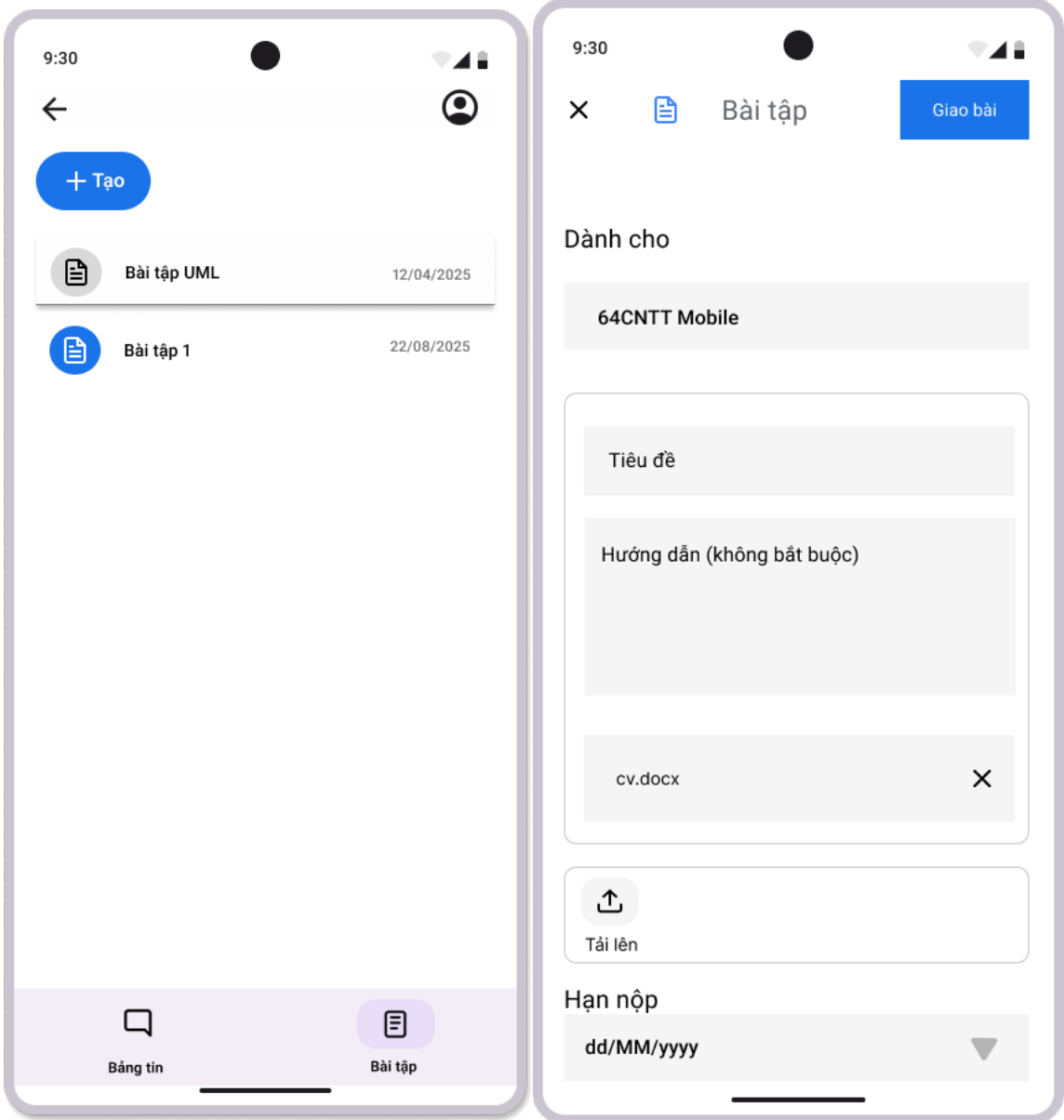
3.1.6. Thiết kế giao diện Đăng tin



3.1.7. Thiết kế giao diện Xóa tin



3.1.8. Thiết kế giao diện Tạo bài tập



3.1.9. Thiết kế giao diện Nộp bài/Chấm điểm

×

Kiểm tra thực hành 2

Tên giáo viên - 13/02/2025

10/100Đến hạn 26/04/2025

Đề = STT * 4 % 2

Bài kiểm tra thực hành.docx

Bài làm của bạn

Đã chấm điểm

Bài làm.txt

×

+ Thêm file

Nộp bài

9:30

×

Trả bài

Lê Văn Đức

10/100

Bài làm.txt

Bài làm.txt

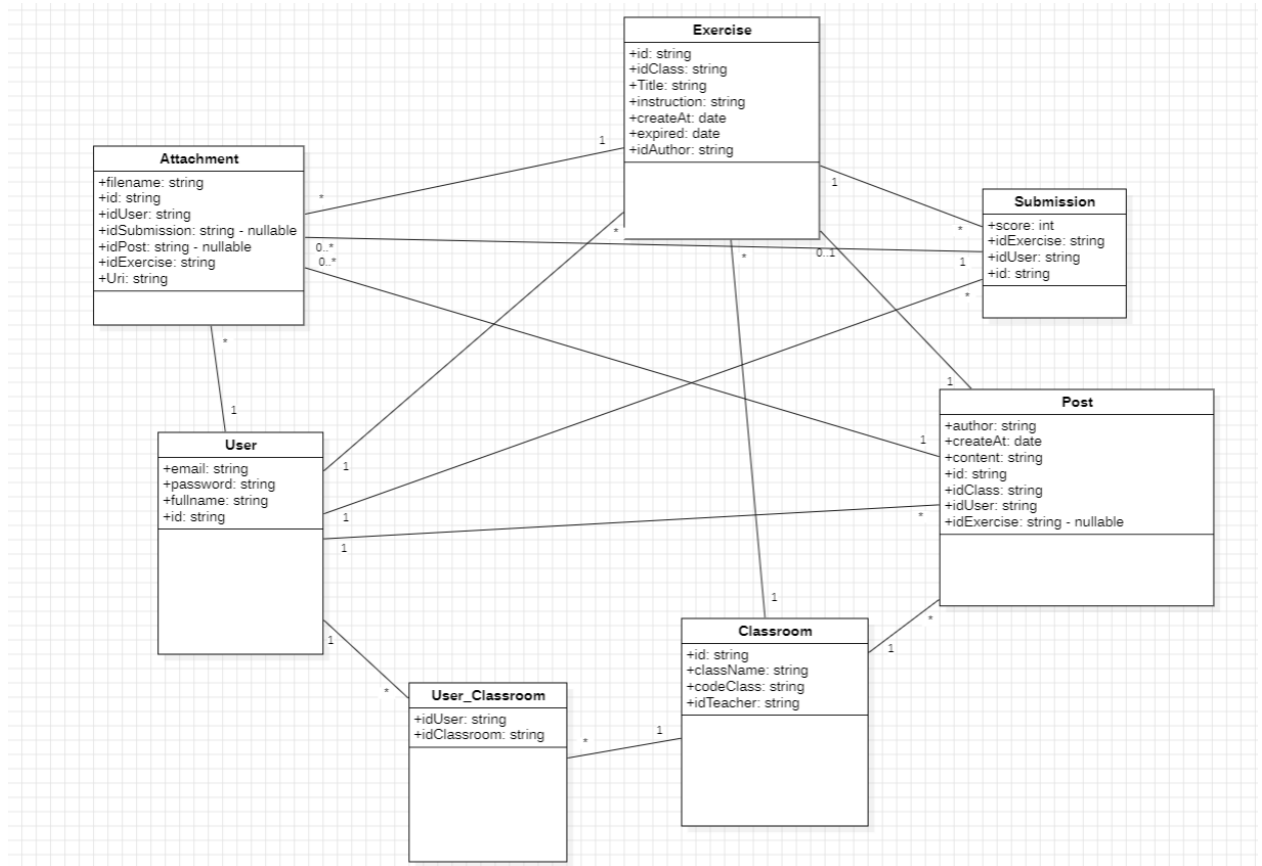
Trần Văn Hòa

50/100

Bài làm.txt

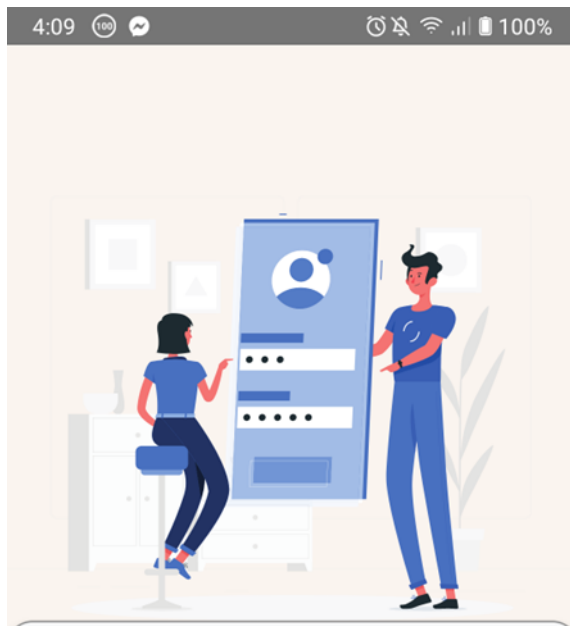
Bài làm.txt

3.2. Thiết kế CSDL



3.3. Giao diện ứng dụng

3.3.1. Màn hình Đăng nhập



4:09 100%

Đăng nhập

Email

nguyennduc2004@gmail.com

Mật khẩu

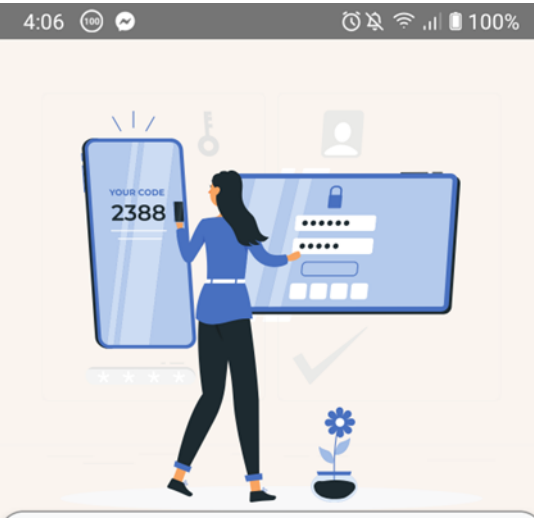
.....

[Quên mật khẩu?](#)

[Đăng nhập](#)

[Đăng ký tài khoản mới](#)

3.3.2. Màn hình Đăng ký



4:06 100% 100%

YOUR CODE
2388

Đăng ký

Họ và tên
NguyenTrung Duc

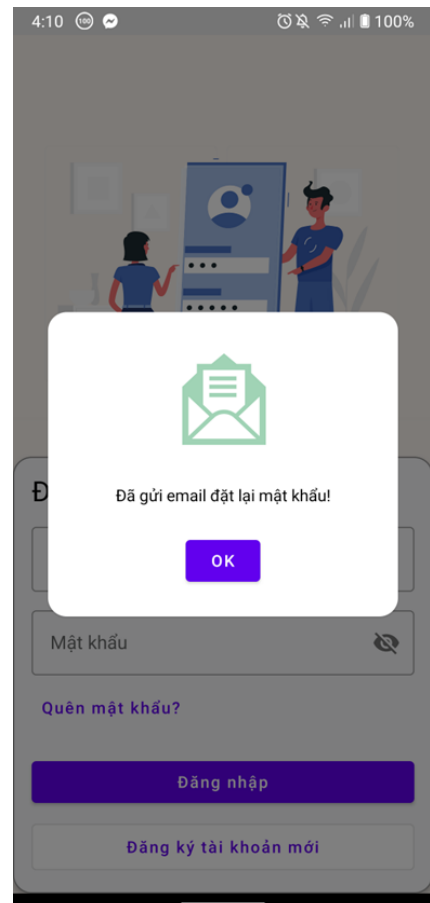
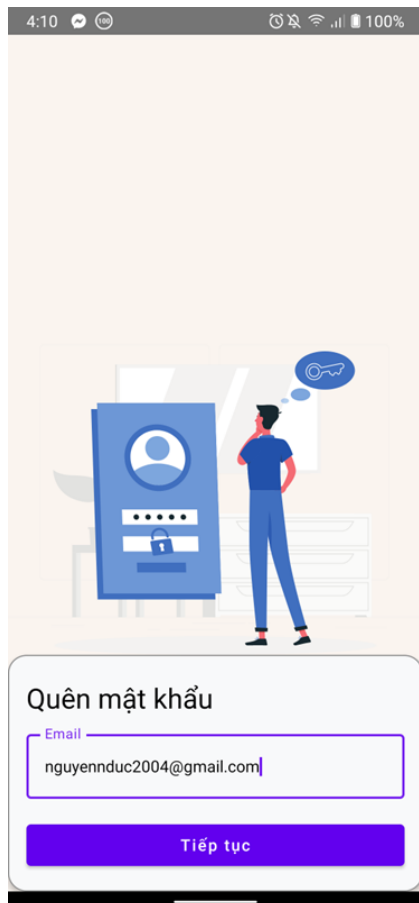
Email
nguyennduc2004@gmail.com

Mật khẩu
.....

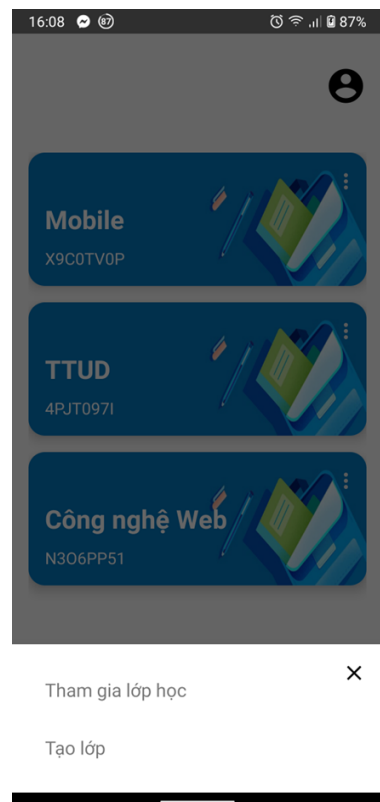
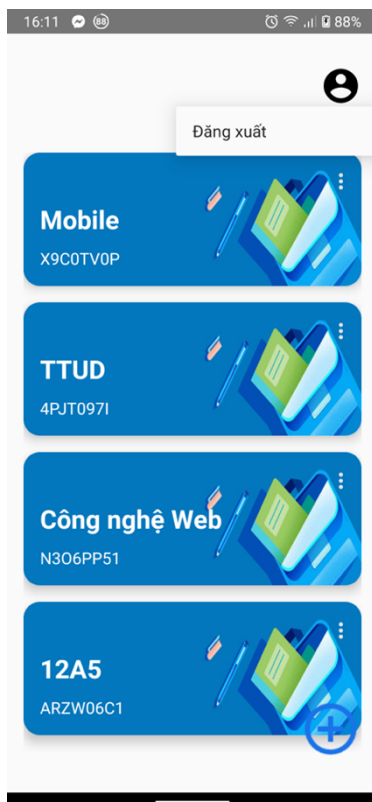
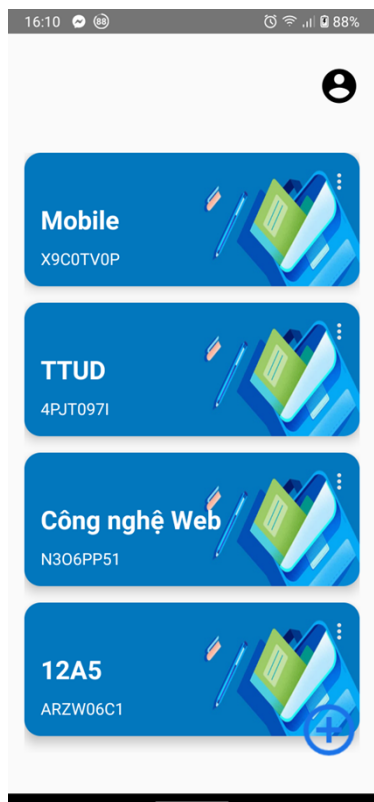
Xác nhận mật khẩu
.....

Đăng ký

3.3.3. Màn hình Quên mật khẩu



3.3.4. Màn hình Danh sách lớp



3.3.5. Màn hình Tạo lớp

16:09 67% 87%

× Tạo lớp học

Tên lớp

64CNTT1

TẠO LỚP

q¹ w² e³ r⁴ t⁵ y⁶ u⁷ i⁸ o⁹ p⁰

a s d f g h j k l

↑ z x c v b n m ↵

?123 VI , Laban Key . Xong

3.3.6. Màn hình Tham gia lớp học

16:10 88% 88%

×

Tham gia
lớp học

Mã lớp

ARZW06C1

THAM GIA

q¹ w² e³ r⁴ t⁵ y⁶ u⁷ i⁸ o⁹ p⁰

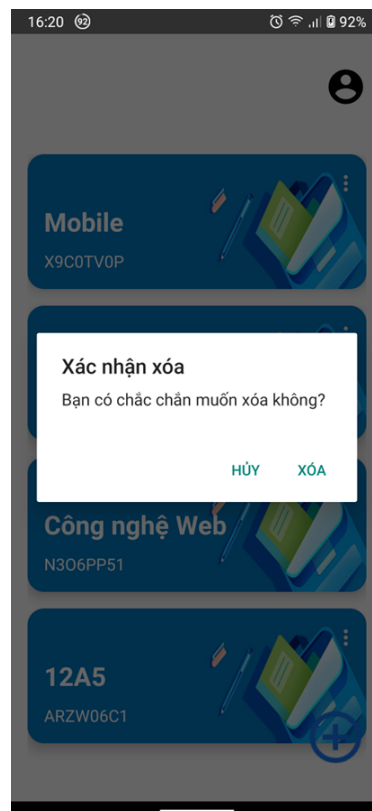
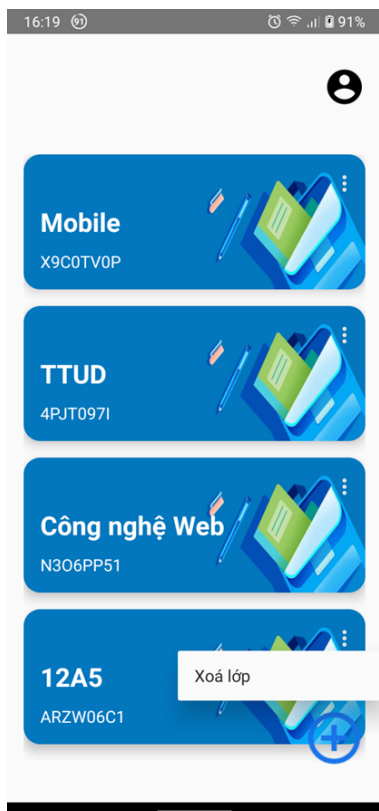
a s d f g h j k l

↑ z x c v b n m ↵

?123 VI , Laban Key . Xong

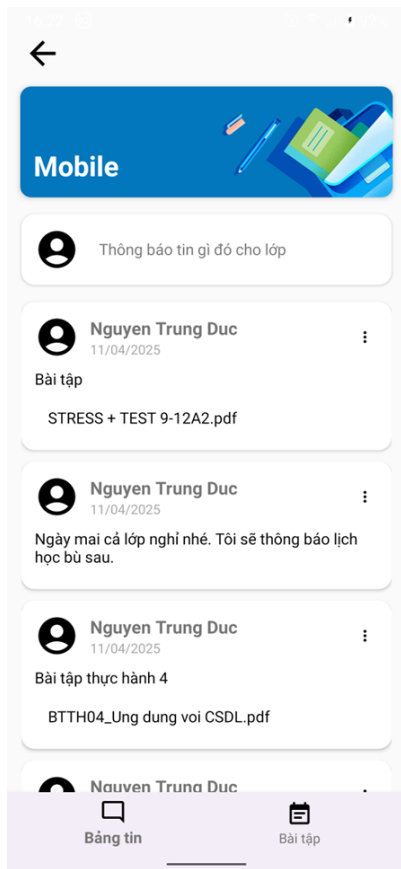
✓

3.3.7. Màn hình Xóa lớp học

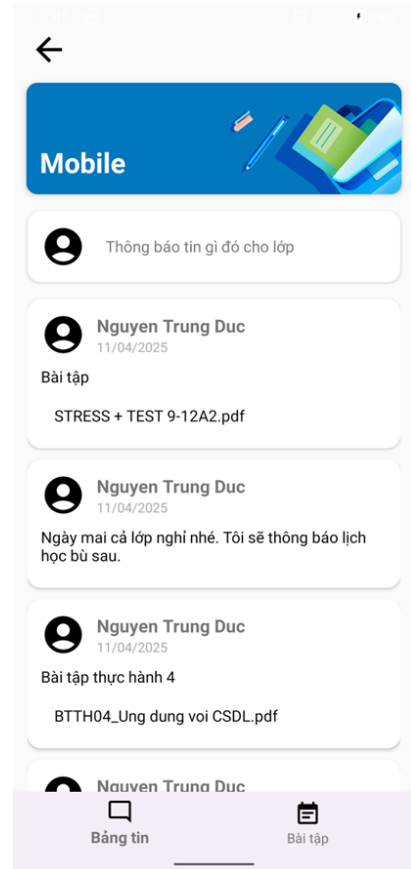


3.3.8. Màn hình Bảng tin

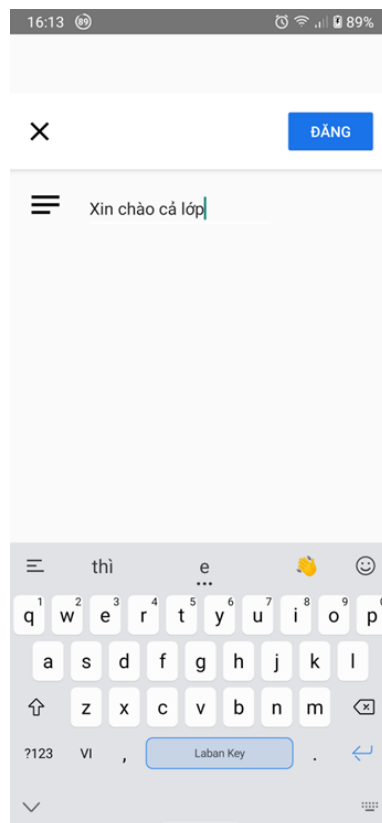
Đối với Teacher



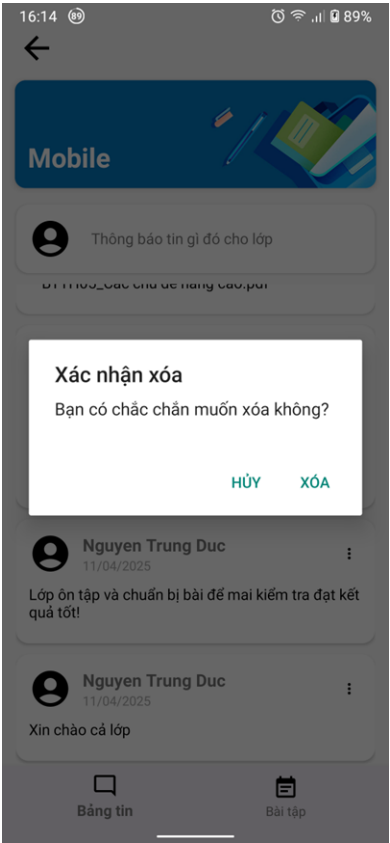
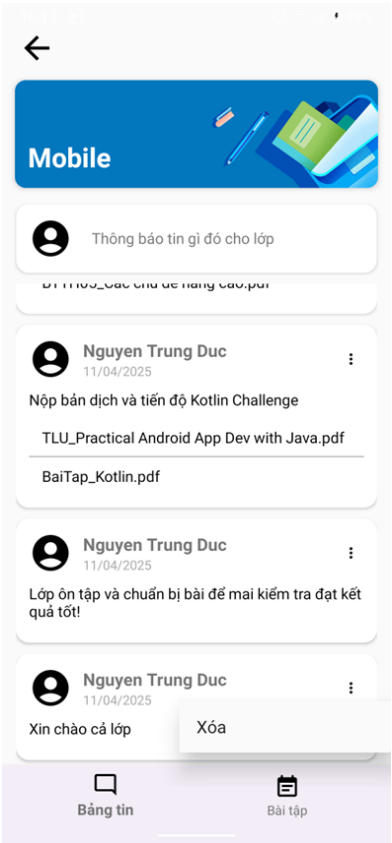
Đối với Student



3.3.9. Màn hình Đăng tin (Teacher)

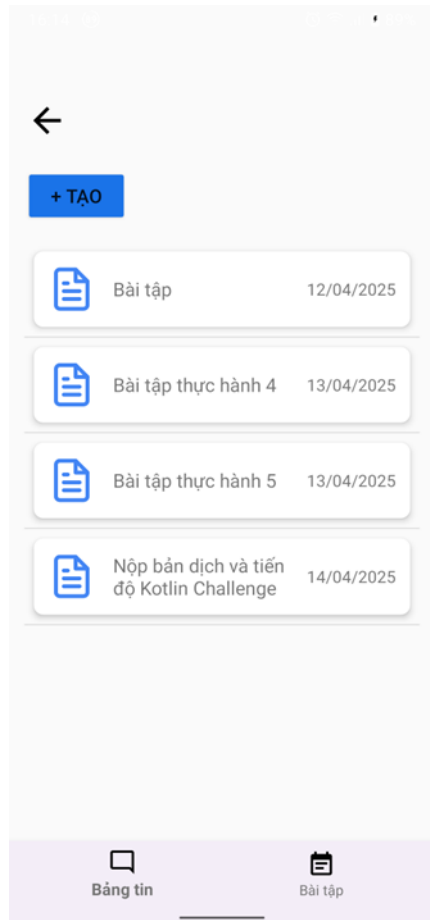


3.3.10. Màn hình Xóa bài đăng (Teacher)

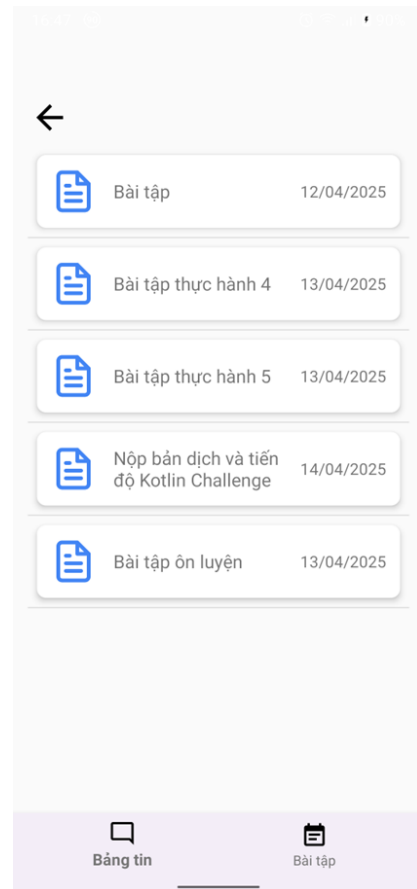


3.3.11. Màn hình Bài tập

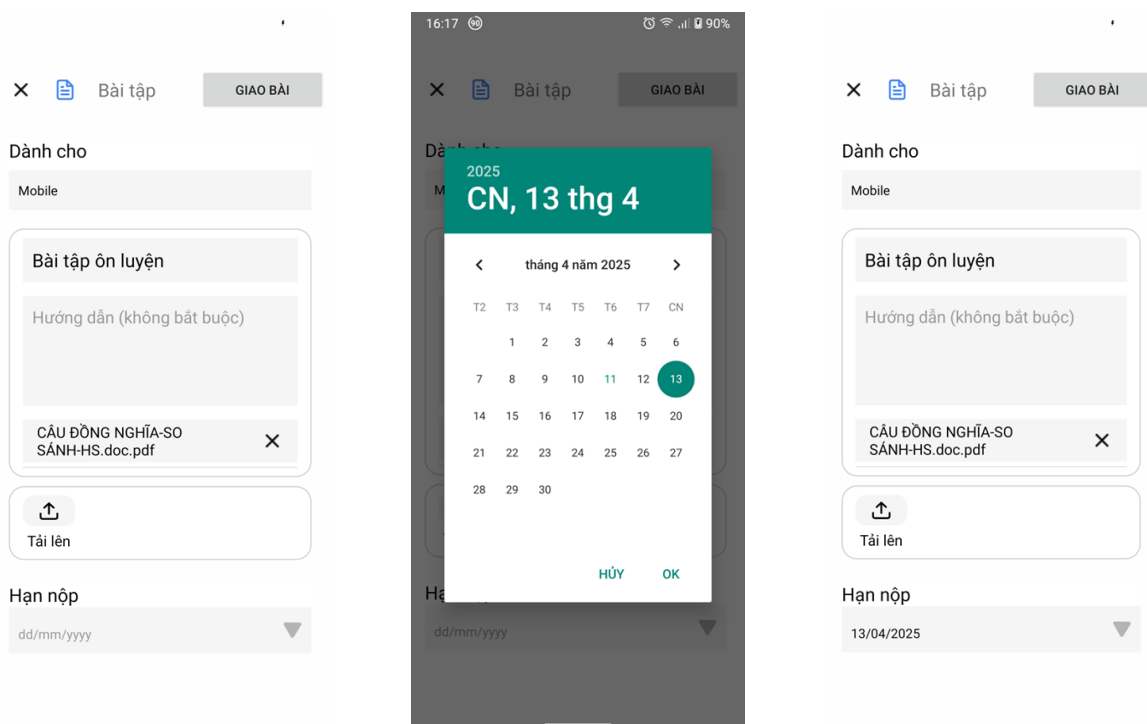
Đối với Teacher



Đối với Student



3.3.12. Màn hình Tạo bài tập (Teacher)



3.3.13. Màn hình Chấm điểm (Teacher)

Trở lại

Điểm số

100/100

✕

Trả bài

Đức Đại

100/100

Tnc the ca KOI 2025.pdf

Tran Van An

60/100

CÂU ĐỒNG NGHĨA-SO SÁNH-HS.doc.pdf

Le Van Duc

0/100

3.3.14. Màn hình Nộp bài (Student)

4:52 3G

X

Nộp bản dịch và tiến độ Kotlin Challenge

Nguyễn Trung Duc-11/04/2025
0/100 Đến hạn 14/04/2025

Làm theo file mẫu

TLU_Practical Android App Dev with Java.pdf

BaiTap_Kotlin.pdf

Bài làm của bạn

+ Thêm file

Nộp bài

3.4. Code minh họa các chức năng cốt lõi

-- **Đăng nhập** (LoginScreen.kt, ComposeActivity.kt, AuthModel.kt)

FUNCTION Login(email, password)

IF email IS empty OR password IS empty THEN

DISPLAY error "Vui lòng nhập đầy đủ thông tin!"

RETURN

END

IF CALL AuthenticateUserWithFirebase(email, password)

IF AuthenticationSuccessful AND EmailVerified THEN

SAVE user session information (UserID, Email, etc. in SharedPreferences)

SET isLoggedIn flag to TRUE in SharedPreferences

NAVIGATE to ListClassScreen

ELSE IF AuthenticationSuccessful AND NOT EmailVerified THEN

DISPLAY error "Vui lòng xác nhận email của bạn!"

ELSE

DISPLAY error "Thông tin đăng nhập không chính xác!"

END IF

END FUNCTION

-- **Đăng ký** (RegistrationScreen.kt, ComposeActivity.kt, AuthModel.kt)

FUNCTION Register(fullName, email, password, confirmPassword)

IF any field (fullName, email, password, confirmPassword) IS empty THEN

DISPLAY error "Vui lòng nhập đầy đủ thông tin!"

RETURN

END IF

IF password != confirmPassword THEN

DISPLAY error "Mật khẩu không khớp!"

RETURN

END IF

IF email IS invalid format THEN


```

        DISPLAY error "Email không hợp lệ!"
        RETURN
    END IF

    IF password length < 6 THEN
        DISPLAY error "Mật khẩu phải có ít nhất 6 ký tự!"
        RETURN
    END IF

    CALL CreateFirebaseUserAccount(fullName, email, password)

    IF AccountCreationSuccessful (gets new FirebaseUser) THEN
        CALL SaveUserDetailsToDatabase(FirebaseUser.UID, fullName, email)
        CALL SendFirebaseVerificationEmail(FirebaseUser)

        DISPLAY success message "Đăng ký thành công! Vui lòng xác nhận email!"

        NAVIGATE back to LoginScreen
    ELSE DISPLAY error (e.g., "Email đã được sử dụng!")
    END IF
END FUNCTION

-- Quên mật khẩu (ForgotPasswordScreen.kt, ComposeActivity.kt, AuthModel.kt)

FUNCTION ForgotPassword(email)
    IF email IS empty THEN
        DISPLAY error "Vui lòng nhập email!"
        RETURN
    END IF

    IF email IS invalid format THEN
        DISPLAY error "Email không hợp lệ!"
        RETURN
    END IF

    CALL SendFirebasePasswordResetEmail(email)

    IF EmailSentSuccessfully THEN

```

```

        DISPLAY success message "Đã gửi email đặt lại mật khẩu!"
        NAVIGATE back to LoginScreen

    ELSE DISPLAY error (e.g., "Email không tồn tại trong hệ thống!")

    END IF

END FUNCTION

-- Đăng xuất (LogoutHelper.java, ListClassActivity.java)

FUNCTION Logout

    CALL FirebaseAuth.SignOut()
    CALL SharedPreferences.Editor().clear().commit()
    (SharedPreferences: isLoggedIn, userID, etc.)

    NAVIGATE to LoginScreen (ComposeActivity) AND CLEAR all previous
    screens in the stack

END FUNCTION

-- Tạo Lớp học (CreateActivity.java, ClassroomAdapter.java)

FUNCTION CreateClassroomScreen

    DISPLAY input field for Class Name

    DISPLAY "Create" button

    DISPLAY "Close" button ON "Create" button click:

    GET ClassName from input field

    IF ClassName IS empty THEN

        DISPLAY error "Tên lớp không được để trống!"

        RETURN

    END IF

    GET current logged-in UserID (as TeacherID) from SharedPreferences
    GENERATE unique 8-character ClassCode (uppercase letters & numbers)
    CALL SaveNewClassroomToDatabase(ClassName, ClassCode, TeacherID)
    IF ClassroomSavedSuccessfully (gets NewClassroomID) THEN

        CALL AddUserToClassroomInDatabase(TeacherID,
        NewClassroomID) // Liên kết giáo viên với lớp

```

```

        DISPLAY success message "Tạo mới lớp học thành công!"
        CLOSE CreateClassroomScreen
    ELSE
        DISPLAY error message "Tạo mới lớp học thất bại!"
    END IF
    ON "Close" button click:
        CLOSE CreateClassroomScreen
END FUNCTION

```

-- Tham gia Lớp học (JoinActivity.java, ClassroomAdapter.java)

```

FUNCTION JoinClassroomScreen
    DISPLAY input field for Class Code
    DISPLAY "Join" button
    DISPLAY "Close" button
    ON "Join" button click:
        GET ClassCode from input field
        IF ClassCode IS empty THEN
            DISPLAY error "Mã lớp không được để trống!"
            RETURN
        END IF
        GET current logged-in UserID from SharedPreferences
        CALL FindClassroomByCode(ClassCode)
        IF Classroom Found (gets ClassroomID) THEN
            CALL CheckIfUserAlreadyInClass(UserID, ClassroomID)
        END IF
        IF User NOT already in class THEN
            CALL AddUserToClassroomInDatabase(UserID,
            ClassroomID)
            DISPLAY success message (optional)
        END IF
    END ON
END FUNCTION

```

```

        CLOSE JoinClassroomScreen
    ELSE
        DISPLAY message "Bạn đã tham gia lớp học này!"
        CLOSE JoinClassroomScreen
    END IF

    ELSE DISPLAY error "Mã lớp không hợp lệ!" // Cần thêm xử lý này // Có
    thể giữ màn hình mở
    END IF

    ON "Close" button click:
        CLOSE JoinClassroomScreen
END FUNCTION

```

-- **Hiện thị Bảng tin** (NewsFeedActivity.java, PostAdapter.java, PostViewHolder.java)

```

FUNCTION DisplayNewsFeedScreen(ClassroomID, UserID, TeacherID)
    CALL FetchClassName(ClassroomID) -> ClassroomName
    DISPLAY ClassroomName in banner
    INITIALIZE empty list for Posts
    SETUP RecyclerView with PostAdapter(Posts, UserID, TeacherID)
    CALL FetchPostsForClassroom(ClassroomID) -> updates Posts list in
    adapter
    IF UserID equals TeacherID THEN
        SHOW "Announce something to your class" input area/button
    ELSE
        HIDE "Announce something to your class" input area/button
    END IF

    DISPLAY Bottom Navigation (News Feed, Exercises) - set News Feed as
    active

```

ON "Announce" area click (if visible):

NAVIGATE to CreatePostScreen with ClassroomID, TeacherID

ON "Exercises" navigation click:

NAVIGATE to ListExerciseScreen with ClassroomID, UserID,
TeacherID

FINISH current NewsFeedScreen

ON Back button click:

FINISH current NewsFeedScreen

END FUNCTION

-- **Tạo Bài đăng** (PostActivity.java, PostRepository.java)

FUNCTION CreatePostScreen(ClassroomID, TeacherID)

DISPLAY EditText for post content

DISPLAY "Post" button

DISPLAY "Close" button

ON "Post" button click:

GET Content from EditText

IF Content IS empty THEN

DISPLAY error "Nội dung không được để trống!"

RETURN

END IF

CALL CreatePostInDatabase(Content, TeacherID, ClassroomID)

FINISH

CreatePostScreen

ON "Close" button click:

FINISH

CreatePostScreen

END FUNCTION

-- **Thêm Bài tập** (AddExerciseActivity.java, ExerciseAdapter.java, AttachmentAdapter.java, AttachmentRepository.java)

FUNCTION AddExerciseScreen(ClassroomID, AuthorID)

 CALL FetchClassName(ClassroomID) -> ClassroomName

 DISPLAY ClassroomName

 INITIALIZE empty list for NewAttachments

 DISPLAY fields for Title, Instruction

 DISPLAY Date Picker trigger for Deadline (shows "No due date" initially)

 DISPLAY "Upload File" button/area

 DISPLAY list area for uploaded NewAttachments (RecyclerView with AttachmentAdapter)

 DISPLAY "Assign" button

 DISPLAY "Close" button

 ON Date Picker trigger click:

 SHOW DatePickerDialog

 ON date selected:

 STORE DeadlineDate

 DISPLAY formatted DeadlineDate (dd/MM/yyyy)

 ON "Upload File" click:

 OPEN system file picker (Intent.ACTION_OPEN_DOCUMENT,
 type "application/pdf")

 ON file selected (returns Uri):

 GET persistable URI permission

 GET FileName from Uri

 CREATE Attachment object (Uri=Uri.toString(),
 FileName=FileName, IdUser=AuthorID, IdExercise=null)

 UPDATE displayed attachment list (adapter)

 ON attachment item click in list (AttachmentObject):

 OPEN AttachmentObject.Uri using Intent.ACTION_VIEW (works
 for local URIs too)

ON attachment delete button click (AttachmentObject):

REMOVE AttachmentObject from NewAttachments list

UPDATE displayed attachment list

ON "Assign" button click:

GET Title, Instruction from fields

GET DeadlineDate selected

IF DeadlineDate IS NOT selected THEN

DISPLAY error "Vui lòng chọn ngày hết hạn"

RETURN

END IF

IF DeadlineDate is BEFORE Today THEN

DISPLAY error "Ngày hết hạn không hợp lệ"

RETURN

END IF

CREATE Exercise object (ID=null, IdClass=ClassroomID, Title, Instruction, CreateAt=Now, Expired=DeadlineDate, IdAuthor=AuthorID)

CALL SaveExerciseAndLocalAttachments(Exercise object, NewAttachments list) -> ExerciseID

IF ExerciseID is valid THEN

CALL

UploadAttachmentFilesToStorageAndUpdateDB(NewAttachments list, "exercises/")

CALL CreateExerciseAnnouncementPost(Exercise.Title, AuthorID, ClassroomID, ExerciseID)

FINISH AddExerciseScreen

END IF

ON "Close" button click:

FINISH AddExerciseScreen

END FUNCTION

-- **Nộp Bài tập** (SubmitAssignmentActivity.java, AttachmentAdapter.java, SubmissionRepository.java, AttachmentRepository.java)

FUNCTION SubmitAssignmentScreen(ExerciseObject, UserID, TeacherID)
DISPLAY ExerciseObject details (Title, Instruction, CreatedAt, Expired)

CALL FetchUserName(TeacherID) -> TeacherName

DISPLAY TeacherName as Author

CALL FetchScoreForSubmission(ExerciseObject.ID, UserID) -> Score

DISPLAY Score (e.g., "Score: {Score}/100") -

INITIALIZE empty list TeacherAttachments

SETUP ListView TeacherAttachmentListView

CALL FetchAttachmentsForExercise(ExerciseObject.ID) -> updates
TeacherAttachments list

DISPLAY filenames from TeacherAttachments in
TeacherAttachmentListView

ON Teacher attachment click (AttachmentObject):

 OPEN AttachmentObject.Uri (should be download URL) using
 Intent.ACTION_VIEW

 SETUP RecyclerView StudentAttachmentRecyclerView with
 AttachmentAdapter(StudentAttachments)

 CALL FetchStudentAttachmentsForSubmission(ExerciseObject.ID,
 UserID)

 IF CurrentTime > ExerciseObject.Expired THEN

 HIDE "Add File" button

 HIDE "Submit" button

 DISPLAY message "Past Due" (optional)

 ELSE

 SHOW "Add File" button

 SHOW "Submit" button

 END IF

 DISPLAY "Close" button

ON "Add File" click (if visible):

 OPEN system file picker (PDF)

 ON file selected (returns Uri):

 GET persistable URI permission

 GET FileName from Uri

 CREATE Attachment object (Uri=Uri.toString(),
 FileName=FileName, IdUser=UserID,
 IdExercise=ExerciseObject.ID)

 ADD Attachment to StudentAttachments list (via
 adapter)

 UPDATE displayed student attachment list

 ON student attachment delete click (AttachmentObject):
 REMOVE AttachmentObject from StudentAttachments
 list (via adapter)

 UPDATE displayed list

ON "Submit" click (if visible):

 CREATE Submission object (ID=null, Score=0,
 IdExercise=ExerciseObject.ID, IdUser=UserID)

 GET current list of Attachments from
 StudentAttachmentAdapter -> SubmittedAttachmentsList
 CALL

 SaveSubmissionAndStudentAttachments(Submission object,
 SubmittedAttachmentsList) -> SubmissionID

 IF SubmissionID IS valid THEN

 CALL

 UploadAttachmentFilesToStorageAndUpdateDB(SubmittedAttachmentsList, "assignments/")

 FINISH SubmitAssignmentScreen

 END IF

ON "Close" button click:

 FINISH SubmitAssignmentScreen

END FUNCTION

-- **Chăm điểm Bài tập** (GradeActivity.java, SubmissionAdapter.java, SubmissionRepository.java)

FUNCTION GradeScreen(ClassroomID, ExerciseID, TeacherID)

 INITIALIZE empty List<User> StudentsInClass

 INITIALIZE empty Map<StudentID, List<Attachment>>
StudentSubmissionsMap

 SETUP RecyclerView GradingRecyclerView with
SubmissionAdapter(StudentsInClass, StudentSubmissionsMap, ExerciseID)

 CALL FetchStudentsInClass(ClassroomID, TeacherID) -> updates
StudentsInClass list and triggers adapter update

 FOR each Student in StudentsInClass:

 CALL FetchStudentAttachmentsForGrading(ExerciseID, Student.ID)
-> updates StudentSubmissionsMap for that student and triggers adapter
update

 DISPLAY "Submit Grades" button/text

 DISPLAY "Close" button

 ON "Submit Grades" click:

 FOR i = 0 to GradingRecyclerView.childCount - 1:

 GET ViewHolder for item at index i

 GET StudentID associated with ViewHolder

 GET Score from EditText in ViewHolder

 CALL UpdateStudentScoreInDatabase(StudentID, ExerciseID, Score)

 FINISH GradeScreen

 ON "Close" button click:

 FINISH GradeScreen

END FUNCTION

KẾT LUẬN

1. Kết quả đạt được

1. Ứng dụng YourClassroom đã được phát triển thành công trên nền tảng Android, hỗ trợ quản lý lớp học, giao bài tập, và lưu trữ dữ liệu.
2. Giao diện người dùng thân thiện, trực quan, giúp giáo viên và học sinh dễ dàng sử dụng.
3. Tích hợp Firebase:
 - Lưu trữ dữ liệu bài tập và tệp đính kèm.
 - Đồng bộ dữ liệu theo thời gian thực.

2. Nhược điểm

1. Hạn chế trong việc mở rộng:
 - Kiến trúc MVC có thể trở nên phức tạp và khó bảo trì khi ứng dụng mở rộng quy mô.
2. Hiệu suất:
 - Firebase có thể gặp giới hạn về hiệu suất khi số lượng lớn người dùng truy cập đồng thời.
3. Tính năng hạn chế:
 - Hiện tại, ứng dụng chỉ hỗ trợ một số định dạng file (ví dụ: PDF) và chưa hỗ trợ các định dạng khác.
4. Thiếu các tính năng nâng cao:
 - Chưa có tính năng đánh giá tự động hoặc tích hợp AI để hỗ trợ việc học tập.
5. Lưu phiên người dùng:
 - Hiện tại ứng dụng đang lưu phiên người dùng bằng cách lưu thông tin vào Shared Preferences

3. Hướng phát triển

1. Nâng cấp kiến trúc hệ thống:
 - Chuyển đổi sang kiến trúc MVVM để dễ dàng bảo trì và mở rộng ứng dụng.
2. Cải thiện hiệu suất:
 - Tối ưu hóa việc lưu trữ và truy vấn dữ liệu Firebase.
 - Hỗ trợ caching để giảm tải cho server.
3. Mở rộng tính năng:
 - Hỗ trợ nhiều định dạng file khác nhau như hình ảnh, video, hoặc tài liệu Word.

- Tích hợp chatbot AI để hỗ trợ học sinh trong các câu hỏi thường gặp.
4. Tối ưu giao diện người dùng:
 - Thiết kế giao diện đáp ứng (responsive) để tương thích tốt hơn trên các kích thước màn hình khác nhau.
 5. Hỗ trợ đa ngôn ngữ:
 - Cung cấp phiên bản đa ngôn ngữ để mở rộng thị trường sang các quốc gia khác.
 6. Cập nhật phương thức lưu phiên người dùng:
 - Chuyển đổi sang sử dụng phương thức lưu phiên người dùng tối ưu hơn

TÀI LIỆU THAM KHẢO

[1] Google Classroom

[2] Authenticate with Firebase using Password-Based Accounts on Android
(<https://firebase.google.com/docs/auth/android/password-auth>)