

**BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN  
TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP LỚN  
HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG  
ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG ÔN THI BẰNG LÁI XE MÁY A1 A2**

<b>Mã Sinh Viên</b>	<b>Họ và Tên</b>	<b>Ngày Sinh</b>	<b>Lớp</b>
<b>2151173747</b>	<b>Nguyễn Xuân Tùng Anh</b>	<b>17/06/2003</b>	<b>63KTPM1</b>
<b>2151173772</b>	<b>Trần Hiếu</b>	<b>02/04/2003</b>	<b>63KTPM1</b>
<b>2151170581</b>	<b>Đỗ Thị Trang</b>	<b>31/05/2003</b>	<b>63KTPM1</b>
<b>2151170558</b>	<b>Đặng Việt Anh</b>	<b>10/12/2003</b>	<b>63KTPM1</b>

**Hà Nội, năm 2024**

**BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN  
TRƯỜNG ĐẠI HỌC THỦY LỢI**



**PHIẾU CHẤM ĐIỂM**

Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
			Bảng Số	Bảng Chữ
2151173747	Nguyễn Xuân Tùng Anh	17/06/2003		
2151173772	Trần Hiếu	02/04/2003		
2151170581	Đỗ Thị Trang	31/05/2003		
2151170558	Đặng Việt Anh	10/12/2003		

**CÁN BỘ CHẤM THI 1**

**CÁN BỘ CHẤM THI 2**

**Hà Nội, năm 2024**

## MỤC LỤC

<b>PHIẾU CHẤM ĐIỂM.....</b>	<b>2</b>
<b>MỤC LỤC .....</b>	<b>3</b>
<b>LỜI CAM KẾT.....</b>	<b>5</b>
<b>LỜI CẢM ƠN .....</b>	<b>6</b>
<b>LỜI NÓI ĐẦU.....</b>	<b>7</b>
<b>BẢNG PHÂN CHIA CÔNG VIỆC.....</b>	<b>8</b>
<b>CHƯƠNG 1. MÔ TẢ BÀI TOÁN.....</b>	<b>9</b>
<b>1.1. Giới thiệu .....</b>	<b>9</b>
<b>1.2. Chức năng chính .....</b>	<b>9</b>
<b>1.3. Yêu cầu phi chức năng .....</b>	<b>9</b>
<b>CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG.....</b>	<b>11</b>
<b>2.1. Phân tích yêu cầu .....</b>	<b>11</b>
<b>2.2. Thiết kế hệ thống .....</b>	<b>12</b>
2.2.1.Thiết kế kiến trúc (MVC) .....	13
❖ Model .....	13
❖ View .....	13
❖ Controller .....	14
2.2.2. Dựa trên kiến trúc MVC đã chọn, ta có thể xác định các lớp sau .....	14
2.2.3. Mối quan hệ giữa các lớp .....	16
2.2.3.1.Mối quan hệ giữa các lớp Model .....	16
2.2.3.2. Mối quan hệ giữa các lớp Controller và Model .....	17
2.2.3.3. Mối quan hệ giữa các lớp View và Controller .....	18
<b>2.3. Triển khai .....</b>	<b>22</b>
<b>2.4. Vận hành và bảo trì .....</b>	<b>23</b>
<b>CHƯƠNG 3. KẾT QUẢ THỰC HIỆN .....</b>	<b>24</b>
<b>3.1. Công nghệ đã sử dụng .....</b>	<b>24</b>
<b>3.2. Tiến độ thực hiện .....</b>	<b>24</b>
<b>3.3.Hình ảnh sản phẩm .....</b>	<b>32</b>
3.3.1.Giao diện chung.....	32
3.3.2. Chức năng đề ngẫu nhiên .....	35
3.3.3. Chức năng ôn tập theo câu hỏi .....	39

3.3.4. Chức năng ôn tập theo chủ đề, nhóm câu hỏi .....	42
3.3.5. Chức năng thi theo bộ đề.....	45
<b>KẾT LUẬN .....</b>	<b>52</b>
<b>1. Kết quả đạt được .....</b>	<b>52</b>
<b>2. Hướng phát triển .....</b>	<b>52</b>
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>53</b>
<b>QUY ĐỊNH TRÌNH BÀY TRONG BÁO CÁO CÀI TẬP LỚN.....</b>	<b>Error! Bookmark not defined.</b>

## **LỜI CAM KẾT**

Nhóm chúng em xin cam đoan tất cả những thông tin, dữ liệu và kiến thức được trình bày trong bài tập lớn này là hoàn toàn trung thực và đáng tin cậy. Nhóm chúng em đã tự thực hiện công việc nghiên cứu, thu thập dữ liệu, phân tích dữ liệu. Tất cả các nguồn tài liệu tham khảo liên quan đều được chúng em trích dẫn và ghi lại nguồn tài liệu tham khảo đầy đủ và chính xác.

Nhóm chúng em cũng cam kết rằng đã tuân thủ đầy đủ các quy định và hướng dẫn của giảng viên và trường đại học đề ra trong quá trình thực hiện bài tập lớn này.

## LỜI CẢM ƠN

Sau bốn năm học tập và nghiên cứu tại Khoa Công Nghệ Thông Tin, Trường Đại Học Thủy Lợi. Chúng em đã được trải nghiệm trong môi trường đào tạo chất lượng và chuyên nghiệp, chúng em đã được sự chỉ dạy nhiệt tình cùng hướng dẫn tận tâm của các thầy cô trong Khoa và các thầy cô trong Trường.

Trước hết, chúng em xin được bày tỏ lòng biết ơn tới các thầy cô giáo trong Khoa Công Nghệ Thông Tin nói riêng và các thầy cô Trường Đại học Thủy Lợi nói chung đã trang bị cho chúng em những kiến thức quý báu làm hành trang giúp sức chúng em trong những năm nghiên cứu và học tập sắp tới.

Chúng em xin được gửi lời cảm ơn chân thành đến ThS. Kiều Tuấn Dũng đã hỗ trợ và giúp đỡ chúng em, trong quá trình thực hiện bài tập lớn này. Nhờ sự cung cấp những kiến thức và kỹ năng quý báu để thực hiện bài tập lớn một cách hiệu quả. Những lời khuyên, định hướng và phản hồi của cô đã giúp chúng em giải quyết các khó khăn và thắc mắc trong quá trình thực hiện bài tập.

Trong quá trình làm bài tập này, mặc dù cá nhân chúng em đã rất nỗ lực nhưng đồ án cũng như sản phẩm không thể tránh khỏi những thiếu sót do còn hạn chế về kiến thức và kinh nghiệm thực tế cũng như thời gian. Chúng em rất mong nhận được sự cảm thông và những góp ý từ Thầy Cô và các bạn để đề tài được hoàn thiện và phát triển hơn.

Nhóm chúng em xin chân thành cảm ơn.

## LỜI NÓI ĐẦU

Trong bối cảnh nhu cầu thi bằng lái xe máy ngày càng gia tăng, việc ôn luyện kiến thức để đạt kết quả tốt trong kỳ thi là một yêu cầu cần thiết đối với nhiều người. Hiểu được điều này, chúng em đã chọn dự án “Xây dựng ứng dụng ôn thi bằng lái xe máy A1, A2” nhằm hỗ trợ người dùng chuẩn bị cho kỳ thi bằng cách cung cấp một nền tảng học tập hiệu quả, dễ sử dụng và tiện lợi.

Ứng dụng không chỉ cung cấp bộ câu hỏi theo chuẩn của kỳ thi, mà còn giúp người dùng làm quen với các dạng câu hỏi và tình huống giao thông thực tế. Qua đó, người dùng có thể nắm vững kiến thức, luyện tập các bài thi thử, và tự tin hơn khi tham gia kỳ thi chính thức.

Dự án này là cơ hội để chúng em áp dụng kiến thức lập trình Android đã học, đồng thời phát triển thêm các kỹ năng phân tích yêu cầu, thiết kế giao diện, và xử lý dữ liệu người dùng. Quá trình xây dựng ứng dụng đã giúp chúng em cải thiện kỹ năng làm việc nhóm, giải quyết vấn đề và quản lý thời gian một cách hiệu quả.

Chúng em xin gửi lời cảm ơn đến giảng viên đã luôn tận tình hướng dẫn, cung cấp những kiến thức bổ ích, và đồng hành cùng chúng em trong quá trình thực hiện dự án. Đồng thời, chúng em trân trọng cảm ơn sự đóng góp và nỗ lực của tất cả các thành viên trong nhóm, những người đã cống hiến hết mình để hoàn thành dự án này.

### **BẢNG PHÂN CHIA CÔNG VIỆC**

<b>STT</b>	<b>Tên thành viên</b>	<b>Nội dung công việc</b>
<b>1</b>	Nguyễn Xuân Tùng Anh	<ul style="list-style-type: none"><li>▪ Chức năng Thi theo bộ đề</li><li>▪ Chức năng Xem lịch sử</li></ul>
<b>2</b>	Trần Hiếu	<ul style="list-style-type: none"><li>▪ Chức năng ôn tập theo câu hỏi</li></ul>
<b>3</b>	Đỗ Thị Trang	<ul style="list-style-type: none"><li>▪ Chức năng ôn tập theo chủ đề</li></ul>
<b>3</b>	Đặng Việt Anh	<ul style="list-style-type: none"><li>▪ Chức năng thi theo đề ngẫu nhiên</li></ul>



# CHƯƠNG I. MÔ TẢ BÀI TOÁN

## 1.1. Giới thiệu

- Ứng dụng Ôn Thi Bằng Lái Xe Máy A1,A2 là một công cụ hỗ trợ học lý thuyết thi bằng lái xe máy, cho phép người dùng làm bài thi,xem toàn bộ câu hỏi,làm theo từng chủ đề,tạo đề ngẫu nhiên và mặc định. Ứng dụng sẽ được xây dựng bằng Java và chạy trên giao diện console (command-line).

## 1.2. Chức năng chính

❖ Ứng dụng ôn thi bằng lái xe máy A1,A2 cần có các chức năng sau:

- **Đề ngẫu nhiên:** Chức năng này cho phép người dùng làm bài thi với các câu hỏi được chọn ngẫu nhiên từ ngân hàng câu hỏi. Nó giúp người dùng ôn tập linh hoạt và đánh giá khả năng nhớ bài mà không bị ràng buộc bởi thứ tự câu hỏi cố định.
- **Thi theo bộ đề:** Chức năng này cung cấp các bài thi theo các bộ đề chính thức đã được sắp xếp theo cấu trúc đề thi GPLX A1,A2. Người dùng có thể luyện tập giống như khi tham gia kỳ thi thật, giúp chuẩn bị tốt hơn cho phần thi lý thuyết.
- **200 câu hỏi:** Chức năng này cung cấp danh sách 200 câu hỏi lý thuyết chính thức trong kỳ thi GPLX A1,A2. Người dùng có thể xem và học thuộc từng câu hỏi và đáp án trước khi tham gia thi..
- **Chủ đề:** Chức năng này phân chia các câu hỏi thành các chủ đề cụ thể, giúp người dùng ôn tập theo từng chủ đề (ví dụ: khái niệm và quy tắc giao thông đường bộ, văn hóa và đạo đức, biển báo và đường bộ, v.v.), từ đó củng cố kiến thức theo lĩnh vực.
- **Xem lịch sử làm bài:** Chức năng mới này cho phép người dùng xem lại lịch sử các bài thi đã làm trước đó, bao gồm số điểm, thời gian hoàn thành, và danh sách câu hỏi đã trả lời đúng/sai. Điều này giúp người dùng theo dõi quá trình học tập và cải thiện kết quả thi.

## 1.3. Yêu cầu phi chức năng

- **Dễ sử dụng:**
  - **Yêu cầu:** Giao diện người dùng cần rõ ràng, trực quan và dễ sử dụng, đặc biệt là đối với người dùng không rành về công nghệ.
  - **Hiệu năng:** Ứng dụng cần hoạt động nhanh chóng và hiệu quả, ngay cả khi có nhiều công việc.
  - **Mục tiêu:** Người dùng có thể dễ dàng chọn các tính năng như "Thi theo bộ đề", "200 câu hỏi", "Đề ngẫu nhiên", v.v. mà không gặp khó khăn.
  - **Kết quả mong đợi:** Thời gian học cách sử dụng ứng dụng dưới 5 phút, không cần hướng dẫn chi tiết.
- **Độ tin cậy:**

- **Yêu cầu:** Dữ liệu như câu hỏi thi, lịch sử làm bài phải được lưu trữ và bảo quản một cách an toàn, không bị mất mát dữ liệu khi tắt máy hoặc mất kết nối.
- **Mục tiêu:** Đảm bảo tỷ lệ mất dữ liệu là 0%, đồng thời lịch sử làm bài thi có thể truy cập mọi lúc.
- **Kết quả mong đợi:** Người dùng có thể yên tâm làm bài thi mà không lo mất dữ liệu giữa chừng.
- **Hiệu năng:**
  - **Yêu cầu:** Ứng dụng cần hoạt động mượt mà, phản hồi nhanh chóng ngay cả khi có nhiều câu hỏi hoặc dữ liệu lịch sử làm bài.
  - **Mục tiêu:** Thời gian phản hồi cho mọi thao tác phải dưới 2 giây.
  - **Kết quả mong đợi:** Không có độ trễ đáng kể khi chuyển qua lại giữa các chức năng hoặc khi thực hiện các thao tác truy vấn lịch sử làm bài thi.

## CHƯƠNG II. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Phân tích yêu cầu

#### ❖ Xác định người dùng:

- **Đối tượng:** Bất kỳ ai chuẩn bị tham gia kỳ thi lấy giấy phép lái xe máy hạng A1 hoặc A2, bao gồm: Người dùng có thể có các mức độ am hiểu về máy tính khác nhau, do đó ứng dụng cần dễ sử dụng và thân thiện với người dùng.
  - **Học sinh, sinh viên:** Những người trẻ tuổi, lần đầu thi bằng lái xe.
  - **Người đi làm:** Nhân viên văn phòng, công nhân, hoặc bất kỳ ai cần bằng lái xe máy để di chuyển.
  - **Người cao tuổi:** Những người trung niên hoặc lớn tuổi có nhu cầu thi bằng lái xe máy để tuân thủ luật pháp.
- **Mục đích:** Sử dụng ứng dụng để ôn luyện kiến thức lý thuyết và chuẩn bị tốt nhất cho kỳ thi lấy giấy phép lái xe máy hạng A1, A2.

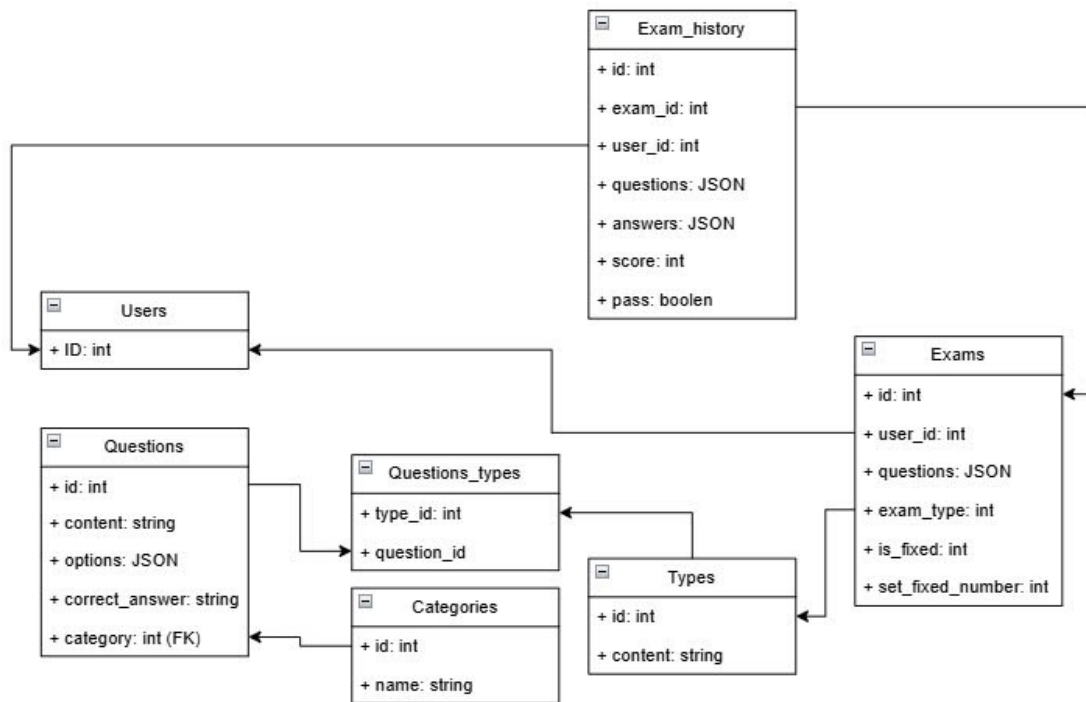
#### ❖ Thu thập yêu cầu: Dựa trên mô tả bài toán, ta đã xác định được

- Chức năng chính (Thi theo đề ngẫu nhiên, thi theo đề bộ, ôn tập 200 câu và 450 câu, ôn theo chủ đề, xem lịch sử bài làm)
- Yêu cầu phi chức năng (dễ sử dụng, hiệu năng, độ tin cậy) của ứng dụng.

#### ❖ Phân tích yêu cầu:

- **Thi theo đề ngẫu nhiên:** Ứng dụng cần chọn ngẫu nhiên một tập hợp câu hỏi từ ngân hàng câu hỏi 200 câu hoặc 450 câu của A2. Mỗi bài thi có thể bao gồm số lượng câu hỏi cố định (ví dụ 20-25 câu).
- **Thi theo bộ đề:** Ứng dụng cung cấp các bộ đề cố định dựa trên cấu trúc thi chính thức của kỳ thi GPLX A1, A2. Mỗi bộ đề cần có số câu hỏi và phân loại câu hỏi phù hợp với quy định.
- **Ôn tập 200 câu và 450 câu:** Ứng dụng cho phép người dùng xem toàn bộ danh sách 200 câu hỏi chính thức, bao gồm câu hỏi và đáp án, dưới dạng danh sách hoặc dạng thẻ (flashcard).
- **Ôn thi theo chủ đề:** Ứng dụng phân loại các câu hỏi trong ngân hàng theo các chủ đề như luật giao thông, biển báo đường bộ, xử lý tình huống, v.v. Người dùng có thể chọn ôn tập theo từng chủ đề.
- **Xem lịch sử làm bài:** Ứng dụng lưu trữ lịch sử thi của người dùng, bao gồm các thông tin về số điểm, thời gian làm bài, số câu đúng/sai, và hiển thị khi người dùng cần xem lại.

## 2.2. Thiết kế hệ thống



*Biểu đồ lớp cho mô hình miền*

### Biểu đồ lớp cho MVC:

- Mô tả kiến trúc của ứng dụng, bao gồm các lớp trong Model, View và Controller.
- Thể hiện mối quan hệ giữa các lớp trong MVC và cách chúng tương tác với nhau.
- Ví dụ: trong ứng dụng, biểu đồ lớp cho MVC sẽ chứa các lớp như Task, TaskList, TaskView, TaskController, FileHandler (hoặc GoogleSheetHandler).

### 2.2.1.Thiết kế kiến trúc (MVC)

#### ❖ Model

- **Exam:** thiết kế để đại diện cho một kỳ thi hoặc bài kiểm tra trong ứng dụng.
- **QuestionExam:** đại diện cho một câu hỏi cụ thể trong bài thi (Exam).
- **QuestionSave:** có chức năng lưu lại câu trả lời của người dùng cho một câu hỏi cụ thể trong quá trình làm bài thi.
- **RandomResponse:** thiết kế để đại diện cho một phản hồi hoặc kết quả ngẫu nhiên cho một bài thi cụ thể.
- **SaveAnswer:** lưu trữ câu trả lời của người dùng cho một bài thi cụ thể.
- **SaveAnswerResponse:** thiết kế để đại diện cho phản hồi từ máy chủ sau khi người dùng nộp câu trả lời cho bài thi.
- **AnswerHistory:** thiết kế để lưu lại lịch sử các câu trả lời của người dùng cho một bài thi hoặc câu hỏi cụ thể.
- **History:** thiết kế để lưu trữ thông tin lịch sử của một bài thi đã hoàn thành, bao gồm thông tin chi tiết về bài thi, câu hỏi, câu trả lời, điểm số, và kết quả đạt hay không đạt.
- **QuestionHistory:** thiết kế để đại diện cho một câu hỏi trong lịch sử của một bài thi đã hoàn thành.
- **AnswerQuestionHistory:** thiết kế để lưu trữ các câu trả lời của người dùng cho một câu hỏi trong bài thi, cùng với thông tin về việc mỗi câu trả lời là đúng hay sai.
- **Answers:** thiết kế để lưu trữ một tập hợp các câu trả lời dưới dạng cặp key-value.
- **Option:** thiết kế để lưu trữ các lựa chọn trả lời và các thông tin bổ sung liên quan đến một câu hỏi trắc nghiệm.
- **Question:** thiết kế để đại diện cho một câu hỏi trắc nghiệm trong hệ thống, với các thuộc tính mô tả chi tiết nội dung câu hỏi, các lựa chọn, đáp án đúng, và phân loại.

#### ❖ View

- **Main:**
  - **MainActivity:** Hiển thị giao diện trang chủ của app
  - **ActivityA1:** Hiển thị giao diện trang chủ của A1
  - **ActivityA2:** Hiển thị giao diện trang chủ của A2

- **Content:**
  - **ContentActivity:** Hiển thị giao diện Chủ đề
  - **DetailQuestionActivity:** Hiển thị giao diện chọn câu hỏi
  - **ListContentActivity:** Hiển thị cái list câu hỏi
- **Random**
  - **RandomExamActivity:** giao diện của bộ đề ngẫu nhiên
- **Exam**
  - **ExamActivity:** Hiển thị giao diện bộ đề
  - **DetailQuestionExamActivity:** Hiển thị giao diện câu hỏi của bộ đề
  - **ExamTypeActivity:** Hiển thị các danh sách câu hỏi
- ❖ **Controller**
  - **ExamController:** Xử lý dữ liệu từ Api trả về
  - **HistoryController:** Xử lý dữ liệu từ Api trả về
  - **QuestionController:** Xử lý dữ liệu từ Api trả về

### 2.2.2. Dựa trên kiến trúc MVC đã chọn, ta có thể xác định các lớp sau

- ❖ **Lớp Exam**
  - Thuộc tính: ID, User\_id, List<QuestionExam> questions, exam\_type, is\_fixed, set\_fixed\_number
  - Phương thức: Getters và setters cho các thuộc tính.
- ❖ **Lớp QuestionExam :**
  - Thuộc tính: question\_id
  - Phương thức: Getters và setters cho các thuộc tính.
- ❖ **Lớp QuestionSave :**
  - Thuộc tính: question\_id, user\_answer
  - Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp RandomResponse :**

- Thuộc tính: exam\_id
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp SaveAnswer:**

- Thuộc tính: user\_id = "", exam\_id = "", List<QuestionSave> answer = new ArrayList<>()
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp SaveAnswerResponse:**

- Thuộc tính: message, score, pass
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp AnswerHistory:**

- Thuộc tính: answer
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp History:**

- Thuộc tính: id, exam\_id, user\_id, List<QuestionHistory> questions, List<AnswerQuestionHistory> answers, score, pass
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp QuestionHistory:**

- Thuộc tính: id
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp AnswerQuestionHistory:**

- Thuộc tính: answer
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp Answer:**

- Thuộc tính: answer
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp Options:**

- Thuộc tính: a,b,c,d,image,description
- Phương thức: Getters và setters cho các thuộc tính.

❖ **Lớp Question:**

- Thuộc tính: id,content,options,correct\_answer,category
- Phương thức: Getters và setters cho các thuộc tính.

### 2.2.3. Mối quan hệ giữa các lớp

#### 2.2.3.1.Mối quan hệ giữa các lớp Model

❖ **Exam và QuestionExam**

- Exam chứa một danh sách các QuestionExam thông qua thuộc tính  
List<QuestionExam> questions.
- Mối quan hệ: Composition - Exam là thực thể chứa các QuestionExam để tạo thành một bài thi.

❖ **QuestionExam và Question**

- QuestionExam có thể liên kết với Question để lấy thông tin chi tiết của câu hỏi.
- Mối quan hệ: Association - QuestionExam sử dụng Question để truy xuất nội dung câu hỏi và các lựa chọn.

❖ **SaveAnswer và QuestionSave:**

- SaveAnswer chứa một danh sách các QuestionSave thông qua thuộc tính  
List<QuestionSave> answer.
- Mối quan hệ: Composition - SaveAnswer bao gồm nhiều QuestionSave để lưu trữ câu trả lời của người dùng cho từng câu hỏi trong bài thi.

❖ **SaveAnswerResponse và SaveAnswer**

- SaveAnswerResponse có thể tham chiếu tới SaveAnswer để cung cấp phản hồi về kết quả bài làm của người dùng.
- Mối quan hệ: Association - SaveAnswerResponse cung cấp phản hồi cho các câu trả lời đã được lưu trong SaveAnswer.

❖ **History và QuestionHistory, AnswerQuestionHistory**



- History chứa danh sách các QuestionHistory và AnswerQuestionHistory, lưu lại chi tiết các câu hỏi và câu trả lời trong lịch sử bài thi.
- Mỗi quan hệ: Composition - History lưu trữ toàn bộ thông tin của một bài thi đã hoàn thành, bao gồm các câu hỏi (QuestionHistory) và các câu trả lời (AnswerQuestionHistory).

#### ❖ **QuestionHistory và AnswerQuestionHistory**

- QuestionHistory có thể liên kết với AnswerQuestionHistory để lưu trữ câu trả lời cho từng câu hỏi trong lịch sử bài thi.
- Mỗi quan hệ: Association - QuestionHistory liên kết với AnswerQuestionHistory để ghi lại chi tiết câu trả lời của người dùng cho từng câu hỏi.

#### ❖ **AnswerQuestionHistory và Answer**

- AnswerQuestionHistory chứa các Answer để lưu trữ câu trả lời của người dùng, cùng với thông tin đúng hoặc sai.
- Mỗi quan hệ: Composition - AnswerQuestionHistory bao gồm các Answer cho từng câu trả lời của người dùng.

#### ❖ **Question và Option**

- Question chứa một Option để lưu các lựa chọn đáp án của câu hỏi.
- Mỗi quan hệ: Composition - Question bao gồm Option để cung cấp thông tin về các đáp án có thể có.

### 2.2.3.2. Mỗi quan hệ giữa các lớp Controller và Model

#### ❖ **ExamController và Exam**

- ExamController tương tác với Exam để lấy và xử lý dữ liệu từ API, sau đó trả về cho View.
- Mỗi quan hệ: Association - ExamController sử dụng Exam để thao tác với dữ liệu bài thi.

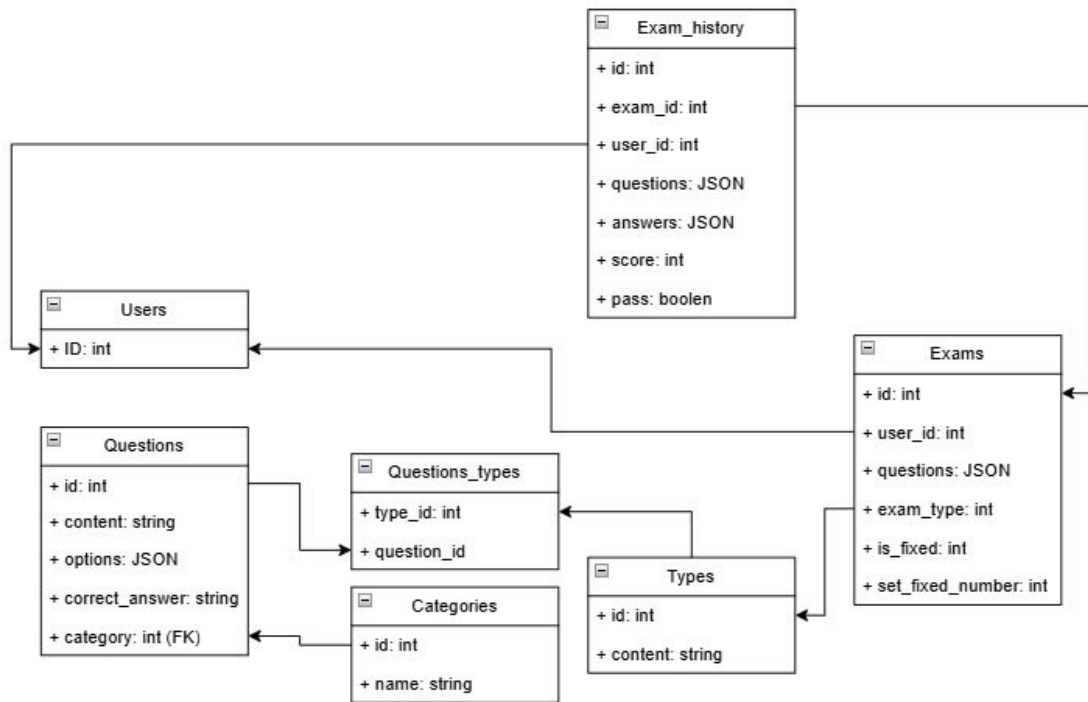
#### ❖ **HistoryController và History**

- HistoryController làm việc với History để lấy thông tin lịch sử các bài thi từ API.

- Mối quan hệ: Association - HistoryController sử dụng History để lấy và xử lý thông tin lịch sử bài thi.
- ❖ **QuestionController và Question**
  - QuestionController tương tác với Question để lấy và xử lý dữ liệu câu hỏi từ API.
  - Mối quan hệ: Association - QuestionController sử dụng Question để thao tác và trả dữ liệu về cho View.

### 2.2.3.3. Mối quan hệ giữa các lớp View và Controller

- ❖ **ExamActivity và ExamController**
  - ExamActivity sử dụng ExamController để gọi các API và xử lý dữ liệu bài thi.
  - Mối quan hệ: Association - ExamActivity sử dụng ExamController để lấy và hiển thị dữ liệu về các bài thi.
  - **ContentActivity, DetailQuestionActivity, ListContentActivity**
- ❖ **QuestionController**
  - Các View này sử dụng QuestionController để lấy dữ liệu câu hỏi và hiển thị lên giao diện.
  - Mối quan hệ: Association - Các lớp View tương tác với QuestionController để lấy dữ liệu câu hỏi từ hệ thống.
- ❖ **MainActivity, ActivityA1, ActivityA2**
  - Các lớp này không tương tác trực tiếp với dữ liệu Model, nhưng có thể chuyển hướng đến các Activity khác để hiển thị dữ liệu.
  - Mối quan hệ: Association - Các Activity chính chuyển người dùng tới các View chi tiết khác trong ứng dụng.



*Biểu đồ lớp cho dự án*

## Thiết kế cơ sở dữ liệu

- Tạo cơ sở dữ liệu mới bằng pgAdmin
- Tạo cơ sở dữ liệu Driver\_License
- Tạo 7 bảng tương ứng với biểu đồ lớp

### 1. Tạo bảng user

```
CREATE TABLE users (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY  
);
```

### 2. Tạo bảng exams

```
CREATE TABLE exams (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    user_id BIGINT UNSIGNED,  
    questions JSON,  
    exam_type INT,  
    is_fixed INT,  
    set_fixed_number INT,  
    created_at TIMESTAMP NULL DEFAULT NULL,  
    updated_at TIMESTAMP NULL DEFAULT NULL,  
    CONSTRAINT fk_user_id FOREIGN KEY (user_id) REFERENCES  
    users(id) ON DELETE CASCADE  
);
```

### 3. Tạo bảng categories

```
CREATE TABLE categories (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255),  
    created_at TIMESTAMP NULL DEFAULT NULL,  
    updated_at TIMESTAMP NULL DEFAULT NULL  
);
```

#### 4. Tạo bảng types

```
CREATE TABLE types (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    content VARCHAR(255),  
    created_at TIMESTAMP NULL DEFAULT NULL,  
    updated_at TIMESTAMP NULL DEFAULT NULL  
);
```

#### 5. Tạo bảng exam\_histories

```
CREATE TABLE exam_histories (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    exam_id BIGINT UNSIGNED NOT NULL,  
    user_id BIGINT UNSIGNED NOT NULL,  
    questions JSON,  
    answers JSON,  
    score INT,  
    pass BOOLEAN,  
    created_at TIMESTAMP NULL DEFAULT NULL,  
    updated_at TIMESTAMP NULL DEFAULT NULL,  
    FOREIGN KEY (exam_id) REFERENCES exams(id),  
    FOREIGN KEY (user_id) REFERENCES users(id)  
);
```

#### 6. Tạo bảng questions

```
CREATE TABLE questions (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    content TEXT,  
    options JSON,  
    correct_answer VARCHAR(255),  
    category BIGINT UNSIGNED NOT NULL,  
    created_at TIMESTAMP NULL DEFAULT NULL,
```

```

        updated_at TIMESTAMP NULL DEFAULT NULL,
        FOREIGN KEY (category) REFERENCES categories(id)
    );

```

## 7. Tạo bảng question\_types

```

CREATE TABLE question_types (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    type_id BIGINT UNSIGNED NOT NULL,
    question_id BIGINT UNSIGNED NOT NULL,
    created_at TIMESTAMP NULL DEFAULT NULL,
    updated_at TIMESTAMP NULL DEFAULT NULL,
    FOREIGN KEY (type_id) REFERENCES types(id),
    FOREIGN KEY (question_id) REFERENCES questions(id)
);

```

### 2.3. Triển khai

- ❖ **Viết code: Sử dụng Java để cài đặt các class trong mô hình MVC, đọc/ghi file, xử lý dữ liệu và hiển thị giao diện**
  - Cài đặt Glide để xử lý hình ảnh qua url.
  - Tích hợp API của bên thứ ba: Sử dụng Retrofit để gọi các API từ bên thứ ba nhằm hiển thị dữ liệu. Dữ liệu trả về được xử lý và hiển thị trong các thành phần giao diện.
  - **Tích hợp OkHttp** xử lý việc kết nối, quản lý các phiên kết nối, và cung cấp các tính năng như caching, gzip compression, và automatic redirection.
  - **HttpLoggingInterceptor**: là một interceptor cụ thể trong OkHttp giúp bạn ghi lại các thông tin chi tiết về yêu cầu và phản hồi HTTP. Bạn có thể tùy chỉnh mức độ ghi lại, ví dụ: chỉ ghi lại URL, headers, hoặc toàn bộ body của yêu cầu và phản hồi.
- ❖ **Kiểm thử: Viết các test case để kiểm tra các chức năng của ứng dụng, đảm bảo ứng dụng hoạt động đúng theo yêu cầu.**

## 2.4. Vận hành và bảo trì

### ❖ Cài đặt và triển khai:

- Hướng dẫn người dùng cách cài đặt và chạy ứng dụng trên thiết bị Android bằng cách tải về file APK từ Google Play hoặc từ nguồn phân phối khác.
- Cài đặt thư viện: Khi tải ứng dụng từ nguồn ngoài, người dùng cần cài đặt các thư viện cần thiết, như Firebase SDK, bằng cách cập nhật Gradle (Android Studio sẽ tự động quản lý các thư viện khi triển khai trên Google Play).

### ❖ Bảo trì

- **Sửa lỗi phát sinh:** Khắc phục các lỗi phát sinh trong quá trình sử dụng, bao gồm các lỗi về giao diện, lỗi kết nối.
- **Cập nhật tính năng mới:** Khi có yêu cầu hoặc phản hồi từ người dùng, cập nhật thêm các tính năng, luôn lắng nghe người dùng.
- **Cải thiện hiệu năng:** Tối ưu hóa ứng dụng để giảm thời gian tải và cải thiện hiệu suất khi xử lý dữ liệu lớn, đảm bảo ứng dụng chạy mượt mà.

## CHƯƠNG III. KẾT QUẢ THỰC HIỆN

### 3.1. Công nghệ đã sử dụng

- **Ngôn ngữ lập trình: Java**
- **Công cụ: Android Studio**
- **Thư viện**
  - **Glide** : để quản lý cơ sở dữ liệu người dùng, quản lý tài khoản và các dịch vụ xác thực.
  - **Retrofit**: để thực hiện các kết nối API cho việc truy xuất dữ liệu phim.
  - **Các thư viện hỗ trợ khác**: thư viện đọc/ghi file, xử lý dữ liệu JSON.

### 3.2. Tiến độ thực hiện

Link github tới dự án: [CSE441 Project: Nguyễn Xuân Tùng Anh, Trần Hiếu, Đặng Việt Anh, Đỗ Thị Trang](#)

#### Hướng dẫn các bước đã thực hiện:

##### B1. Tạo dự án mới:

- Mở Android Studio.
- Chọn "Create New Project".
- Chọn "Java" làm ngôn ngữ lập trình.
- Chọn JDK-23
- Nhập tên dự án: Driver'sLicense
- Lưu vào ổ D/Android
- Nhấn "Finish" để tạo dự án.



## **B2. Tạo các package:**

- Trong cửa sổ "Project", click chuột phải vào thư mục "src".
- Chọn "New" -> "Package".
- Tạo các package sau:
  - model
  - view
  - controller
  - adapter
  - DataLocal
  - Api

## **B3. Tạo các lớp:**

- ❖ Trong mỗi package, click chuột phải và chọn "New" -> "Java Class" để tạo các lớp tương ứng
- ❖ **model:** Trong model tạo các package exam, history, question.
  - exam chứa các lớp: Exam, QuestionExam, QuestionSave, RandomResponse, SaveAnswer, SaveAnswerResponse.
  - history chứa các lớp: AnswerHistory, History, QuestionHistory
  - question chứa các lớp: AnswerQuestionHistory, Answers, Option, Question

❖ **view**: Trong view tạo các package content, exam, history, main, question, random.

- content chứa các lớp: ContentActivity, DetailQuestionActivity, ListContentActivity.
- exam chứa các lớp: ExamActivity, DetailQuestionExamActivity, ExamTypeActivity.
- history chứa các lớp: HistoryActivity.
- main chứa các lớp: ActivityA1, ActivityA2, MainActivity.
- question chứa các lớp: QuestionActivity
- random chứa các lớp: RandomExamActivity

❖ **controller** chứa các lớp:

- ExamController
- HistoryController
- QuestionController

❖ **adapter** chứa các lớp:

- examAdapter
- ExamTypeAdapter
- HistoryAdapter
- ListContentAdapter
- ListQuestionAdapter

- RandomAdapter

#### ❖ **DataLocal**

- DataMemory

#### ❖ **Api** chứa các lớp:

- ApiService

### **B4.Tạo giao diện người dùng:**

#### **1.Thiết kế layout:**

❖ Trong thư mục res/layout, tạo các file XML để thiết kế giao diện cho các Activity.

- activity\_a1: hiện giao diện chính của phần A1
- activity\_a2: hiện giao diện chính của phần A2
- activity\_content: Hiện giao diện của phần chủ đề
- activity\_detail\_question: hiện chi tiết các câu hỏi
- activity\_exam: Hiện danh sách đề theo từng bộ A1 hoặc A2
- activity\_exam\_type: Hiện phần làm bài theo từng đề
- activity\_history: hiển thị lịch sử bài làm
- activity\_list\_content: hiển thị danh sách các chủ đề
- activity\_main: giao diện màn hình chính
- activity\_question: hiển thị ra danh sách các câu hỏi theo từng bộ đề A1 hoặc A2
- activity\_random\_exam: Hiển thị lên các câu hỏi trong bộ đề random

## **2.Thiết lập giao diện**

- Mở từng file XML vừa tạo và sử dụng các thành phần như TextView, Button, LinearLayout, Table, ListView, ImageView, để thiết kế giao diện.
- Điều chỉnh các thuộc tính như textSize, textStyle, layout\_width, layout\_height, paddingStart, textColor, gravity, v.v., để tạo giao diện thân thiện và dễ dùng.

## **3.Kết nối giao diện với activity**

- Trong từng lớp Activity, sử dụng findViewById để kết nối các thành phần trong file XML với các biến trong code, để có thể xử lý sự kiện cho các nút và hiển thị dữ liệu.

## **B5. Thiết lập ListView cho danh sách:**

### **1.Tạo và cấu hình các Adapter cho ListView:**

- Trong thư mục adapter cài đặt các lớp Adapter: ExamAdapter, ExamTypeAdapter, HistoryAdapter, ListContentAdapter, ListQuestionAdapter, RandomAdapter.
- Những Adapter này đều kế thừa từ ArrayAdapter và được cấu hình để hiển thị các mục khác nhau trong danh sách.

### **2.Thiết lập các ListView trong Activity:**

- Trong các Activity tương ứng (chẳng hạn ExamActivity, HistoryActivity, ListContentActivity), thêm ListView vào file XML của layout, với ID riêng biệt cho từng mục đích hiển thị khác nhau (ví dụ: @id/list\_exam, @id/list\_history, @id/list\_content).
- Trong mỗi Activity, tìm ListView bằng findViewById và thiết lập Adapter tương ứng cho từng ListView bằng phương thức setAdapter():
- ExamActivity sẽ sử dụng ExamAdapter.

- HistoryActivity sẽ sử dụng HistoryAdapter.
- ListContentActivity và các hoạt động liên quan đến câu hỏi sẽ sử dụng ListContentAdapter hoặc ListQuestionAdapter.

### **3. Đổ dữ liệu vào Adapter :**

- Sử dụng các lớp Model tương ứng (Exam, History, Question) để đổ dữ liệu vào Adapter khi khởi tạo.

## **B6. Tích hợp API với Retrofit và sử dụng trong ứng dụng:**

### **1. Cấu hình Retrofit:**

- Cấu hình Retrofit để kết nối với API . Thêm thư viện Retrofit vào file build.gradle

### **2. Định nghĩa Interface ApiService:**

- Sử dụng ApiService đã định nghĩa để chứa các phương thức API, ví dụ:  
getExamsData(int examType): Lấy danh sách các bài thi dựa trên examType.  
getUserExamHistory(int typeId): Lấy lịch sử thi của người dùng dựa trên typeId.  
saveAnswer(SaveAnswer saveAnswer): Gửi dữ liệu câu trả lời người dùng để lưu.

### **3. Gọi API trong các Activity**

- Trong mỗi Activity, sử dụng Retrofit để gọi các API từ ApiService.

### **4. Xử lý dữ liệu API và cập nhật giao diện:**

- Sau khi nhận được dữ liệu từ API, cập nhật Adapter của ListView tương ứng

### **5. Xử lý dữ liệu lỗi hoặc thông báo đến người dùng:**

- Đảm bảo xử lý các trường hợp lỗi như lỗi kết nối mạng, lỗi phản hồi không

thành công và hiển thị thông báo cho người dùng biết.

### **B7. Cài đặt DataLocal để lưu dữ liệu cục bộ:**

- Trong DataLocal/DataMemory, sử dụng SharedPreferences để lưu các thông tin cơ bản, như kết quả thi của người dùng, các câu hỏi đã trả lời, hoặc trạng thái đăng nhập.

### **B8. Xử lý sự kiện và logic trong Controller**

#### **1. Xử lý dữ liệu từ API:**

- Controller sử dụng các dịch vụ (services) như ApiService để thực hiện các yêu cầu mạng đến API và lấy dữ liệu. Ví dụ, ExamController, HistoryController, và QuestionController đều là các Controller chuyên xử lý việc chuyển đổi dữ liệu JSON nhận được từ API thành các đối tượng Java.

#### **2. Chuyển đổi dữ liệu:**

- Mỗi Controller (như ExamController, HistoryController, QuestionController) thực hiện việc deserialization của dữ liệu JSON, tức là biến đổi dữ liệu từ định dạng JSON thành các đối tượng Java có thể sử dụng trong ứng dụng. Việc này giúp tổ chức dữ liệu một cách dễ dàng và có thể truy cập.

#### **3.Quản lý logic ứng dụng:**

- Controller xử lý các logic liên quan đến việc tương tác với dữ liệu.

#### **4.Giao tiếp với các Activity**

- Controller nhận các sự kiện từ người dùng qua các Activity (hoặc Fragment) và dựa vào đó để gọi các phương thức thích hợp từ các dịch vụ.3.3.

Hình ảnh sản phẩm

### **B9. Kiểm tra và sửa lỗi**

#### **1. Kiểm tra chức năng:**

- Chạy ứng dụng trên máy ảo hoặc thiết bị thật để kiểm tra các chức năng như lấy đề thi ngẫu nhiên, thi theo bộ đề, danh sách các câu hỏi, chủ đề các câu hỏi.

## **2. Sửa lỗi (Debugging):**

- Kiểm tra logcat để phát hiện lỗi (nếu có) và sửa lỗi liên quan đến các vấn đề như API, ListView, hoặc xử lý dữ liệu.

## **B10. Hoàn thiện và tối ưu hóa**

### **1. Tối ưu hóa hiệu suất:**

- Tối ưu mã nguồn, loại bỏ các dòng code thừa, và đảm bảo rằng ứng dụng hoạt động mượt mà.

### **2. Thiết kế UI thân thiện:**

- Điều chỉnh UI sao cho phù hợp với các kích thước màn hình khác nhau và dễ sử dụng.

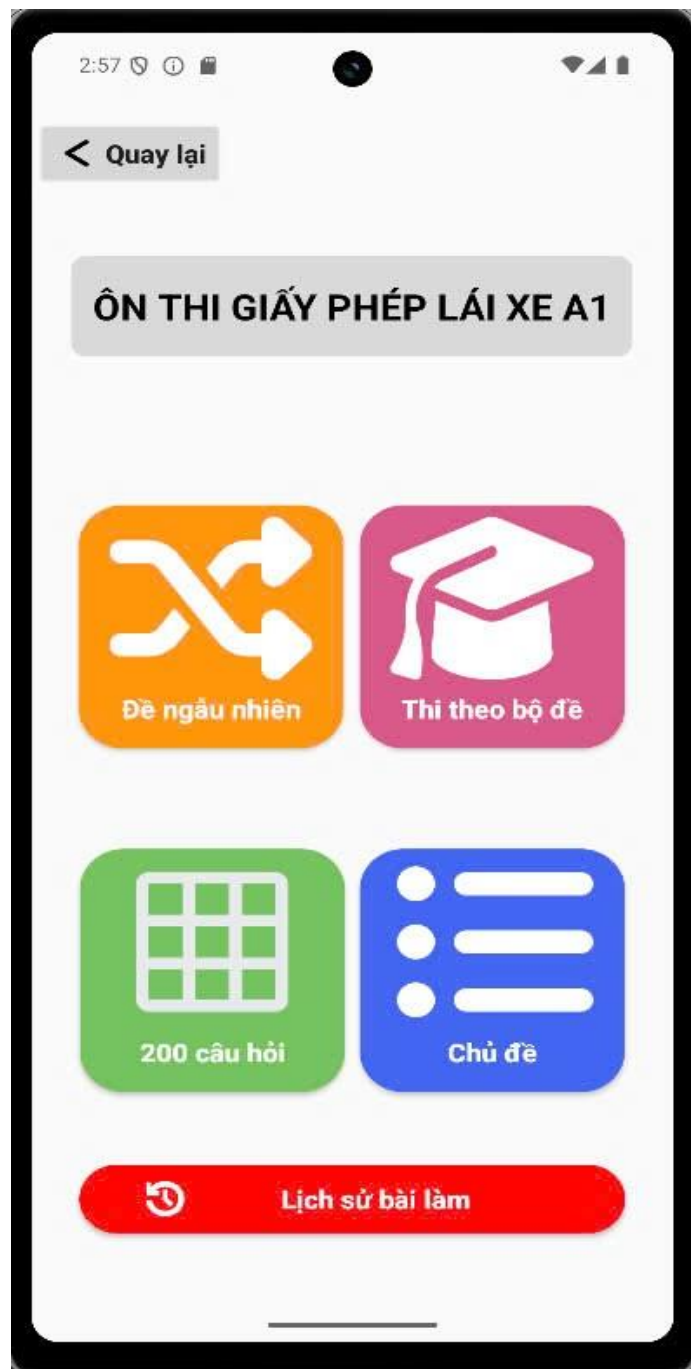
### 3.3.Hình ảnh sản phẩm

#### 3.3.1.Giao diện chung

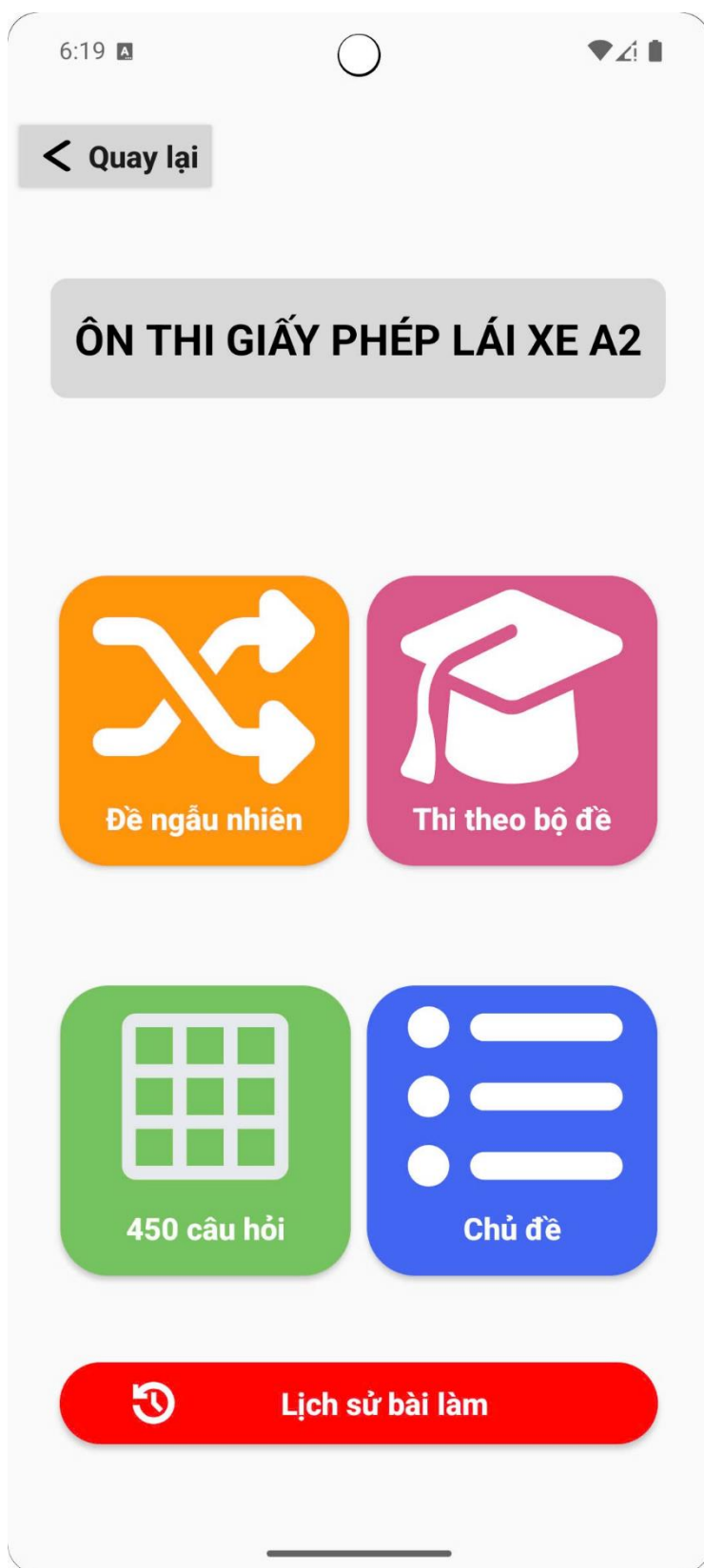


Hình 1. Giao diện trang chủ





Hình 2. Giao diện A1

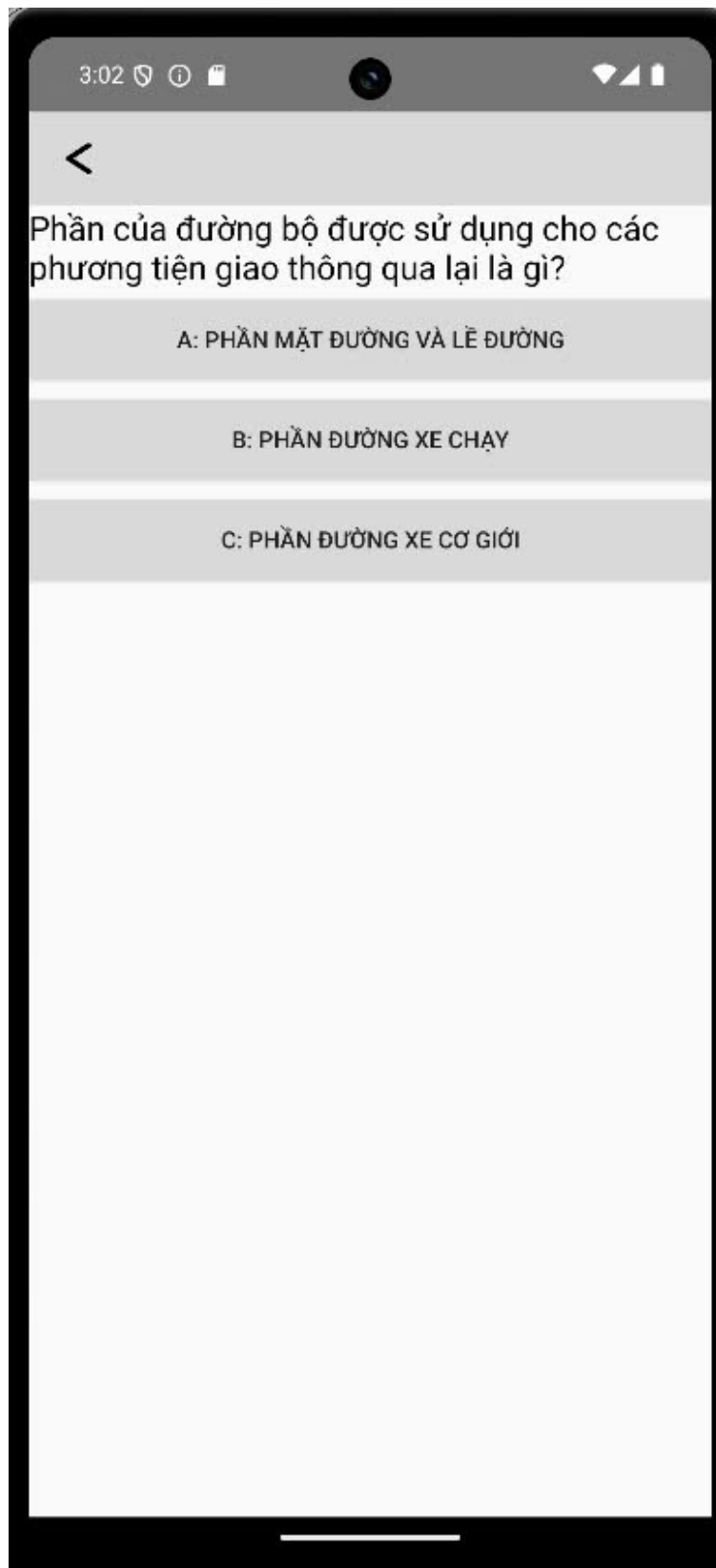


Hình 3. Giao diện A2

### 3.3.2. Chức năng đề ngẫu nhiên



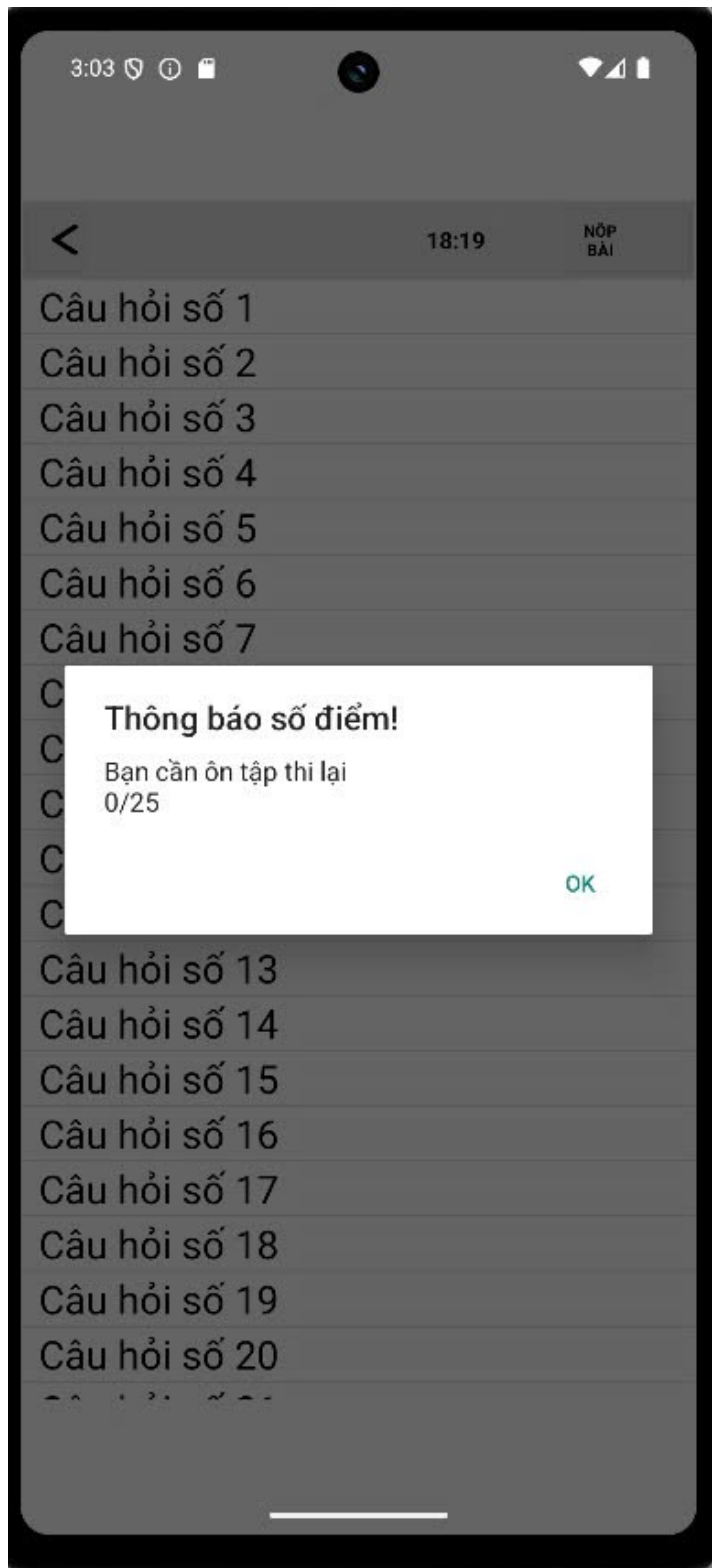
Hình 4. Giao diện chức năng đề ngẫu nhiên



Hình 5. Giao diện câu hỏi



Hình 6. Giao diện nộp bài

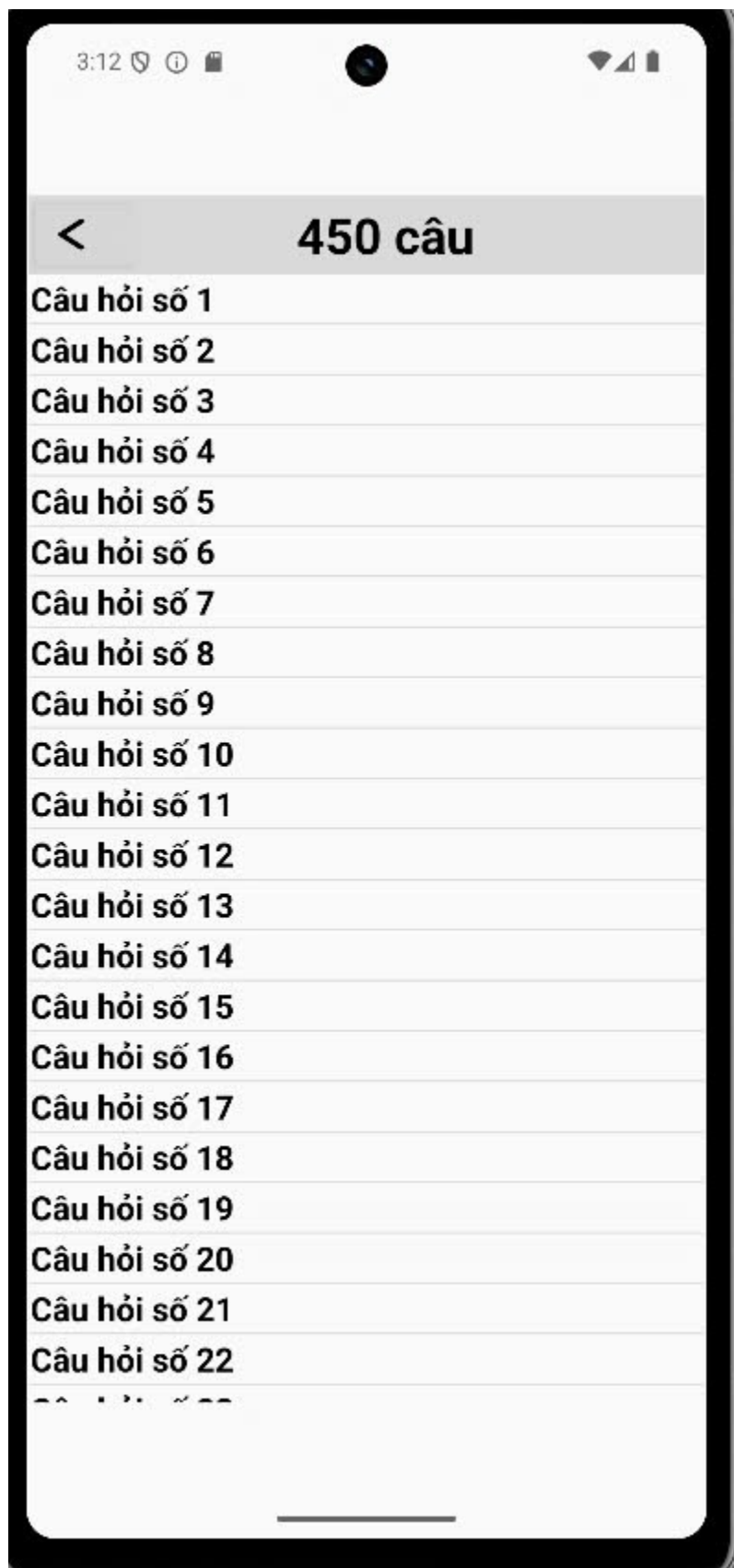


Hình 7. Thông báo nộp bài thành công

### 3.3.3. Chức năng ôn tập theo câu hỏi



Hình 8. Giao diện ôn tập các câu hỏi GPLX A1



Hình 9. Giao diện ôn tập các câu hỏi GPLX A2





Hình 10. Giao diện câu hỏi

Khi trả lời đúng, câu trả lời sẽ chuyển qua màu xanh và hiện ra hướng dẫn.  
Khi trả lời sai, câu trả lời sẽ chuyển đỏ và hiện ra hướng dẫn

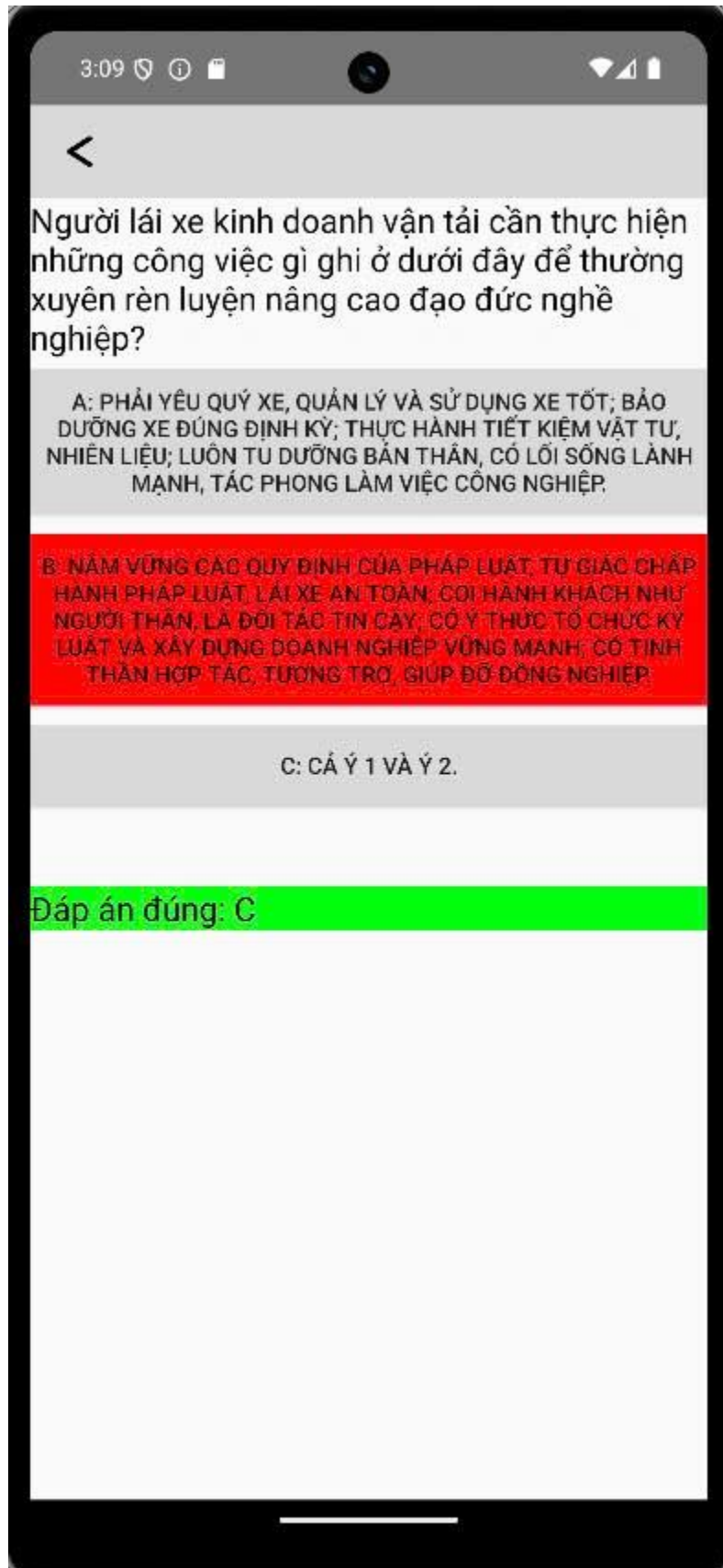
### 3.3.4. Chức năng ôn tập theo chủ đề, nhóm câu hỏi



Hình 11. Giao diện chức năng ôn theo chủ đề



Hình 12. Giao diện ôn theo chủ đề văn hoá và đạo đức

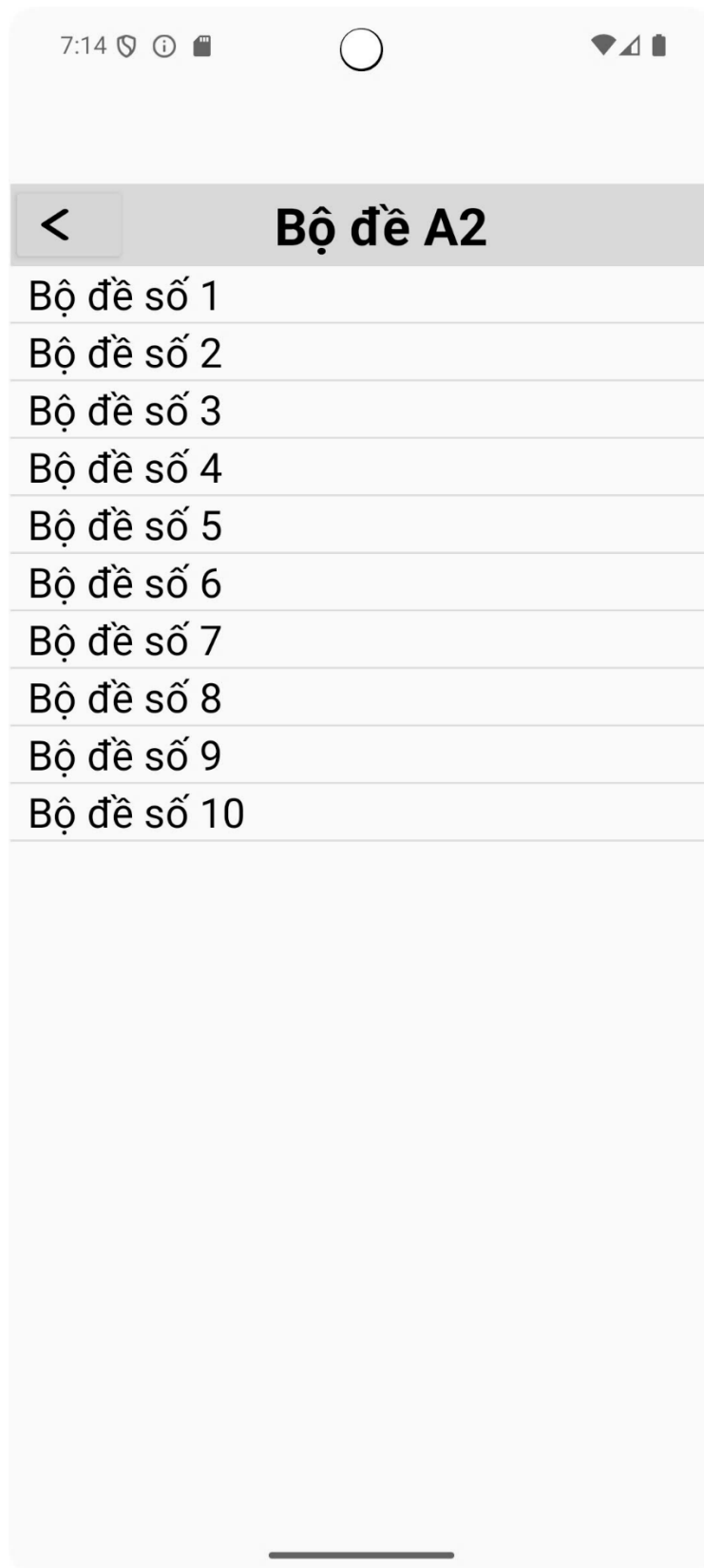


Hình 13. Giao diện câu hỏi

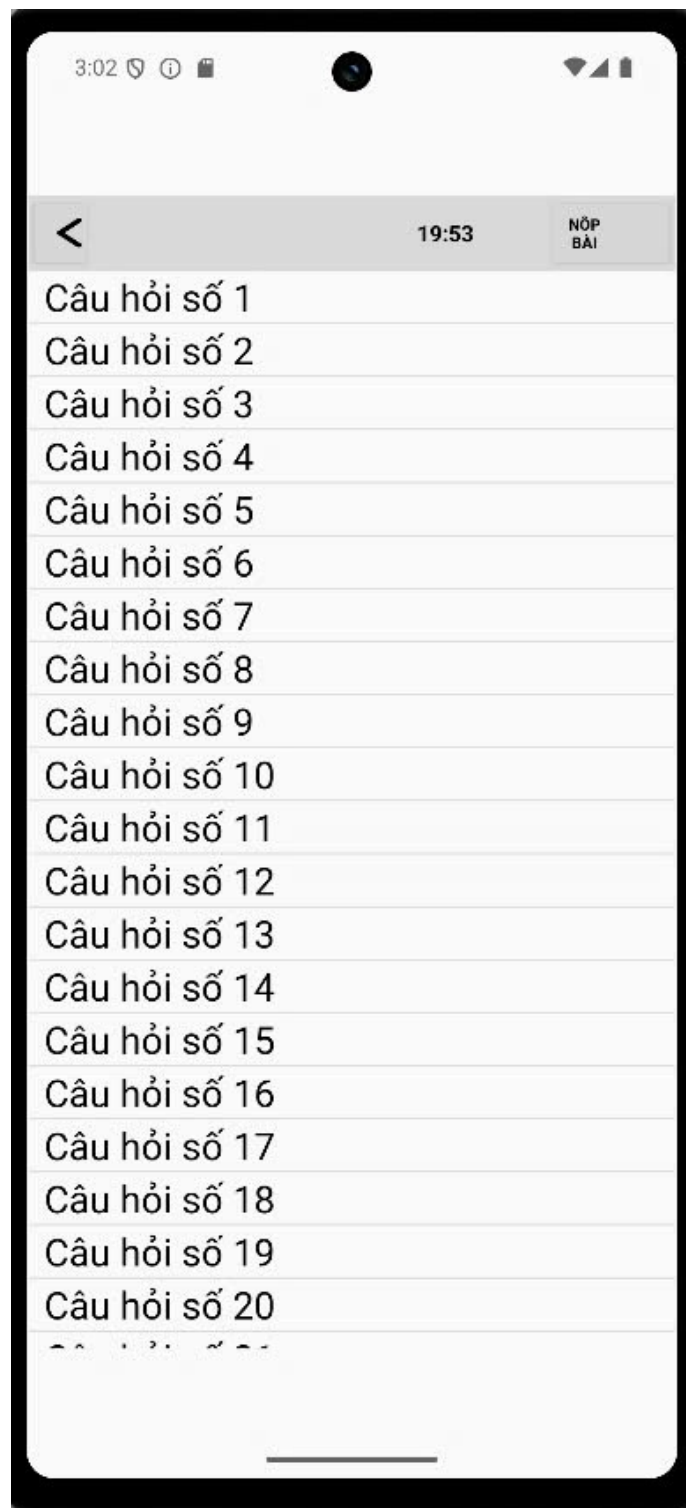
### 3.3.5. Chức năng thi theo bộ đề



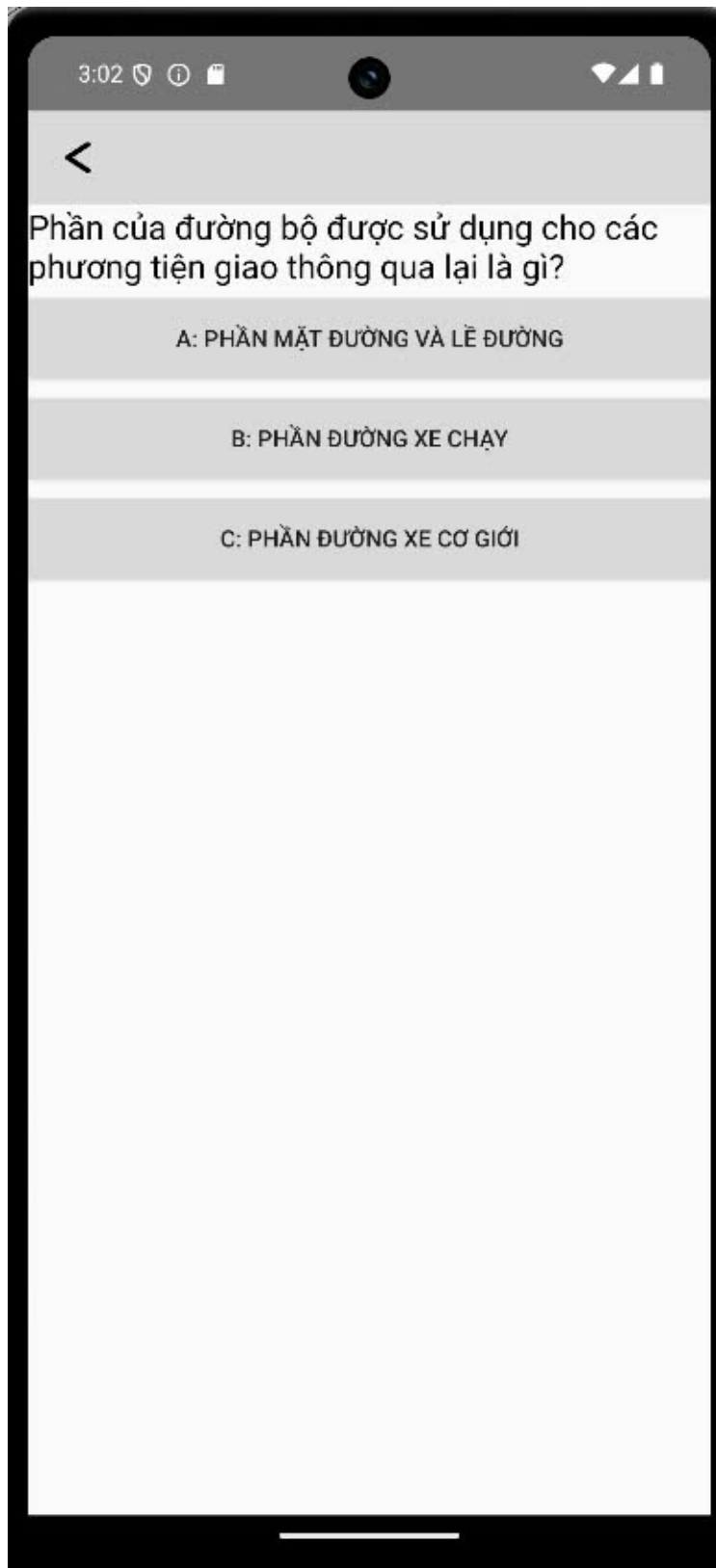
Hình 14. Chức năng thi theo bộ đề GPLX A1



Hình 15. Chức năng thi theo bộ đề GPLX A2



Hình 16. Giao diện làm bài của bộ đề

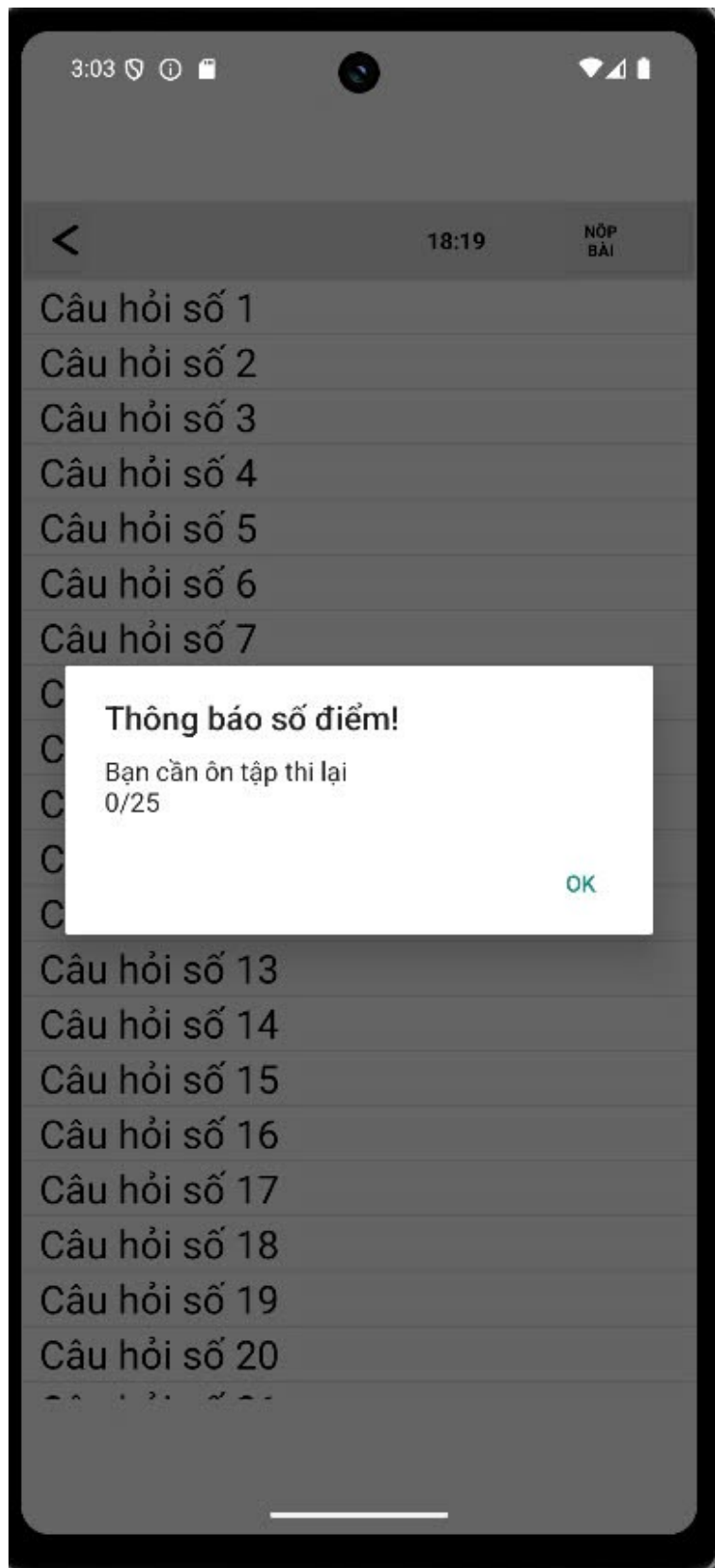


*Hình 17. Giao diện câu hỏi*





Hình 18. Giao diện xác nhận nộp bài



Hình 19. Giao diện nộp bài thành công

### 3.3.6. Chức năng lịch sử



The screenshot shows a mobile application interface with a status bar at the top displaying the time 3:04 and various icons. Below the status bar is a header bar with a back arrow and the title 'Lịch sử bài làm'. The main content area contains a table with three columns: 'Bài số' (Article Number), 'Điểm số' (Score), and 'Kết quả' (Result). The table lists six entries, all of which show a score of 0 and a result of 'Trượt' (Failed). The last entry has a score of 19, which appears to be a discrepancy with the other entries.

Bài số	Điểm số	Kết quả
Bài số: 1	Điểm số: 0	Kết quả: Trượt
Bài số: 2	Điểm số: 2	Kết quả: Trượt
Bài số: 6	Điểm số: 0	Kết quả: Trượt
Bài số: 7	Điểm số: 1	Kết quả: Trượt
Bài số: 8	Điểm số: 0	Kết quả: Trượt
Bài số: 19	Điểm số: 19	Kết quả: Trượt

Hình 20. Chức năng lịch sử

## **KẾT LUẬN**

### **1. Kết quả đạt được**

Sau thời gian thực hiện nghiên cứu, chúng em đã hoàn thành được những mục tiêu được đề ra như sau:

- Nghiên cứu phương pháp và các vấn đề xây dựng một phần mềm ứng dụng Ôn thi lý thuyết bằng lái xe máy
- Khảo sát và tìm hiểu những đặc trưng của quản lý và kinh doanh trong lĩnh vực giáo dục
- Phát triển ứng dụng cho các thiết bị di động

### **2. Hướng phát triển**

Sau thời gian làm bài tập lớn, mặc dù nhóm đã nỗ lực làm việc và nghiên cứu nhưng vì thời gian, kiến thức, cũng như kinh nghiệm thực tế mà sản phẩm chưa thể hoàn thành một cách tốt nhất.

Chúng em sẽ cố gắng, tiếp tục nghiên cứu để sản phẩm được hoàn thành tốt hơn trong tương lai.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. **Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.**
- [2]. **Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.**
- [3]. **James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.**
- [4]. **Raymond Gallardo, Scott Hommel, Sowmya Kannan, Joni Gordon, Sharon Biocca Zakhour. The Java Tutorial: A Short Course on the Basics, 6th edition.**
- [5]. **Dave MacLean, Satya Komatineni, Grant Allen. Pro Android 5.**
- [6]. **Lê Hoàng Sơn, Nguyễn Thọ Thông. NXB: Xây dựng. Giáo trình Lập trình Android: Giáo trình dành cho bậc đại học ngành công nghệ thông tin.**