

**BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG ỨNG DỤNG ÔN THI LÝ THUYẾT BẰNG LÁI XE  
SỬ DỤNG ANDROID STUDIO**

**CBHD: ThS. Nguyễn Thị Cẩm Ngoan**

**Sinh viên: Đặng Xuân Hoàng**

**Mã số sinh viên: 2020605812**

**ĐẶNG XUÂN HOÀNG**

**CÔNG NGHỆ THÔNG TIN**

**Hà Nội – Năm 2024**

## LỜI NÓI ĐẦU

Trong thời đại hiện đại, việc sở hữu một phương tiện di chuyển như xe máy hoặc ô tô là một yếu tố không thể thiếu. Việc tham gia và điều khiển phương tiện giao thông đã trở thành một phần không thể tách rời trong cuộc sống hàng ngày của mỗi người. Chính vì vậy việc sở hữu một tấm bằng lái xe là vô cùng cần thiết. Hiểu rõ nhu cầu ấy của mọi người, em đã quyết định xây dựng ứng dụng ôn thi lý thuyết bằng lái xe dành cho tất cả mọi người.

Với niềm đam mê lập trình, tìm hiểu về công nghệ mới. Cùng với sự hướng dẫn tận tình của cô **ThS. Nguyễn Thị Cẩm Ngoan**, em đã quyết định chọn đề tài: “Xây dựng ứng dụng ôn thi lý thuyết bằng lái xe sử dụng Android Studio” để giúp cho mọi người có một công cụ để học tập và củng cố kiến thức một cách kỹ lưỡng, dễ dàng và thuận tiện.

Với những kiến thức của bản thân qua quá trình học tập và nghiên cứu, cũng như kinh nghiệm từ thực tế, em mong rằng ứng dụng sẽ đáp ứng được nhu cầu của người dùng và giúp người dùng dễ dàng hơn trong việc ôn thi lý thuyết bằng lái xe.

Đặc biệt, em xin chân thành cảm ơn **ThS. Nguyễn Thị Cẩm Ngoan** đã giúp đỡ và định hướng em trong quá trình hoàn thành đồ án.

Em rất mong nhận được những lời nhận xét, đánh giá của thầy cô để bài báo cáo trở nên hoàn thiện và kỹ năng của bản thân được phát triển xa hơn nữa trong tương lai.

Em xin chân thành cảm ơn!

## MỤC LỤC

<b>LỜI NÓI ĐẦU .....</b>	<b>ii</b>
<b>MỤC LỤC.....</b>	<b>ii</b>
<b>DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT.....</b>	<b>v</b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b>vi</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>ix</b>
<b>MỞ ĐẦU .....</b>	<b>1</b>
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....</b>	<b>3</b>
1.1. Giới thiệu về hệ điều hành Android .....	3
1.1.1. Lịch sử hệ điều hành Android .....	3
1.1.2. Kiến trúc của hệ điều hành Android.....	4
1.1.2.1. Applications .....	5
1.1.2.2. Application Framework .....	5
1.1.2.3. Android Runtime .....	6
1.1.2.4. Native Libraries .....	7
1.1.2.5. Linux Kernel .....	8
1.1.3. Ưu và nhược điểm của hệ điều hành Android.....	8
1.1.3.1. Ưu điểm của hệ điều hành Android .....	8
1.1.3.2. Nhược điểm của hệ điều hành Android .....	9
1.2. Giới thiệu về Android Studio .....	10
1.3. Giới thiệu về ngôn ngữ lập trình Kotlin .....	12
1.3.1. Lịch sử hình thành .....	13
1.3.2. Ứng dụng của Kotlin .....	13
1.4. Room Database.....	14
<b>CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....</b>	<b>16</b>
2.1. Đặt vấn đề, giới thiệu đề tài.....	16
2.2. Các yêu cầu chức năng.....	17
2.3. Tài liệu thu thập được.....	18
2.4. Phân tích thiết kế hệ thống .....	22
2.4.1. Biểu đồ use case tổng quát .....	22
2.4.2. Mô tả và phân tích chi tiết Use case .....	23

2.4.2.1. Use case Học lý thuyết.....	23
2.4.2.2. Use case Chọn hạng bằng thi .....	25
2.4.2.3. Use case Kết thúc thi sát hạch .....	27
2.4.2.4. Use case Xem mẹo thi.....	29
2.4.2.5. Use case Tạo thông báo .....	30
2.4.2.6. Use case Xem câu hỏi hay sai.....	33
2.4.2.7. Use case Học biển báo .....	35
2.4.2.8. Use case Thi sát hạch.....	36
2.4.2.9. Use case Tạo đề thi thử.....	38
2.4.3. Thiết kế cơ sở dữ liệu .....	40
2.4.3.1. Biểu đồ thực thể liên kết.....	40
2.4.3.2. Thiết kế bảng .....	41
<b>CHƯƠNG 3: THIẾT KẾ GIAO DIỆN, CÀI ĐẶT VÀ KIỂM THỬ .....</b>	<b>44</b>
3.1. Thiết kế giao diện .....	44
3.1.1. Màn hình chọn hạng bằng thi .....	44
3.1.2. Màn hình Cài đặt .....	44
3.1.3. Màn hình Học lý thuyết.....	45
3.1.4. Màn hình Xem mẹo thi.....	45
3.1.5. Màn hình Học biển báo .....	46
3.1.6. Màn hình Tạo thông báo.....	46
3.1.7. Màn hình Xem câu hỏi hay sai .....	47
3.1.8. Màn hình Thi sát hạch .....	47
3.1.9. Màn hình Kết thúc thi sát hạch.....	48
3.1.10. Màn hình Kết quả .....	48
3.1.11. Màn hình Tạo đề thi thử .....	49
3.1.12. Màn hình Trang chủ .....	49
3.2. Cài đặt chương trình .....	50
3.2.1. Yêu cầu thiết bị.....	50
3.2.2. Tiến hành cài đặt.....	50
3.3. Kết quả đạt được.....	50
3.3.1. Màn hình Onboarding.....	50

3.3.2. Màn hình Trang chủ .....	51
3.3.3. Màn hình Học lý thuyết .....	51
3.3.4. Màn hình Học biển báo .....	52
3.3.5. Màn hình Mẹo thi .....	52
3.3.6. Màn hình Câu hỏi hay sai .....	53
3.3.7. Màn hình Xem tỉ lệ đỗ .....	53
3.3.8. Màn hình Cài đặt .....	54
3.3.9. Màn hình Thông báo.....	54
3.3.10. Màn hình Thi sát hạch .....	55
3.3.11. Màn hình Tạo đề thi thử .....	55
3.3.12. Màn hình Kết thúc thi sát hạch.....	56
3.3.13. Màn hình Kết quả .....	56
3.4. Kiểm thử .....	57
3.4.1. Kế hoạch kiểm thử.....	57
3.4.1.1. Mục tiêu .....	57
3.4.1.2. Các chức năng được đưa vào kiểm thử.....	57
3.4.1.3. Hướng tiếp cận.....	57
3.4.1.4. Tiêu chí kiểm thử thành công/thất bại .....	58
3.4.2. Kịch bản kiểm thử .....	59
3.4.2.1. Kiểm thử cơ sở dữ liệu.....	59
3.4.2.2. Kiểm thử chức năng.....	59
3.4.3. Kết quả kiểm thử .....	64
3.5. Định hướng phát triển.....	64
<b>KẾT LUẬN VÀ KIẾN NGHỊ .....</b>	<b>66</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>68</b>

## DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

Từ ngữ	Ý nghĩa	Mô tả
API	Application Programming Interface	Giao diện lập trình ứng dụng
DAO	Data Access Object	Là một interface định nghĩa các phương thức để thao tác với dữ liệu trong cơ sở dữ liệu.
IDE	Integrated Development Environment	Môi trường tích hợp dùng để viết code để phát triển ứng dụng
iOS	iPhone Operating System	Hệ điều hành được Apple phát triển độc lập và chỉ sử dụng cho phần cứng của iPhone, iPad, iPod touch
UI	User Interface	Giao diện người dùng
UX	User Experience	Trải nghiệm người dùng
VOPC	View – Operation – Presenter - Controller	Biểu đồ lớp VOPC là một biểu đồ phân loại các thành phần trong kiến trúc phần mềm, giúp tổ chức và hiểu được cách mà các thành phần tương tác với nhau trong ứng dụng.

## DANH MỤC HÌNH ẢNH

Hình 1. 1. Lịch sử phát triển của Android .....	3
Hình 1. 2. Kiến trúc của Android.....	5
Hình 1. 3. Application.....	5
Hình 1. 4. Tầng Application Framework .....	5
Hình 1. 5. Android Runtime.....	7
Hình 1. 6. Native Libraries.....	7
Hình 1. 7. Logo Android Studio .....	10
Hình 1. 8. Thành phần của Room Database.....	14
Hình 2. 1. Câu hỏi khảo sát người dùng 1 .....	19
Hình 2. 2. Câu hỏi khảo sát người dùng 2 .....	19
Hình 2. 3. Câu hỏi khảo sát người dùng 3 .....	20
Hình 2. 4. Câu hỏi khảo sát người dùng 4 .....	20
Hình 2. 5. Câu hỏi khảo sát người dùng 5 .....	20
Hình 2. 6. Câu hỏi khảo sát người dùng 6 .....	21
Hình 2. 7. Câu hỏi khảo sát người dùng 7 .....	21
Hình 2. 8. Câu hỏi khảo sát người dùng 8 .....	21
Hình 2. 9. Biểu đồ use case tổng quát.....	22
Hình 2. 10. Biểu đồ VOPC Use case Học lý thuyết .....	24
Hình 2. 11. Biểu đồ Basic Flow Use case Học lý thuyết.....	25
Hình 2. 12. Biểu đồ VOPC Use case Chọn hạng bằng thi.....	26
Hình 2. 13. Biểu đồ Basic Flow Use case Chọn hạng bằng thi .....	27
Hình 2. 14. Biểu đồ VOPC Use case Kết thúc thi sát hạch .....	28
Hình 2. 15. Biểu đồ Basic Flow Use case Kết thúc thi sát hạch.....	28
Hình 2. 16. Biểu đồ VOPC Use case Xem mẹo thi .....	29
Hình 2. 17. Biểu đồ Basic Flow Use case Xem mẹo thi.....	30
Hình 2. 18. Biểu đồ VOPC Use case Tạo thông báo .....	31
Hình 2. 19. Biểu đồ Basic Flow Use case Tạo thông báo.....	32
Hình 2. 20. Biểu đồ VOPC Use case Xem câu hỏi hay sai .....	34
Hình 2. 21. Biểu đồ Basic Flow Use case Xem câu hỏi hay sai .....	34

Hình 2. 22. Biểu đồ VOPC Use case Học biển báo.....	35
Hình 2. 23. Biểu đồ Basic Flow Use case Học biển báo .....	36
Hình 2. 24. Biểu đồ VOPC Use case Thi sát hạch.....	37
Hình 2. 25. Biểu đồ Basic Flow Use case Thi sát hạch .....	38
Hình 2. 26. Biểu đồ VOPC Use case Tạo đề thi thử.....	39
Hình 2. 27. Biểu đồ Basic Flow Use case tạo đề thi thử.....	40
Hình 2. 28. Biểu đồ thực thể liên kế .....	40
Hình 3. 1. Wireframe màn hình Chọn hạng bằng thi.....	44
Hình 3. 2. Wireframe màn hình Cài đặt.....	44
Hình 3. 3. Wireframe màn hình Học lý thuyết .....	45
Hình 3. 4. Wireframe màn hình Xem mẹo thi .....	45
Hình 3. 5. Wireframe màn hình Học biển báo .....	46
Hình 3. 6. Wireframe màn hình Tạo thông báo .....	46
Hình 3. 7. Wireframe màn hình Xem câu hỏi hay sai.....	47
Hình 3. 8. Wireframe màn hình Thi sát hạch.....	47
Hình 3. 9. Wireframe màn hình Kết thúc thi sát hạch .....	48
Hình 3. 10. Wireframe màn hình Kết quả.....	48
Hình 3. 11. Wireframe màn hình Tạo đề thi thử.....	49
Hình 3. 12. Wireframe màn hình Trang chủ .....	49
Hình 3. 13. Màn hình giao diện Onboarding .....	50
Hình 3. 14. Màn hình giao diện Trang chủ .....	51
Hình 3. 15. Màn hình giao diện Học lý thuyết.....	51
Hình 3. 16. Màn hình giao diện Học biển báo .....	52
Hình 3. 17. Màn hình giao diện Xem mẹo thi .....	52
Hình 3. 18. Màn hình giao diện Câu hỏi hay sai.....	53
Hình 3. 19. Màn hình giao diện Xem tỉ lệ đỗ .....	53
Hình 3. 20. Màn hình giao diện Cài đặt.....	54
Hình 3. 21. Màn hình giao diện Thông báo .....	54
Hình 3. 22. Màn hình giao diện Thi sát hạch.....	55
Hình 3. 23. Màn hình giao diện Tạo đề thi thử.....	55



Hình 3. 24. Màn hình giao diện Kết thúc thi sát hạch .....	56
Hình 3. 25. Màn hình giao diện Kết quả.....	56

## DANH MỤC BẢNG BIỂU

Bảng 1. 1. Mô tả Use case Học lý thuyết.....	23
Bảng 1. 2. Mô tả Use case Chọn hạng bằng thi .....	25
Bảng 1. 3. Mô tả Use case Kết thúc thi sát hạch.....	27
Bảng 1. 4. Mô tả Use case Xem mẹo thi.....	29
Bảng 1. 5. Mô tả Use case Tạo thông báo.....	30
Bảng 1. 6. Mô tả Use case Xem câu hỏi hay sai.....	33
Bảng 1. 7. Mô tả Use case Học biển báo .....	35
Bảng 1. 8. Mô tả Use case Thi sát hạch .....	36
Bảng 1. 9. Mô tả Use case Tạo đề thi thử .....	38
Bảng 2. 1. Bảng CategoryLicense.....	41
Bảng 2. 2. Bảng CategoryQuestion.....	41
Bảng 2. 3. Bảng Examination .....	41
Bảng 2. 4. Bảng History .....	41
Bảng 2. 5. Bảng ImageQuestion .....	42
Bảng 2. 6. Bảng Notification.....	42
Bảng 2. 7. Bảng Question .....	42
Bảng 2. 8. Bảng QuestionWrong .....	43
Bảng 2. 9. Bảng Trick .....	43
Bảng 3. 1. Kiểm thử các bảng trong cơ sở dữ liệu.....	59
Bảng 3. 2. Kiểm thử chức năng.....	59

## MỞ ĐẦU

**Tên đề tài:** Xây dựng ứng dụng ôn thi lý thuyết bằng lái xe sử dụng Android Studio.

**Lí do chọn đề tài:**

Hiện nay, an toàn giao thông luôn là một trong những vấn đề quan trọng và đáng chú ý tại Việt Nam. Mỗi ngày, có hàng nghìn vụ tai nạn giao thông xảy ra và nguyên nhân chủ yếu của chúng thường bắt nguồn từ những vi phạm về luật lệ giao thông.

Một phần nguyên nhân của tình trạng này là do kiến thức giao thông của một số người lái xe còn hạn chế, thiếu hiểu biết và kiến thức về luật lệ giao thông.

Trước nhu cầu đó, em bắt tay vào việc xây dựng một ứng dụng ôn thi lý thuyết lái xe, không chỉ giúp người dùng chuẩn bị tốt cho bài thi mà còn trang bị những kiến thức cho người lái, góp phần vào việc giảm thiểu tai nạn giao thông và bảo vệ tính mạng của hàng triệu người trên khắp cả nước.

**Mục tiêu nghiên cứu:**

Áp dụng những kiến thức đã được học về Android Studio, Room Database, ngôn ngữ lập trình Kotlin để xây dựng một ứng dụng ôn thi lý thuyết bằng lái xe trên nền tảng Android và đưa vào sử dụng trong thực tế.

Tạo ra một giao diện người dùng thân thiện, dễ sử dụng, giúp người dùng dễ dàng tiếp cận và tiếp thu kiến thức.

Cung cấp các chức năng ôn tập linh hoạt như lựa chọn chủ đề, làm bài kiểm tra, xem lại kết quả... để người dùng có thể tự kiểm tra kiến thức và chuẩn bị tốt cho kỳ thi sát hạch lái xe.

Đảm bảo tính năng ổn định, mượt mà trên các thiết bị di động Android phổ biến.

**Đối tượng nghiên cứu:**

Sử dụng Figma để thiết kế giao diện cho ứng dụng.

Sử dụng Android Studio để xây dựng ứng dụng.

Sử dụng ngôn ngữ lập trình Kotlin cùng các thư viện đi kèm.

Sử dụng Room Database để quản lý cơ sở dữ liệu.

Sử dụng Jetpack Compose để xây dựng giao diện UI cho ứng dụng.

### **Phạm vi nghiên cứu:**

Ứng dụng ôn thi lý thuyết lái xe giúp người dùng có thể ôn tập kiến thức trước khi tham gia bài kiểm tra lý thuyết lái xe tại bất kỳ địa điểm nào, bất kỳ khi nào mà họ muốn một cách dễ dàng và thuận tiện.

### **Kết quả mong muốn:**

Tạo ra một ứng dụng di động hoàn chỉnh với đầy đủ chức năng đã đặt ra, cung cấp nội dung ôn thi lý thuyết một cách đầy đủ và chi tiết.

Người dùng có thể dễ dàng ôn tập và tự kiểm tra kiến thức thông qua giao diện trực quan và tiện lợi.

Nâng cao hiệu suất học tập của người dùng thông qua việc cung cấp các bài kiểm tra, câu hỏi trắc nghiệm đa dạng, phong phú và giải thích chi tiết để tăng cường khả năng hiểu và nhớ lâu của người dùng.

Cung cấp cho người dùng những kiến thức hữu ích để hỗ trợ quá trình ôn thi và chuẩn bị cho kỳ thi lý thuyết lái xe một cách tự tin và hiệu quả.

### **Cấu trúc báo cáo:**

Báo cáo thực tập được chia làm 3 chương:

Chương 1: Cơ sở lý thuyết.

Chương 2: Phân tích thiết kế hệ thống.

Chương 3: Thiết kế giao diện, cài đặt, kiểm thử.

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### 1.1. Giới thiệu về hệ điều hành Android

#### 1.1.1. Lịch sử hệ điều hành Android

**Hệ điều hành Android** là một trong những nền tảng phổ biến nhất trên thị trường di động hiện nay. Được phát triển bởi Google, Android đã trở thành một phần không thể thiếu của cuộc sống hàng ngày, không chỉ trong việc sử dụng điện thoại di động mà còn trong nhiều thiết bị thông minh khác như máy tính bảng, smartwatch, smart TV và nhiều thiết bị khác.

Được thiết kế dựa trên nền tảng mã nguồn mở Linux. Năm 2003, Android Inc được thành lập bởi một nhóm các nhà phát triển, bao gồm Andy Rubin, Rich Miner, Nick Sears và Chris White, với mục tiêu phát triển một hệ điều hành cho thiết bị di động. Sau khi Google mua lại Android vào năm 2005, nền tảng này bắt đầu phát triển nhanh chóng.

Vào tháng 11 năm 2007, Android được giới thiệu công khai cùng với một liên minh các nhà sản xuất và nhà mạng, thông qua việc ra mắt điện thoại di động đầu tiên chạy hệ điều hành Android, T-Mobile G1 (hay còn gọi là HTC Dream). Từ đó, Android đã trở thành một trong những hệ điều hành phổ biến nhất trên thế giới, và tiếp tục phát triển qua các phiên bản cập nhật định kỳ, mang lại những tính năng mới và cải thiện đáng kể về hiệu suất và bảo mật.



Hình 1. 1. Lịch sử phát triển của Android

Trong thập kỷ qua, Android đã không ngừng khẳng định vị thế của mình là một trong những hệ điều hành phổ biến và ảnh hưởng nhất trên toàn cầu. Với sự phát triển vượt bậc từ thị phần đến số lượng người dùng, Android đã trở thành một biểu tượng của sự sáng tạo và tiện ích trong ngành công nghiệp di động. Thành công của nó không chỉ dừng lại ở việc thu hút một lượng lớn người dùng mỗi ngày, mà còn thể hiện qua việc tạo ra một hệ sinh thái ứng dụng phong phú và đa dạng trên Google Play Store.

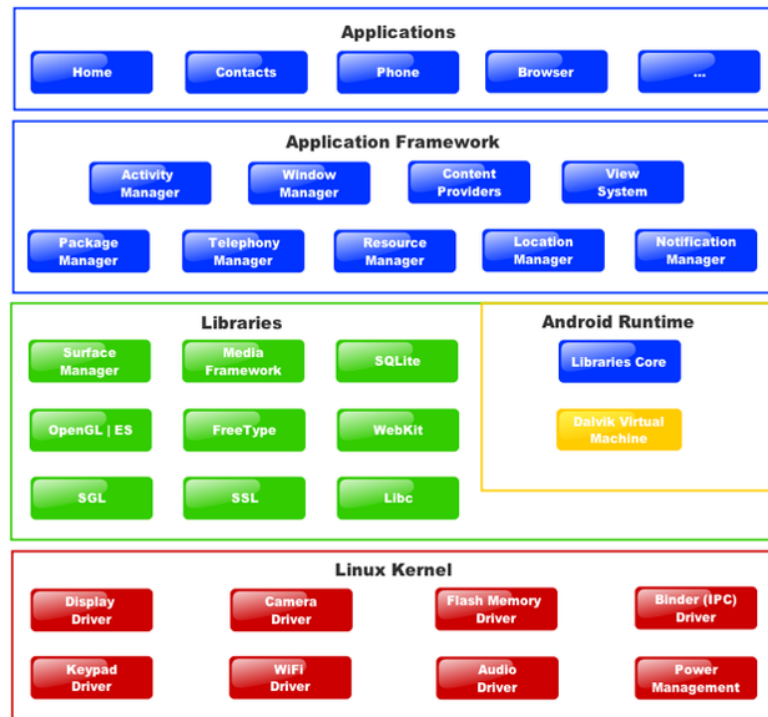
Không chỉ là một hệ điều hành cho điện thoại thông minh, Android còn mở rộng ảnh hưởng của mình sang nhiều loại thiết bị khác như máy tính bảng, đồng hồ thông minh và thiết bị đeo thông minh khác. Việc liên tục cập nhật và phát triển các phiên bản mới cũng chứng tỏ cam kết của Google đối với việc cải thiện trải nghiệm người dùng và tối ưu hóa hiệu suất của hệ điều hành này.

Android cũng đã có những bước tiến đáng kể trong việc cải thiện bảo mật và quyền riêng tư của người dùng. Google đã liên tục cập nhật và giới thiệu các tính năng mới như bảo vệ theo thời gian thực, kiểm soát quyền truy cập ứng dụng, và cập nhật bảo mật định kỳ qua Google Play Protect. Những cải tiến này không chỉ nâng cao niềm tin của người dùng đối với hệ điều hành, mà còn đáp ứng các yêu cầu ngày càng khắt khe về bảo mật thông tin trong kỷ nguyên số.

Trong một thị trường đầy cạnh tranh như ngày nay, Android vẫn tiếp tục duy trì vị thế của mình như một trong những người dẫn đầu, thúc đẩy sự tiến bộ và đổi mới trong ngành công nghiệp công nghệ thông tin. Điều này chứng tỏ sự linh hoạt và khả năng thích ứng của Android với những thách thức và yêu cầu ngày càng tăng của thị trường di động.

### **1.1.2. Kiến trúc của hệ điều hành Android**

Kiến trúc của hệ điều hành Android được chia thành 5 phần và 4 lớp chính như trong hình bên dưới.



Hình 1. 2. Kiến trúc của Android

### 1.1.2.1. Applications



Hình 1. 3. Application

Applications là lớp trên cùng của kiến trúc Android. Cung cấp các ứng dụng bao gồm các phần mềm có các chức năng mặc định của thiết bị Mobile. Nó cũng có thể phát triển để sử dụng Application Framework (AOSP). Các ứng dụng này được phát triển trên Eclipse IDE hoặc Android Studio IDE, và sử dụng ARM Emulator cơ bản on QEMU (Quick EMUlator).

### 1.1.2.2. Application Framework



Hình 1. 4. Tầng Application Framework

Tầng Application Framework là một phần quan trọng của hệ điều hành Android, đóng vai trò là một bộ khung cung cấp các dịch vụ và tài nguyên cần thiết cho việc phát triển ứng dụng di động. Đây là nơi mà các nhà phát triển có thể tạo ra các ứng dụng đa dạng, từ ứng dụng giải trí đến ứng dụng doanh nghiệp, với sự hỗ trợ từ các thành phần và tính năng đã được xây dựng sẵn.

Tầng Application Framework cung cấp một loạt các API (Application Programming Interface) cho việc phát triển ứng dụng. Các API này bao gồm các công cụ và giao diện lập trình ứng dụng để truy cập các chức năng cụ thể của hệ điều hành Android như quản lý nguồn dữ liệu, giao tiếp mạng, quản lý tập tin, định vị địa lý và nhiều hơn nữa. Nhờ vào các API này, việc phát triển ứng dụng trên nền tảng Android trở nên linh hoạt và dễ dàng hơn.

Ngoài ra, Tầng Application Framework cũng bao gồm các thành phần giao diện người dùng (UI) như Activities, Fragments và Views, giúp cho việc xây dựng giao diện người dùng trở nên thuận tiện hơn. Các nhà phát triển có thể tận dụng các thành phần này để tạo ra các trải nghiệm người dùng độc đáo và thu hút.

### **1.1.2.3. Android Runtime**

Android Runtime (ART) là một phần không thể thiếu trong hệ điều hành Android, đóng vai trò quan trọng trong việc cung cấp môi trường chạy ứng dụng cho các thiết bị di động. ART bao gồm các thành phần như thư viện lõi và máy ảo Dalvik (DVM), đồng thời là nền tảng cho Tầng Application Framework, hỗ trợ việc thực thi các ứng dụng Android.

Tương tự như Máy ảo Java (JVM), máy ảo Dalvik (DVM) được thiết kế và tối ưu hóa đặc biệt cho Android để đảm bảo hiệu suất tối ưu và tiết kiệm tài nguyên. DVM phụ thuộc vào nhân Linux lớp để quản lý bộ nhớ và phân luồng, giúp thiết bị chạy nhiều ứng dụng một cách hiệu quả.

ART cung cấp các thư viện cốt lõi cho việc triển khai ứng dụng Android bằng ngôn ngữ lập trình Java hoặc Kotlin. Điều này tạo điều kiện thuận lợi cho



các nhà phát triển để tạo ra các ứng dụng mạnh mẽ và đa dạng, sử dụng các công nghệ và tính năng tiên tiến nhất của nền tảng Android.



*Hình 1. 5. Android Runtime*

#### 1.1.2.4. Native Libraries



*Hình 1. 6. Native Libraries*

Platform Libraries là một phần quan trọng của hệ điều hành Android, bao gồm các thư viện lõi được viết bằng C/C++ và các thư viện dựa trên Java như Media, Graphics, Surface Manager, OpenGL, và nhiều thư viện khác nữa. Các thư viện này đóng vai trò quan trọng trong việc hỗ trợ cho quá trình phát triển ứng dụng trên nền tảng Android.

**Media library** cung cấp hỗ trợ để phát và ghi lại các định dạng âm thanh và video.

**Surface manager** chịu trách nhiệm quản lý quyền truy cập vào hệ thống con hiển thị.

**SGL và OpenGL** đều là giao diện chương trình ứng dụng đa ngôn ngữ, đa nền tảng (API) được sử dụng cho đồ họa máy tính 2D và 3D.

**SQLite** cung cấp hỗ trợ cơ sở dữ liệu và **FreeType** cung cấp hỗ trợ phông chữ.

**Web-Kit**, công cụ trình duyệt web mã nguồn mở này cung cấp tất cả các chức năng để hiển thị nội dung web và đơn giản hóa việc tải trang.

**SSL (Secure Sockets Layer)** là công nghệ bảo mật để thiết lập liên kết được mã hóa giữa máy chủ web và trình duyệt web.

#### 1.1.2.5. Linux Kernel

Linux Kernel là trái tim của kiến trúc Android, đóng vai trò quan trọng trong việc quản lý và điều khiển các thành phần phần cứng trên thiết bị di động. Kernel này quản lý và điều khiển tất cả các trình điều khiển cần thiết như trình điều khiển hiển thị, máy ảnh, Bluetooth, âm thanh, bộ nhớ và nhiều thành phần khác.

##### Các tính năng của Linux kernel là:

- + **Security**: Nhân Linux xử lý bảo mật giữa ứng dụng và hệ thống.
- + **Memory Management**: Nó xử lý hiệu quả việc quản lý bộ nhớ, do đó mang lại sự tự do để phát triển các ứng dụng.
- + **Process Management**: Nó quản lý quy trình tốt, phân bổ tài nguyên cho các quy trình bất cứ khi nào cần.
- + **Network Stack**: Nó xử lý hiệu quả việc truyền thông mạng.
- + **Driver Model**: Nó đảm bảo rằng ứng dụng hoạt động bình thường trên thiết bị và các nhà sản xuất phần cứng chịu trách nhiệm xây dựng trình điều khiển của họ vào bản dựng Linux.

#### 1.1.3. Ưu và nhược điểm của hệ điều hành Android

##### 1.1.3.1. Ưu điểm của hệ điều hành Android

**Đa dạng thiết bị**: Hệ điều hành Android hỗ trợ từ điện thoại thông minh đến máy tính bảng và thiết bị đeo thông minh, Android có mặt trên một loạt các thiết bị khác nhau, cung cấp sự linh hoạt và lựa chọn cho người dùng.

**Tích hợp dịch vụ Google:** Android tích hợp sâu các dịch vụ của Google như Gmail, Google Maps, Google Drive và nhiều dịch vụ khác. Điều này mang lại cho người dùng tiện ích trong việc truy cập và quản lý dữ liệu của họ trên các thiết bị Android.

**Đa nhiệm và tùy biến cao:** Android cho phép người dùng chạy nhiều ứng dụng cùng một lúc, chia sẻ nội dung dễ dàng và tùy chỉnh giao diện theo sở thích cá nhân. Từ việc thay đổi hình nền, biểu tượng ứng dụng đến sử dụng các gói icon và launcher khác nhau, Android cho phép người dùng tùy chỉnh thiết bị của họ theo cách mà họ muốn.

**Cộng đồng phát triển mạnh mẽ:** Do tính mở của mã nguồn và sự phổ biến của nền tảng, Android thu hút một cộng đồng phát triển mạnh mẽ. Kết quả là, Google Play Store có hàng nghìn ứng dụng và trò chơi khác nhau, đáp ứng nhu cầu đa dạng của người dùng trên toàn thế giới.

#### 1.1.3.2. Nhược điểm của hệ điều hành Android

**Phân tán và không đồng nhất:** Một trong những thách thức lớn nhất của Android là sự phân tán của nền tảng. Với hàng trăm nhà sản xuất thiết bị và biến thể phần mềm khác nhau, việc cập nhật và hỗ trợ từ các nhà sản xuất không đồng nhất, dẫn đến sự không đồng nhất trong việc cập nhật hệ điều hành và bảo mật giữa các thiết bị.

**Tiêu tốn pin và hiệu suất:** Một số ứng dụng chạy ngầm và tùy biến giao diện có thể dẫn đến tiêu tốn pin nhanh chóng và giảm hiệu suất của thiết bị. Điều này có thể làm giảm trải nghiệm người dùng và tăng chi phí sử dụng.

**Bảo mật:** Do tính mở của mã nguồn và sự đa dạng của hệ thống, Android thường phải đối mặt với rủi ro bảo mật từ các phần mềm độc hại và lỗ hổng bảo mật. Người dùng cần phải tự bảo vệ thông tin cá nhân và thiết bị của họ bằng cách cập nhật phần mềm và sử dụng các ứng dụng bảo mật phù hợp.

**Trải nghiệm người dùng không đồng nhất:** Sự đa dạng về giao diện và tùy chỉnh có thể làm cho trải nghiệm người dùng trở nên không nhất quán giữa các thiết bị và phiên bản Android khác nhau. Điều này có thể gây ra sự bất tiện

và khó khăn cho người dùng khi chuyển đổi giữa các thiết bị hoặc phiên bản Android khác nhau.

## 1.2. Giới thiệu về Android Studio

**Android Studio** là một môi trường phát triển tích hợp (IDE) được phát triển bởi Google, chuyên dành cho việc phát triển ứng dụng di động chạy trên hệ điều hành Android. Được ra mắt lần đầu tiên vào năm 2013, Android Studio nhanh chóng trở thành công cụ phổ biến nhất và được ưa chuộng nhất trong cộng đồng phát triển ứng dụng Android.



*Hình 1. 7. Logo Android Studio*

**Android Studio** lần đầu tiên được công bố vào năm 2013 tại hội nghị Google I/O và sau đó được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Với việc ra đời của Android Studio, nó đã trở thành công cụ phát triển chính thức cho ứng dụng Android, thay thế hoàn toàn Eclipse IDE. Google khuyến khích mạnh mẽ việc sử dụng Android Studio, mang lại sự tiện lợi, linh hoạt và hiệu quả cho việc phát triển ứng dụng Android. Điều này đã tạo ra một bước tiến lớn trong cộng đồng phát triển và định hình lại quy trình phát triển ứng dụng trên nền tảng Android.

### **Ưu điểm của Android Studio**

**Giao diện người dùng thân thiện:** Android Studio có một giao diện người dùng dễ sử dụng và trực quan, giúp cho việc phát triển ứng dụng trở nên thuận tiện và hiệu quả.

**Trình biên soạn mã thông minh:** Android Studio đi kèm với một trình biên soạn mã thông minh, hỗ trợ kiểm tra cú pháp, hoàn thành tự động mã, và gợi ý mã, giúp tăng cường năng suất và giảm thiểu lỗi lập trình.

**Giao diện thiết kế và xem trước:** Android Studio cung cấp các công cụ mạnh mẽ cho việc thiết kế giao diện người dùng và xem trước trực quan, giúp cho việc tạo ra các ứng dụng với giao diện đẹp mắt và dễ sử dụng.

**Quản lý dự án và tài nguyên:** Android Studio cho phép quản lý dự án một cách hiệu quả, bao gồm quản lý tài nguyên như hình ảnh, âm thanh và các tệp tin khác.

**Kiểm thử và gỡ lỗi:** Android Studio cung cấp các công cụ mạnh mẽ cho việc kiểm thử và gỡ lỗi ứng dụng, giúp cho việc phát triển ứng dụng trở nên dễ dàng hơn.

**Hỗ trợ cập nhật và tích hợp:** Android Studio liên tục được cập nhật và tích hợp với các công nghệ và dịch vụ mới nhất của Google, giúp cho việc phát triển ứng dụng trên nền tảng Android luôn đảm bảo sự tiên tiến và hiệu quả.

**Cộng đồng hỗ trợ:** Android Studio được hỗ trợ bởi một cộng đồng lớn của các nhà phát triển và chuyên gia Android trên toàn thế giới, giúp người dùng dễ dàng tìm kiếm giải pháp cho các vấn đề phát triển.

### **Nhược điểm của Android Studio**

+ Android Studio yêu cầu một cấu hình máy tính mạnh mẽ để hoạt động mượt mà, đặc biệt là khi xử lý các dự án lớn. Điều này có thể gây khó khăn cho những người sử dụng máy tính có cấu hình thấp.

+ Việc kiểm tra hoạt động thông qua giả lập gây đơ, lag, giật máy và tiêu tốn pin.

+ Đôi khi, Android Studio có thể gặp phải vấn đề về tương thích với các phiên bản Android mới nhất hoặc các thư viện bên thứ ba, điều này có thể gây rắc rối và tốn thêm thời gian giải quyết vấn đề.

### 1.3. Giới thiệu về ngôn ngữ lập trình Kotlin

**Kotlin** là một ngôn ngữ lập trình hiện đại, được phát triển bởi JetBrains, công ty nổi tiếng với việc sản xuất các công cụ phát triển phần mềm như IntelliJ IDEA, PhpStorm và PyCharm. Kotlin được giới thiệu lần đầu vào năm 2011, nhưng nó chính thức trở thành ngôn ngữ lập trình chính thức cho việc phát triển ứng dụng Android từ năm 2017.

Một số đặc điểm của Kotlin bao gồm:

**Tính hiện đại:** Kotlin được thiết kế với các tính năng hiện đại của ngôn ngữ lập trình như null safety, lambda expressions, extension functions, coroutines và nhiều tính năng khác. Điều này giúp làm tăng năng suất và giảm thiểu lỗi trong quá trình phát triển phần mềm.

**Tính tương thích:** Kotlin hoàn toàn tương thích với Java, điều này có nghĩa là các nhà phát triển có thể tích hợp mã Kotlin vào dự án Java hiện có mà không gặp vấn đề lớn. Điều này tạo điều kiện thuận lợi cho việc chuyển đổi từ Java sang Kotlin.

**Tính hướng đối tượng:** Kotlin là một ngôn ngữ hướng đối tượng, hỗ trợ các tính năng như inheritance, encapsulation, polymorphism và abstraction.

**Tính đa nền tảng:** Kotlin không chỉ được sử dụng cho việc phát triển ứng dụng Android mà còn cho các ứng dụng máy chủ, ứng dụng web và ứng dụng máy tính. JetBrains cũng đã giới thiệu Kotlin/Native, một công cụ cho phép biên dịch mã Kotlin thành mã máy tại thời điểm biên dịch, cho phép sử dụng Kotlin để phát triển ứng dụng desktop và ứng dụng di động không dựa trên JVM.

**Cộng đồng phát triển mạnh mẽ:** Kotlin có một cộng đồng phát triển đông đảo, với nhiều tài liệu học tập, ví dụ và thư viện mã nguồn mở. Điều này giúp người dùng dễ dàng tìm kiếm và chia sẻ kiến thức, cũng như giải quyết vấn đề khi phát triển ứng dụng.

### 1.3.1. Lịch sử hình thành

**Kotlin** ra đời vào năm 2011 bởi các nhà phát triển tại JetBrains, một công ty phát triển các công cụ phát triển phần mềm hàng đầu. Ban đầu, Kotlin được thiết kế nhằm mục đích giải quyết những hạn chế và nhược điểm của các ngôn ngữ lập trình hiện có, đặc biệt là Java.

Tháng 7/2011, ngôn ngữ lập trình Kotlin lần đầu tiên được giới thiệu đến công chúng bởi JetBrains. Sự ra đời của Kotlin nhận được sự chào đón tích cực từ cộng đồng phát triển phần mềm, đặc biệt là trong lĩnh vực phát triển ứng dụng di động. Tính hiện đại và tính năng tiên tiến của Kotlin đã thu hút sự quan tâm của các nhà phát triển, đồng thời giúp Kotlin nhanh chóng trở thành một trong những lựa chọn phổ biến cho việc phát triển ứng dụng Android.

Năm 2017, Kotlin chính thức trở thành ứng cử viên sáng giá nhất hoạt động trên Android dưới sự “hậu thuẫn” của Google. Sự ủng hộ từ Google đã giúp Kotlin nhanh chóng trở thành một trong những ngôn ngữ lập trình hàng đầu cho việc phát triển ứng dụng di động trên nền tảng Android.

### 1.3.2. Ứng dụng của Kotlin

Kotlin là một ngôn ngữ lập trình đa năng và linh hoạt, có thể được sử dụng cho nhiều mục đích khác nhau. Dưới đây là một số ứng dụng phổ biến của Kotlin:

**Phát triển ứng dụng di động:** Kotlin là ngôn ngữ chính thức được Google chọn làm ngôn ngữ lập trình cho việc phát triển ứng dụng di động trên nền tảng Android. Với tính năng tương thích tốt với Java và các tính năng hiện đại, Kotlin giúp tăng hiệu suất và giảm thiểu lỗi trong quá trình phát triển ứng dụng di động.

**Phát triển ứng dụng web:** Kotlin có thể được sử dụng để phát triển các ứng dụng web thông qua các framework như Kotlin/JS, Spring Boot và Ktor. Sự linh hoạt của Kotlin giúp tạo ra các ứng dụng web hiệu quả và dễ bảo trì.

**Phát triển ứng dụng máy chủ:** Kotlin cũng được sử dụng để phát triển các ứng dụng máy chủ thông qua các framework như Spring Boot, Ktor và

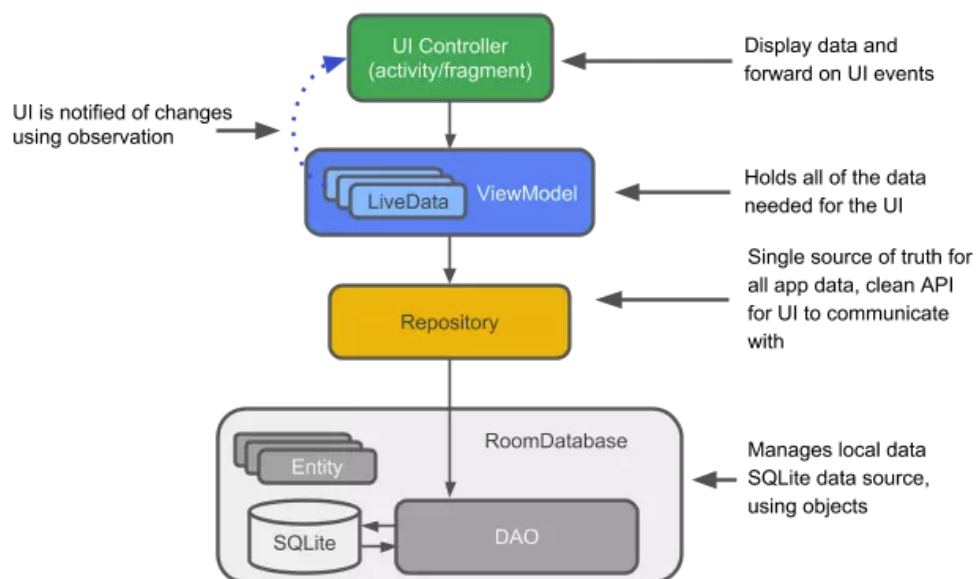
Vert... Với tính năng null safety và coroutine, Kotlin giúp giảm thiểu lỗi và tăng hiệu suất khi phát triển ứng dụng máy chủ.

**Phát triển ứng dụng desktop:** Kotlin/Native là một công cụ cho phép phát triển ứng dụng desktop trên nhiều nền tảng như Windows, macOS và Linux. Kotlin/Native biên dịch mã Kotlin thành mã máy tại thời điểm biên dịch, giúp tăng hiệu suất và giảm kích thước của ứng dụng.

**Các dự án AI và Machine Learning:** Kotlin cũng được sử dụng trong các dự án liên quan đến trí tuệ nhân tạo và học máy nhờ vào tính năng linh hoạt và dễ sử dụng của nó.

#### 1.4. Room Database

**Room Database** là một thành phần quan trọng trong Android Jetpack, được phát triển bởi Google để giúp việc làm việc với cơ sở dữ liệu trở nên dễ dàng và hiệu quả hơn trong ứng dụng Android. Nó được xây dựng trên nền tảng của SQLite, một hệ quản trị cơ sở dữ liệu nhẹ và phổ biến trên nền tảng di động.



Hình 1. 8. Thành phần của Room Database

Các thành phần chính của Room Database bao gồm:

**Entities:** Là các đối tượng (objects) được sử dụng để mô hình hóa cấu trúc dữ liệu của cơ sở dữ liệu. Mỗi entity tương ứng với một bảng trong cơ sở dữ liệu và các trường (fields) của entity sẽ tương ứng với các cột trong bảng.



**DAO (Data Access Object):** Là một interface định nghĩa các phương thức để thao tác với dữ liệu trong cơ sở dữ liệu. Các phương thức này có thể bao gồm các truy vấn (query) để lấy dữ liệu, các phương thức để chèn (insert), cập nhật (update) hoặc xóa (delete) dữ liệu.

**Database:** Là một lớp trừu tượng đại diện cho cơ sở dữ liệu. Nó chứa các đối tượng DAO và được sử dụng để tạo ra một đối tượng cơ sở dữ liệu và lấy các DAO tương ứng từ đó.

**Room Database cung cấp nhiều ưu điểm như:**

- + Dễ sử dụng: Với việc sử dụng các annotations và các thành phần trừu tượng hóa, Room giúp giảm bớt sự phức tạp và lặp lại trong việc làm việc với cơ sở dữ liệu.

- + Hiệu suất cao: Room được tối ưu hóa để làm việc hiệu quả với SQLite, giúp cải thiện hiệu suất và tăng trải nghiệm người dùng.

- + Tích hợp với LiveData và RxJava: Room tích hợp tốt với các thành phần khác của Android Architecture Components như LiveData và RxJava.

**Nhược điểm của Room Database:**

- + Khả năng mở rộng hạn chế: Mặc dù Room cung cấp một cách tiếp cận thuận tiện và linh hoạt cho việc làm việc với cơ sở dữ liệu trong ứng dụng Android, nhưng nó có thể gặp khó khăn trong việc mở rộng cho các ứng dụng có quy mô lớn hoặc cần tính mở rộng cao.

- + Học phức tạp ban đầu: Đối với những người mới bắt đầu, việc học và làm quen với Room Database có thể gặp phải một số khó khăn ban đầu do phải hiểu và áp dụng các khái niệm như DAO, Entities và Relations. Tuy nhiên, sau khi làm quen, việc sử dụng Room sẽ trở nên dễ dàng hơn.

## CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 2.1. Đặt vấn đề, giới thiệu đề tài

Mỗi năm, Việt Nam ghi nhận hàng chục nghìn vụ tai nạn giao thông. Đây là một con số đáng lo ngại, thể hiện tình trạng giao thông ở nước ta đang đối mặt với nhiều thách thức. Tai nạn giao thông không chỉ gây tổn thất về người mà còn ảnh hưởng đến kinh tế và xã hội.

Mặc dù đã có nhiều nỗ lực để tăng cường kiến thức cũng như hiểu biết của mọi người về an toàn giao thông, tuy nhiên tình hình giao thông vẫn còn nhiều bất cập. Nhiều lái xe chưa có đủ kiến thức, kỹ năng cũng như kinh nghiệm để xử lý những tình huống giao thông phức tạp, chưa có sự hiểu biết rõ ràng về luật giao thông đường bộ và sự chủ quan khi tham gia giao thông, dẫn tới vi phạm giao thông và nghiêm trọng hơn là gây ra tai nạn.

Ngoài ra, trên thị trường hiện nay có rất nhiều ứng dụng ôn thi lý thuyết bằng lái xe, tuy nhiên không phải tất cả các ứng dụng đều có khả năng đáp ứng được nhu cầu và mong muốn của người dùng. Một số ứng dụng còn hạn chế về nội dung, giao diện phức tạp gây khó khăn cho người dùng trong việc học tập và ôn thi lý thuyết bằng lái xe.

Trong bối cảnh này, việc “Xây dựng ứng dụng ôn thi lý thuyết bằng lái xe sử dụng Android Studio” sẽ mang lại nhiều lợi ích tích cực cho cộng đồng. Ứng dụng sẽ cung cấp cho người dùng một nền tảng học tập thuận tiện và linh hoạt, giúp họ tiết kiệm thời gian và chi phí so với việc tham gia các lớp học truyền thống. Thứ hai, ứng dụng cung cấp cho người dùng một cách tiếp cận dễ dàng và hiệu quả với các tài liệu ôn thi, bao gồm các câu hỏi trắc nghiệm, bài kiểm tra thử, và tài liệu học lý thuyết.

#### Phương pháp nghiên cứu

**Phân tích nhu cầu người dùng:** Tiến hành khảo sát và phân tích nhu cầu, mong muốn của người dùng khi ôn thi lý thuyết lái xe. Điều này giúp xác định những yếu tố quan trọng cần được tập trung và cải thiện trong ứng dụng.

**Nghiên cứu về nội dung:** Tìm hiểu và thu thập thông tin về các quy định, quy tắc giao thông và kiến thức lý thuyết cần thiết cho kỳ thi lái xe. Xây dựng nội dung học tập phong phú và đa dạng để đáp ứng nhu cầu học tập của người dùng.

**Thiết kế giao diện người dùng (UI/UX):** Phát triển giao diện người dùng thân thiện và dễ sử dụng. Sử dụng các nguyên tắc thiết kế UI/UX để tạo ra trải nghiệm người dùng tốt nhất.

**Kiểm thử và đánh giá:** Tiến hành kiểm thử thực tế và thu thập phản hồi từ người dùng để đánh giá hiệu quả của ứng dụng. Dựa vào kết quả, điều chỉnh và cải tiến ứng dụng để đảm bảo hoạt động tốt và đáp ứng được nhu cầu của người dùng.

**Theo dõi và cập nhật:** Liên tục theo dõi sự phản hồi và thói quen sử dụng của người dùng để cập nhật và cải thiện ứng dụng. Điều này giúp đảm bảo rằng ứng dụng luôn cung cấp cho người dùng trải nghiệm học tập tốt nhất và đáp ứng được những thay đổi trong quy định và kiến thức ôn thi lý thuyết bằng lái xe.

### **Hướng giải quyết**

**Cung cấp giải thích rõ ràng:** Đảm bảo rằng mỗi câu hỏi đều đi kèm với giải thích chi tiết và rõ ràng về đáp án. Giải thích này giúp người dùng hiểu rõ hơn về lý do và quy định đằng sau mỗi đáp án, từ đó nắm vững kiến thức hơn.

**Khả năng tính toán tỉ lệ đỗ:** Giúp người dùng nắm được khả năng thi đỗ của bản thân, từ đó lên kế hoạch cũng như thay đổi lộ trình ôn tập cho phù hợp.

**Tạo thông báo:** Người dùng có thể dễ dàng tạo thông báo để nhắc nhở học tập, từ đó tăng sự chủ động cũng như tính kỷ luật trong quá trình học tập.

**Giao diện thân thiện và dễ sử dụng:** Xây dựng một giao diện đẹp, thân thiện, giúp người dùng dễ dàng học tập và nắm bắt kiến thức.

## **2.2. Các yêu cầu chức năng**

**Mục đích:** Ứng dụng ôn thi lý thuyết bằng lái xe sẽ giúp người dùng tiết kiệm thời gian, tối đa năng suất trong việc học tập kiến thức giao thông. Người

dùng có thể chủ động về việc học, lên kế hoạch thời gian học phù hợp với bản thân mỗi người. Ứng dụng cũng được thiết kế với giao diện hiện đại, phát triển các tính năng cốt lõi như học và thi để đem lại trải nghiệm một cách tốt nhất.

#### **Các chức năng chính:**

**Học lý thuyết :** Người dùng có thể học tập theo các bộ đề hoặc lựa chọn học toàn bộ câu hỏi trong bộ đề thi.

**Thi sát hạch:** Người dùng có thể dễ dàng tiến hành thi thử theo các bộ đề có sẵn hoặc tạo ra một cách ngẫu nhiên.

**Học biển báo:** Người dùng có thể xem danh sách các biển báo theo từng loại, cùng với tên biển báo, hình ảnh và giải thích cho từng biển báo.

**Xem mẹo thi:** Người dùng có thể dễ dàng xem các mẹo thi.

**Xem câu hỏi hay sai:** Người dùng có thể xem và làm lại những câu hỏi mà mình đã làm sai trong quá trình học tập.

**Thay đổi chế độ sáng tối :** Người dùng có thể tùy chỉnh chế độ sáng tối để phù hợp với nhu cầu của bản thân.

**Tạo thông báo:** Người dùng có thể dễ dàng tạo thông báo trong quá trình học tập.

**Xem tỉ lệ đỗ:** Người dùng có thể xem tỉ lệ đỗ của bản thân được hệ thống tính toán theo số lượng bài thi mà người dùng vượt qua.

#### **Các yêu cầu khác**

Tối ưu hóa trải nghiệm với giao diện thân thiện, dễ sử dụng.

Chủ động hóa quá trình nhắc nhở học tập.

Nguồn thông tin đáng tin cậy và chính xác.

Giải thích chi tiết và đầy đủ cho từng câu hỏi.

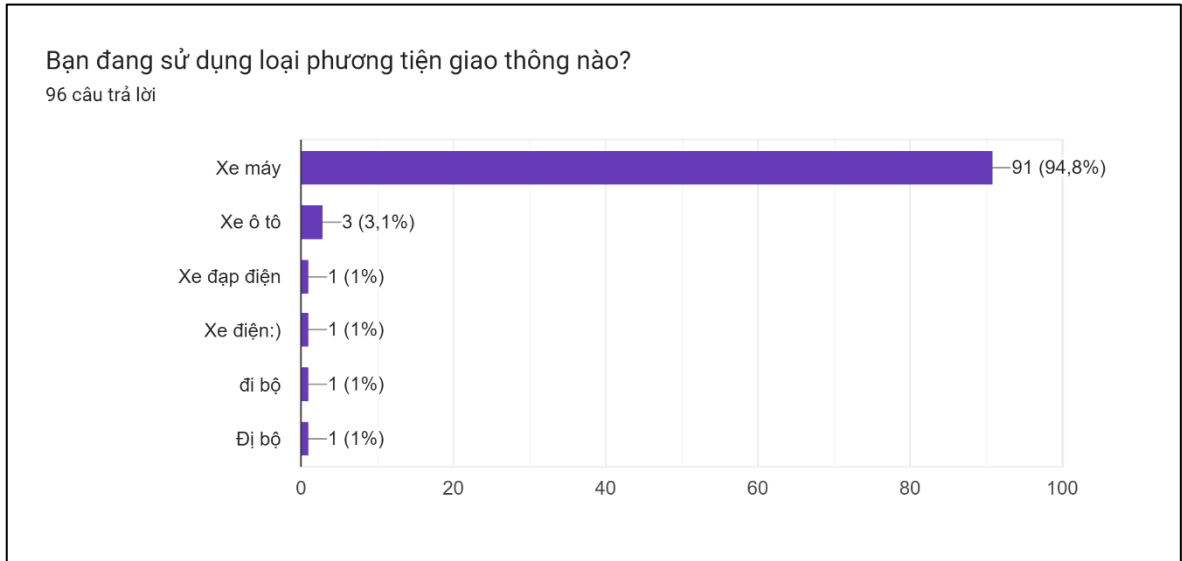
### **2.3. Tài liệu thu thập được**

Khảo sát người dùng là một bước quan trọng không thể thiếu trong quá trình phát triển và cải tiến ứng dụng ôn thi lý thuyết bằng lái xe. Để đảm bảo ứng dụng không chỉ đáp ứng nhu cầu của người học mà còn mang lại trải

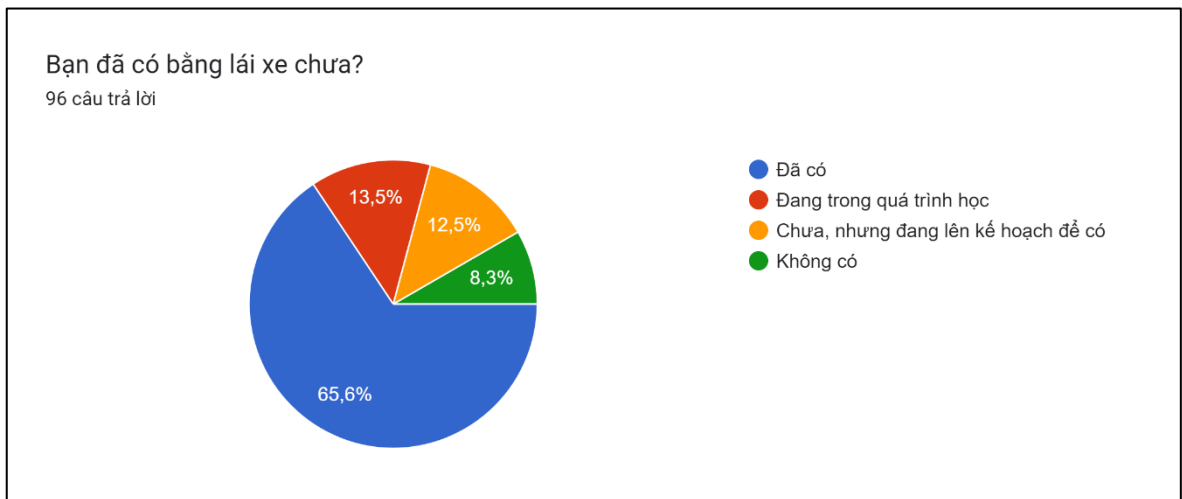
nghiệm học tập hiệu quả và thuận tiện, việc thu thập ý kiến phản hồi từ người dùng là cực kỳ cần thiết.

Việc khảo sát người dùng giúp chúng ta có thể tập trung vào việc nâng cao chất lượng nội dung, tối ưu hóa giao diện người dùng, và bổ sung các tính năng hữu ích như bài thi thử, mẹo ôn tập, hay hệ thống nhắc nhở học tập.

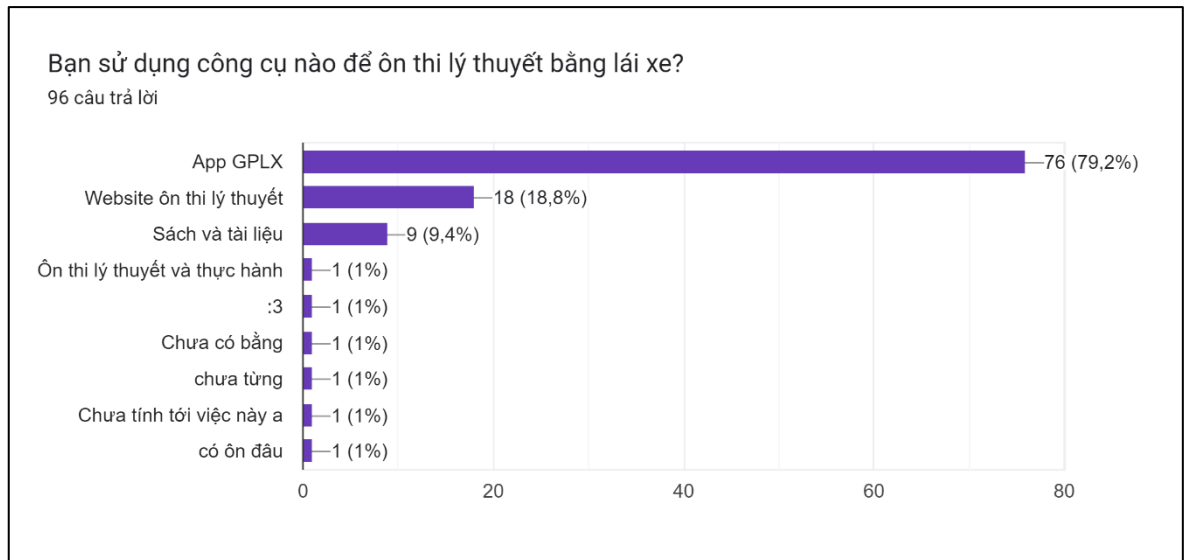
Tài liệu thu thập được trong quá trình khảo sát người dùng:



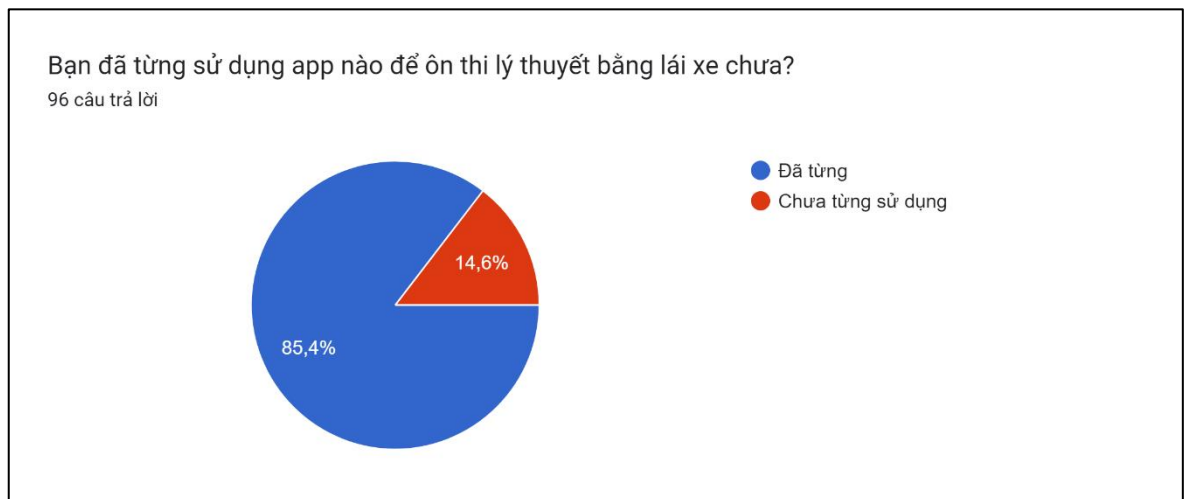
Hình 2. 1. Câu hỏi khảo sát người dùng 1



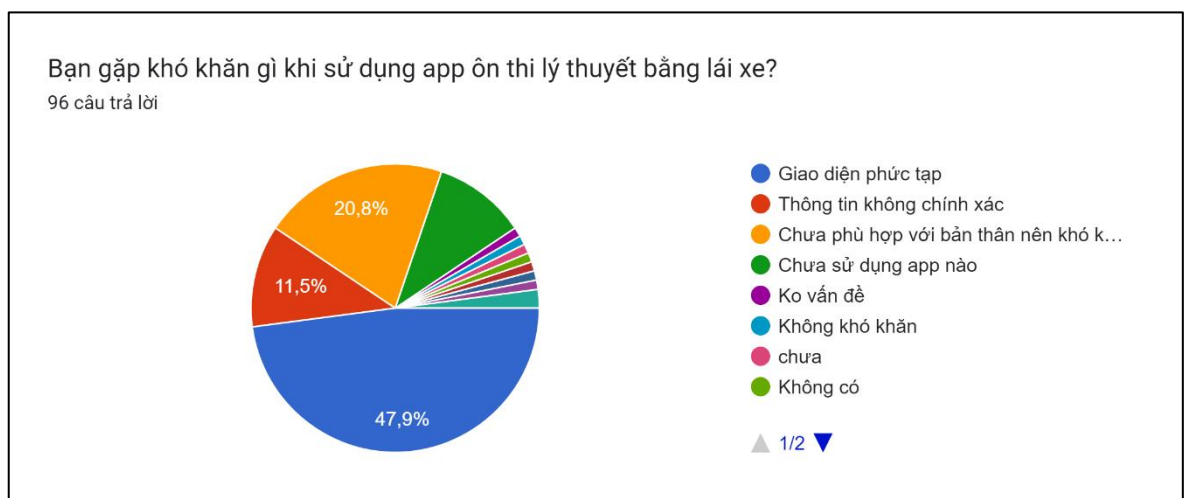
Hình 2. 2. Câu hỏi khảo sát người dùng 2



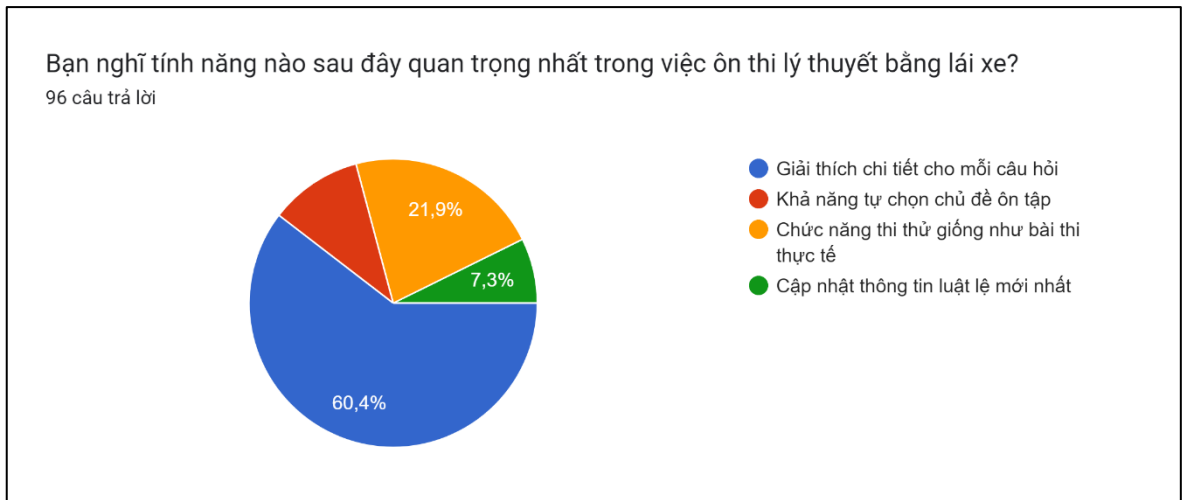
Hình 2. 3. Câu hỏi khảo sát người dùng 3



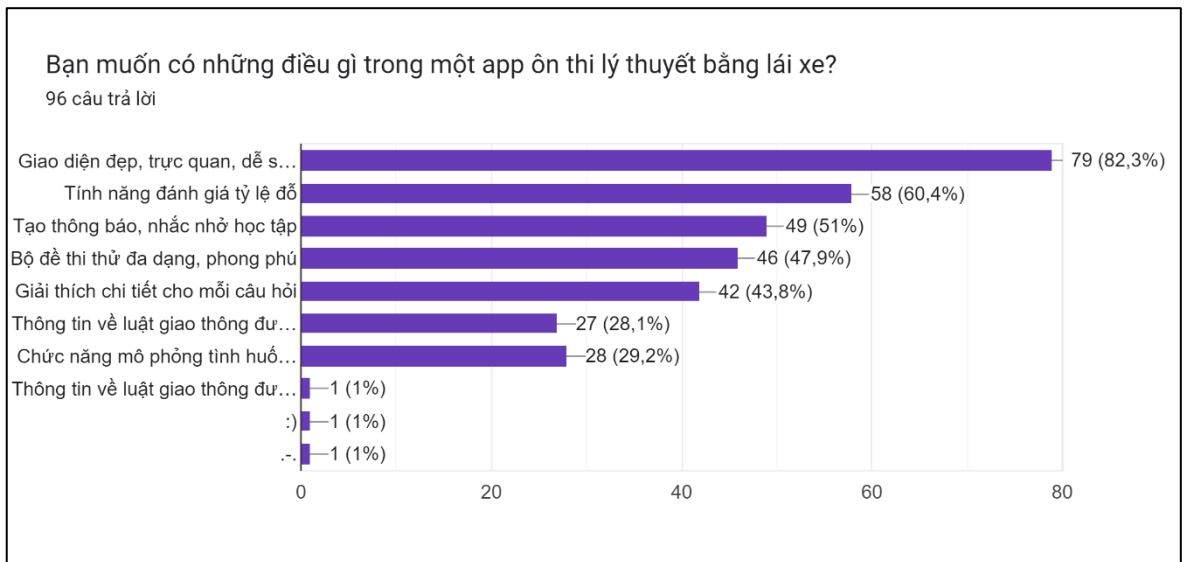
Hình 2. 4. Câu hỏi khảo sát người dùng 4



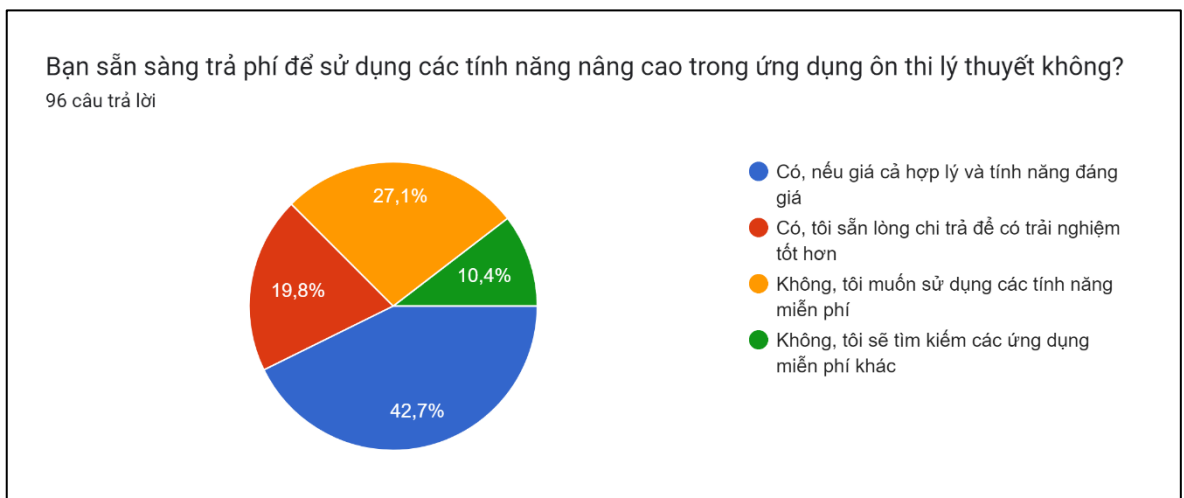
Hình 2. 5. Câu hỏi khảo sát người dùng 5



Hình 2. 6. Câu hỏi khảo sát người dùng 6



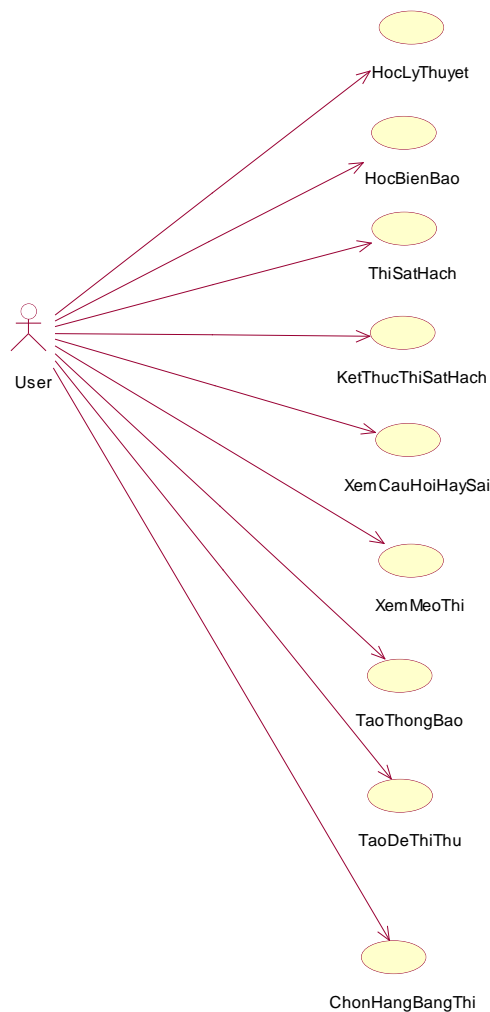
Hình 2. 7. Câu hỏi khảo sát người dùng 7



Hình 2. 8. Câu hỏi khảo sát người dùng 8

## 2.4. Phân tích thiết kế hệ thống

### 2.4.1. Biểu đồ use case tổng quát



Hình 2. 9. Biểu đồ use case tổng quát

**Use case Học lý thuyết:** Use case này cho phép người dùng tham gia học lý thuyết bằng lái xe.

**Use case Thi sát hạch:** Use case này cho phép người dùng tham gia thi sát hạch bằng lái xe.

**Use case Học biển báo:** Use case này cho phép người dùng học các biển báo giao thông.

**Use case Xem mẹo thi:** Use case này cho phép người dùng xem các mẹo thi.

**Use case Xem câu hỏi hay sai:** Use case này cho phép người dùng xem những câu hỏi hay sai.



**Use case Kết thúc thi sát hạch:** Use case này cho phép người dùng kết thúc bài thi sát hạch.

**Use case Tạo thông báo:** Use case này cho phép người dùng tạo thông báo.

**Use case Tạo đề thi thử:** Use case này cho phép người dùng tạo các đề thi thử.

**Use case Chọn hạng bằng thi:** Use case này cho phép người dùng chọn hạng bằng thi.

## 2.4.2. Mô tả và phân tích chi tiết Use case

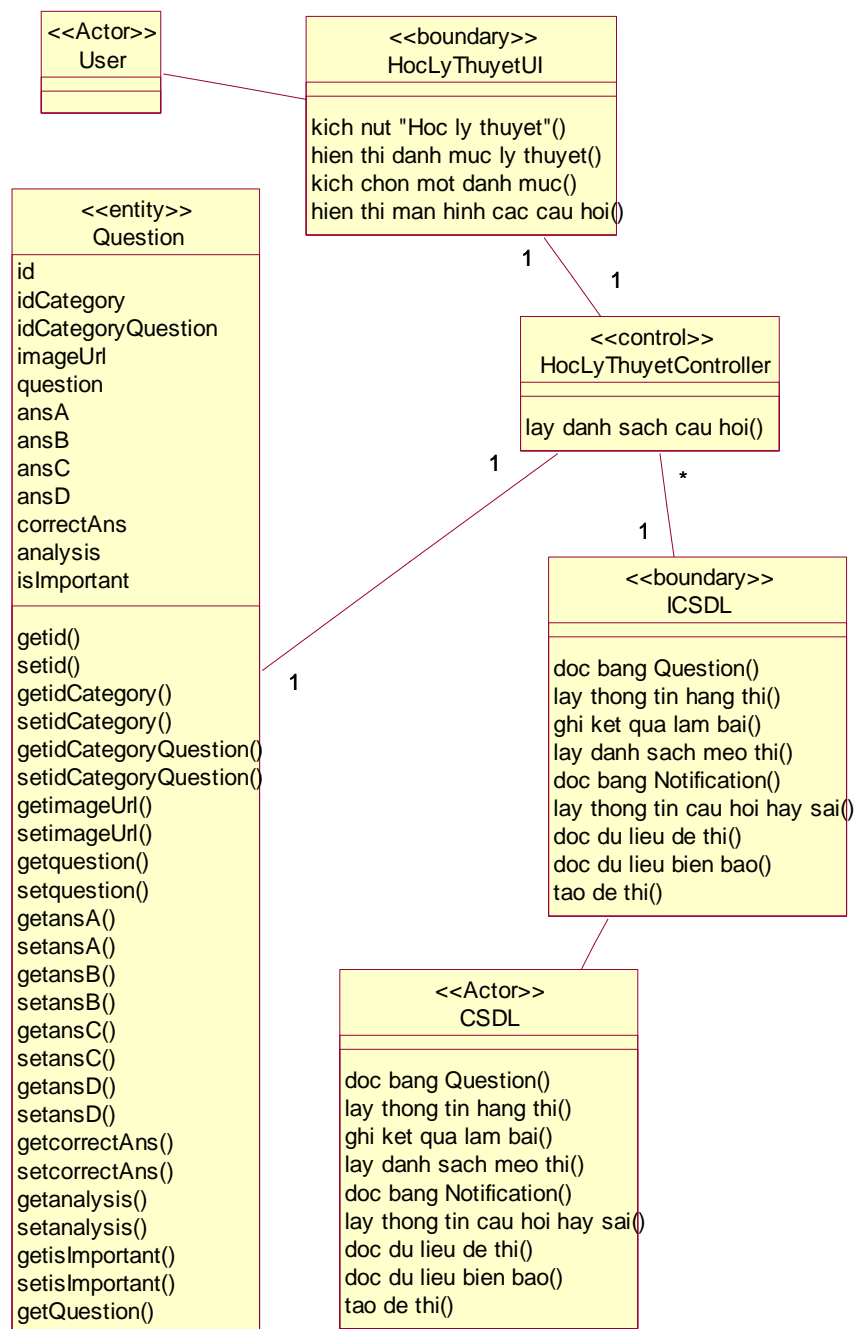
### 2.4.2.1. Use case Học lý thuyết

*Bảng 1. 1. Mô tả Use case Học lý thuyết*

<b>Mô tả</b>		Use case này cho phép người dùng tham gia học lý thuyết bằng lái xe.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	<ol style="list-style-type: none"> <li>1. Use case này bắt đầu khi người dùng kích vào nút “Học lý thuyết” ở trên màn hình chính của ứng dụng. Hệ thống sẽ hiển thị các danh mục lý thuyết lên màn hình.</li> <li>2. Khi người dùng kích chọn vào một danh mục lý thuyết, hệ thống sẽ lấy thông tin câu hỏi từ bảng Question và hiển thị lên màn hình. Use case kết thúc.</li> </ol>
	<b>Luồng rẽ nhánh</b>	<ol style="list-style-type: none"> <li>1. Tại bước 2 trong luồng cơ bản nếu không tìm thấy câu hỏi nào từ bảng Question. Hệ thống hiển thị danh sách trống và use case kết thúc.</li> <li>2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ</li> </ol>

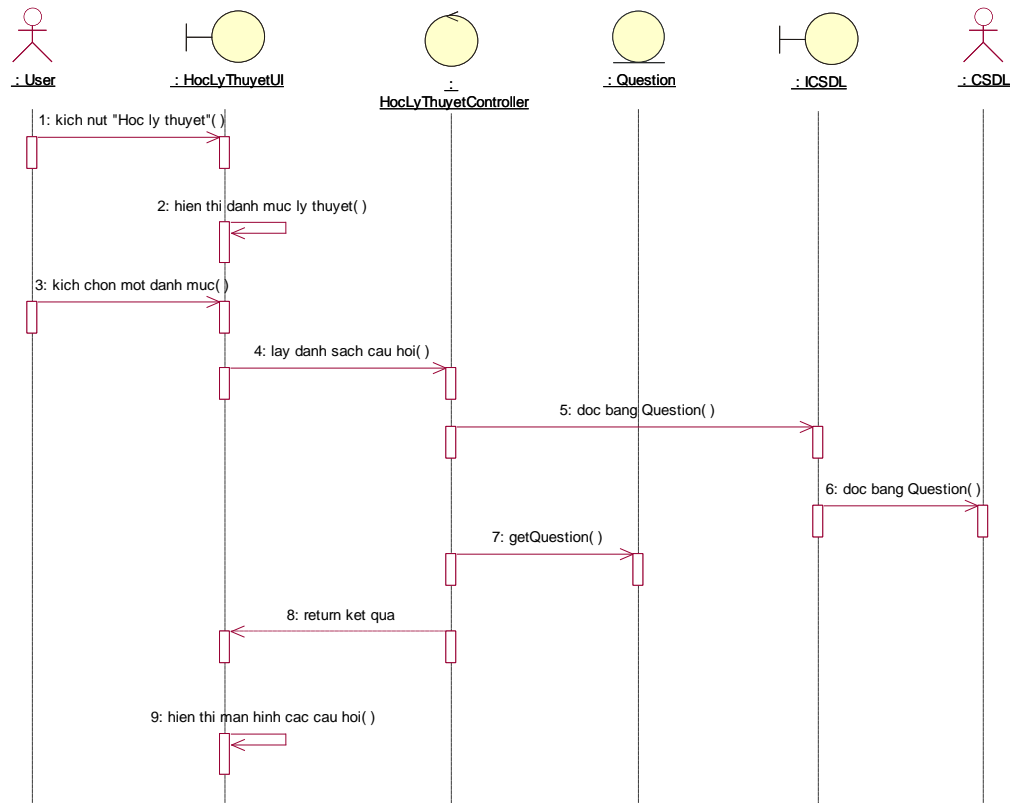
		liệu thì hệ thống sẽ hiển thị thông báo lỗi. Use case kết thúc.
<b>Các yêu cầu đặc biệt</b>	Không	
<b>Hậu điều kiện</b>	Không	
<b>Điểm mở rộng</b>	Không	

### a, Biểu đồ lớp phân tích



Hình 2. 10. Biểu đồ VOPC Use case Học lý thuyết

## b, Biểu đồ trình tự



Hình 2. 11. Biểu đồ Basic Flow Use case Học lý thuyết

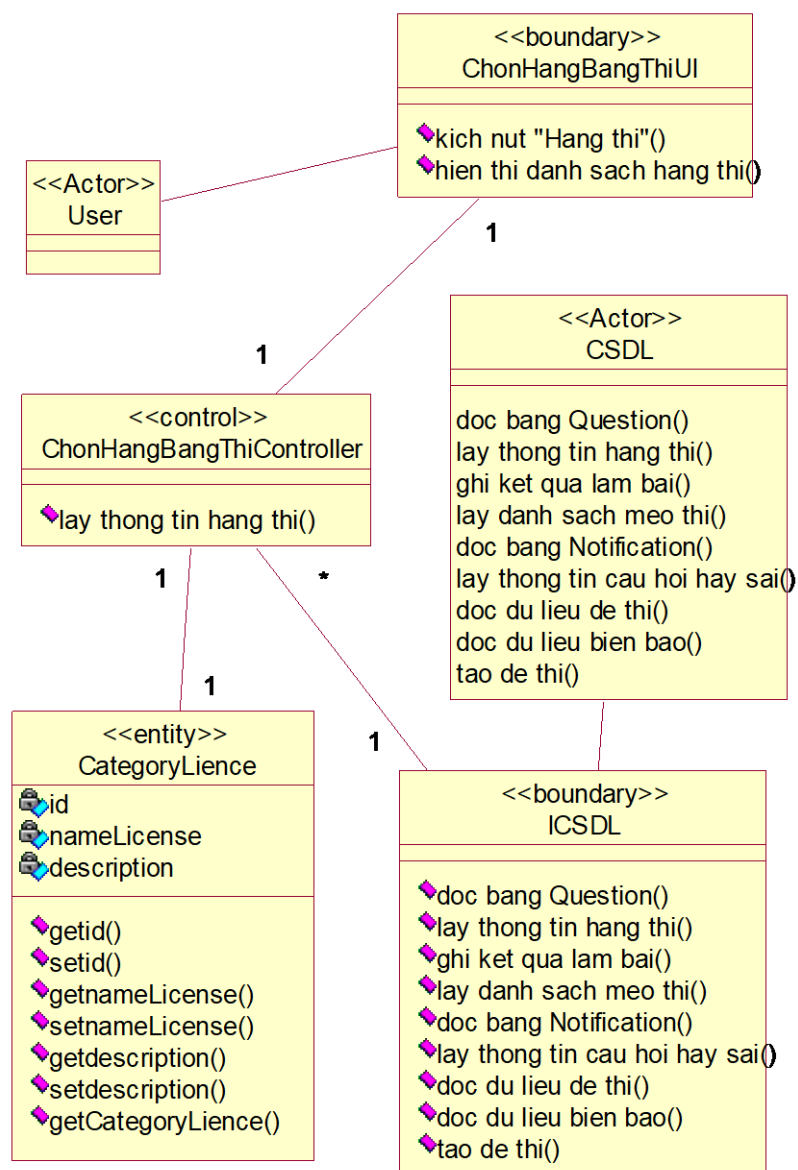
### 2.4.2.2. Use case Chọn hạng bằng thi

Bảng 1. 2. Mô tả Use case Chọn hạng bằng thi

<b>Mô tả</b>		Use case này cho phép người dùng chọn hạng bằng thi.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	1. Use case này bắt đầu khi người dùng kích vào nút “Hạng thi” ở trên màn hình giao diện. Hệ thống sẽ lấy thông tin hạng thi từ bảng CategoryLience và hiển thị danh sách hạng thi lên màn hình. Use case kết thúc.
	<b>Luồng rẽ nhánh</b>	1. Tại bước 1 trong luồng cơ bản nếu không tìm thấy dữ liệu trong bảng CategoryLience. Hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

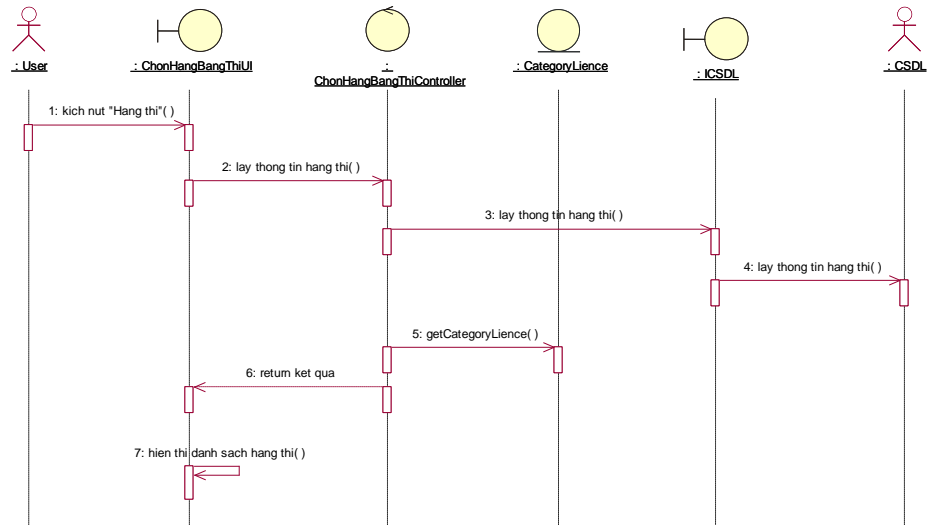
		2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi. Use case kết thúc.
<b>Các yêu cầu đặc biệt</b>	Không	
<b>Hậu điều kiện</b>	Không	
<b>Điểm mở rộng</b>	Không	

**a, Biểu đồ lớp phân tích**



Hình 2. 12. Biểu đồ VOPC Use case Chọn hạng bằng thi

## b, Biểu đồ trình tự



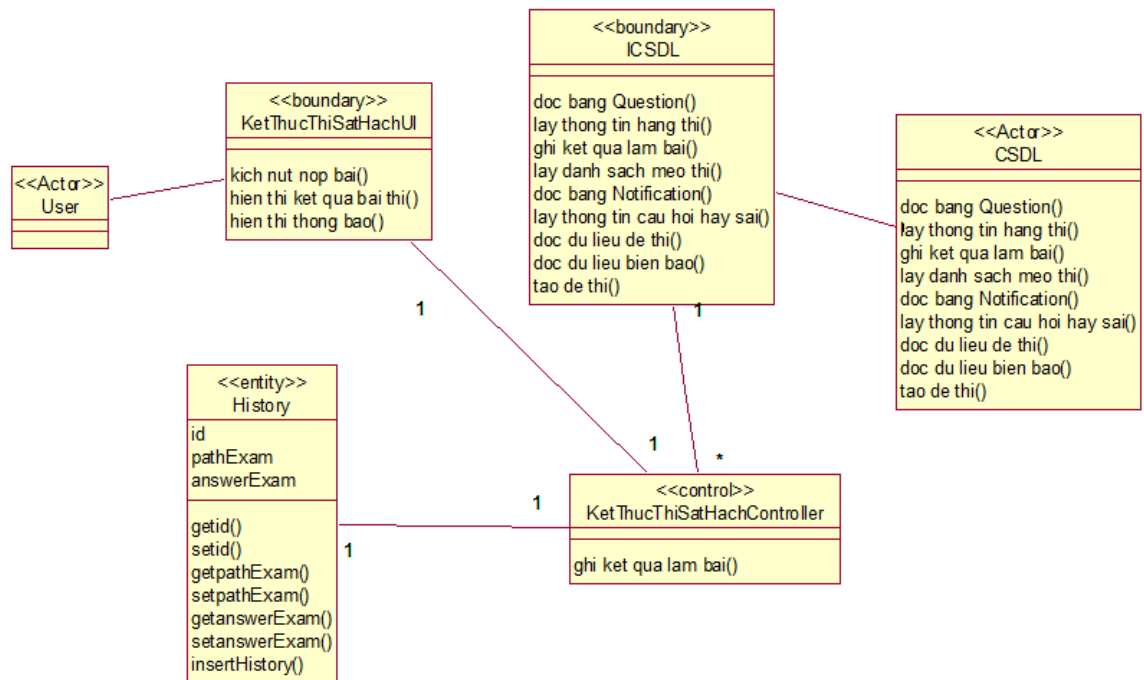
Hình 2. 13. Biểu đồ Basic Flow Use case Chọn hạng bằng thi

### 2.4.2.3. Use case Kết thúc thi sát hạch

Bảng 1. 3. Mô tả Use case Kết thúc thi sát hạch

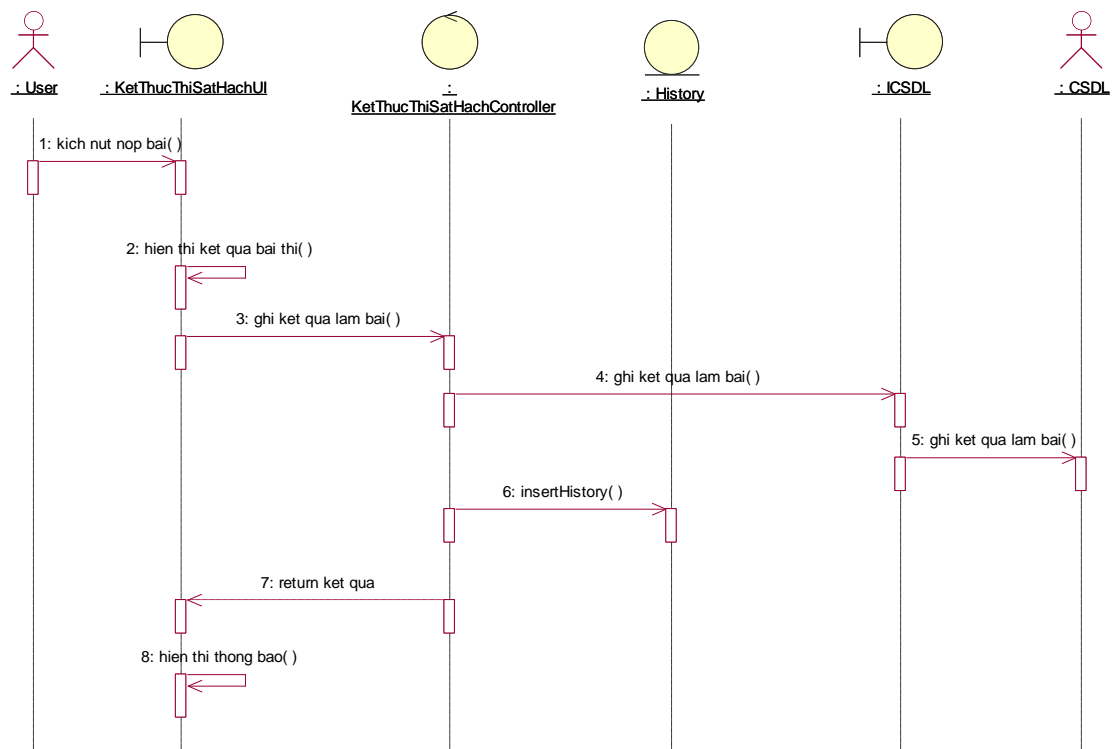
<b>Mô tả</b>		Use case này cho phép người dùng kết thúc bài thi sát hạch.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	1. Use case này bắt đầu khi người dùng kích vào menu và chọn nộp bài ở trên màn hình của hệ thống. Hệ thống sẽ tính toán kết quả làm bài và hiển thị lên trên màn hình đồng thời sẽ ghi lại kết quả vào bảng History. Use case kết thúc.
	<b>Luồng rẽ nhánh</b>	1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi. Use case kết thúc.
<b>Các yêu cầu đặc biệt</b>		Không
<b>Hậu điều kiện</b>		Không
<b>Điểm mở rộng</b>		Không

### a, Biểu đồ lớp phân tích



Hình 2. 14. Biểu đồ VOPC Use case Kết thúc thi sát hạch

### b, Biểu đồ trình tự



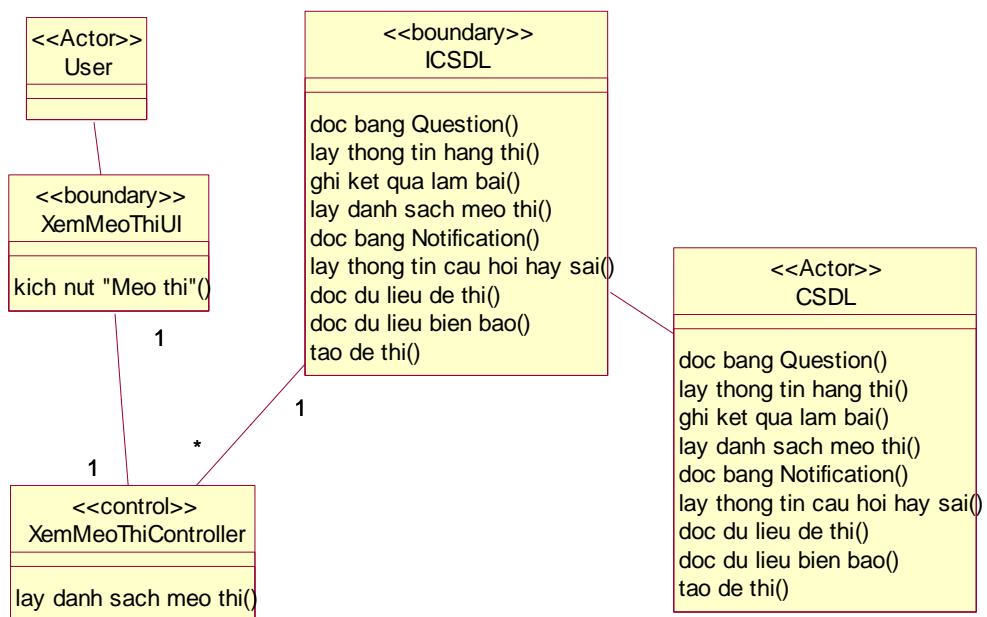
Hình 2. 15. Biểu đồ Basic Flow Use case Kết thúc thi sát hạch

#### 2.4.2.4. Use case Xem mẹo thi

Bảng 1. 4. Mô tả Use case Xem mẹo thi

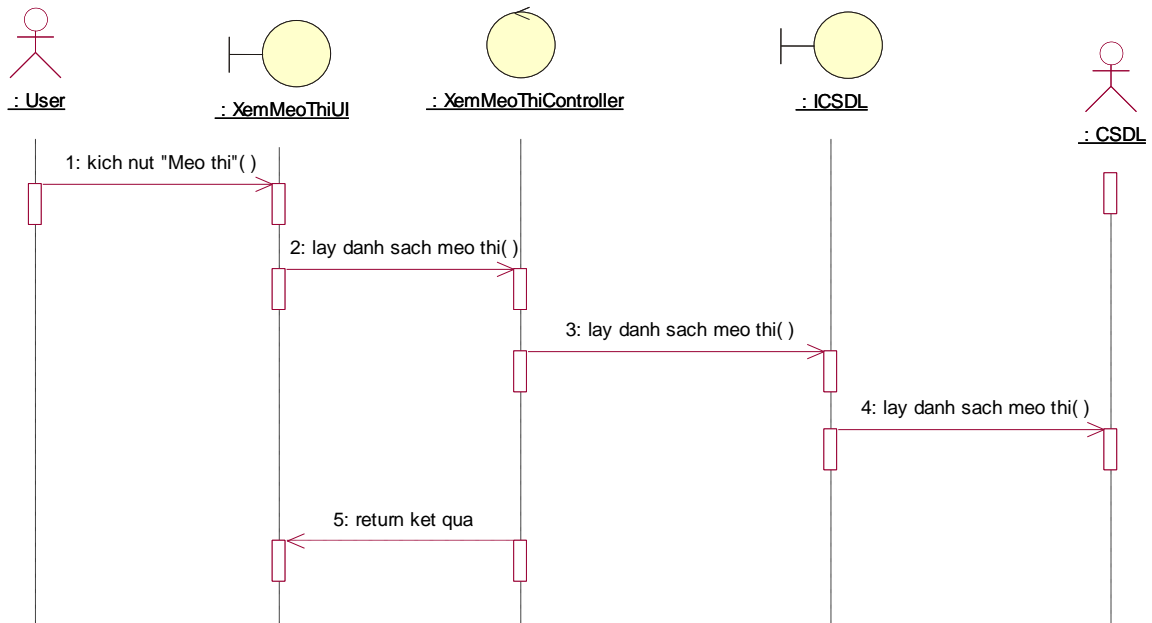
<b>Mô tả</b>		Use case này cho phép người dùng xem các mẹo thi.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	1. Use case này bắt đầu khi người dùng kích vào nút “Mẹo thi” trên màn hình ứng dụng. Hệ thống sẽ đọc dữ liệu từ folder assets của hệ thống Android và hiển thị danh sách các mẹo thi lên màn hình. Use case kết thúc.
	<b>Luồng rẽ nhánh</b>	Không
<b>Các yêu cầu đặc biệt</b>		Không
<b>Hậu điều kiện</b>		Không
<b>Điểm mở rộng</b>		Không

##### a, Biểu đồ lớp phân tích



Hình 2. 16. Biểu đồ VOPC Use case Xem mẹo thi

## b, Biểu đồ trình tự



Hình 2. 17. Biểu đồ Basic Flow Use case Xem meo thi

### 2.4.2.5. Use case Tạo thông báo

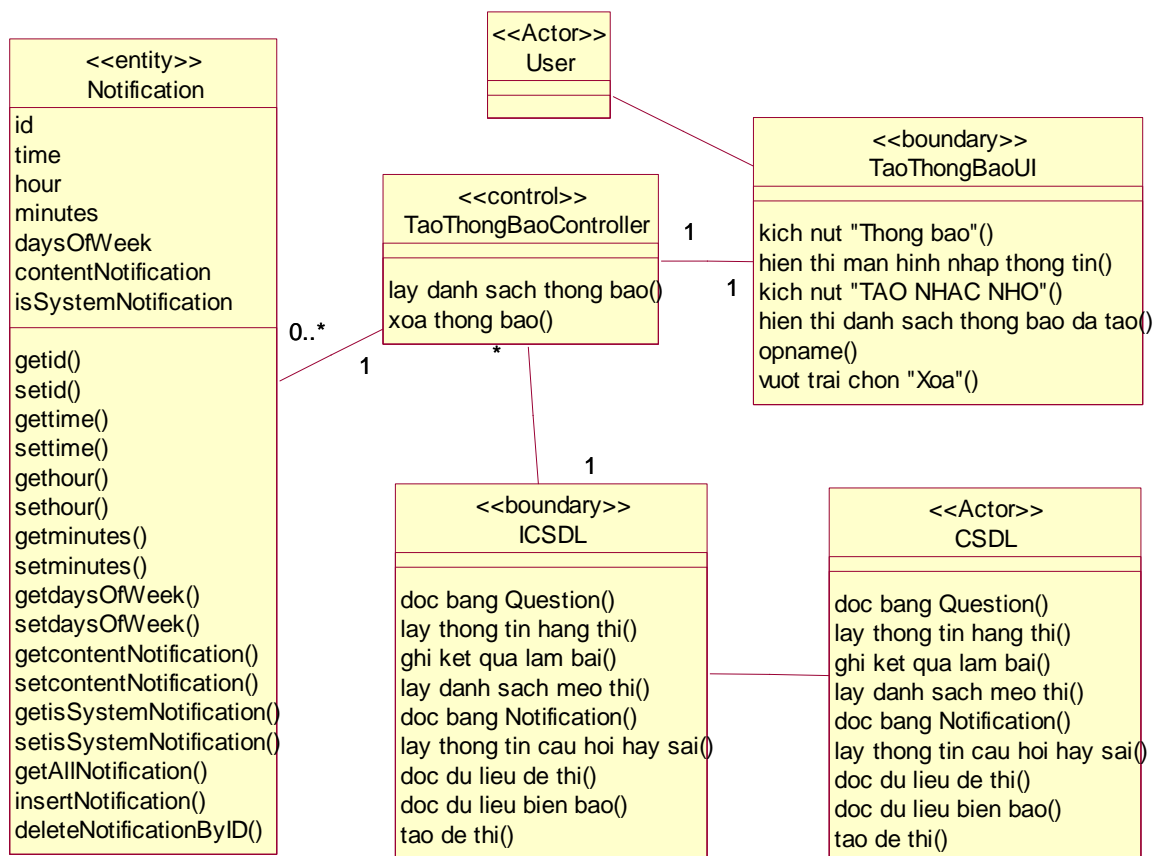
Bảng 1. 5. Mô tả Use case Tạo thông báo

<b>Mô tả</b>		Use case này cho phép người dùng tạo thông báo.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	<p>1. Use case này bắt đầu khi người dùng kích vào nút “Thông báo” trên màn hình ứng dụng. Hệ thống sẽ mở màn hình nhập thông tin bao gồm thời gian, ngày, lời nhắc mà người dùng muốn tạo thông báo.</p> <p>2. Người dùng nhập thông tin tương ứng và kích vào nút “TẠO NHẮC NHỞ” trên màn hình giao diện. Hệ thống sẽ lấy thông tin người dùng nhập và ghi vào bảng Notification và thông báo với thời gian tương ứng. Use case kết thúc.</p>



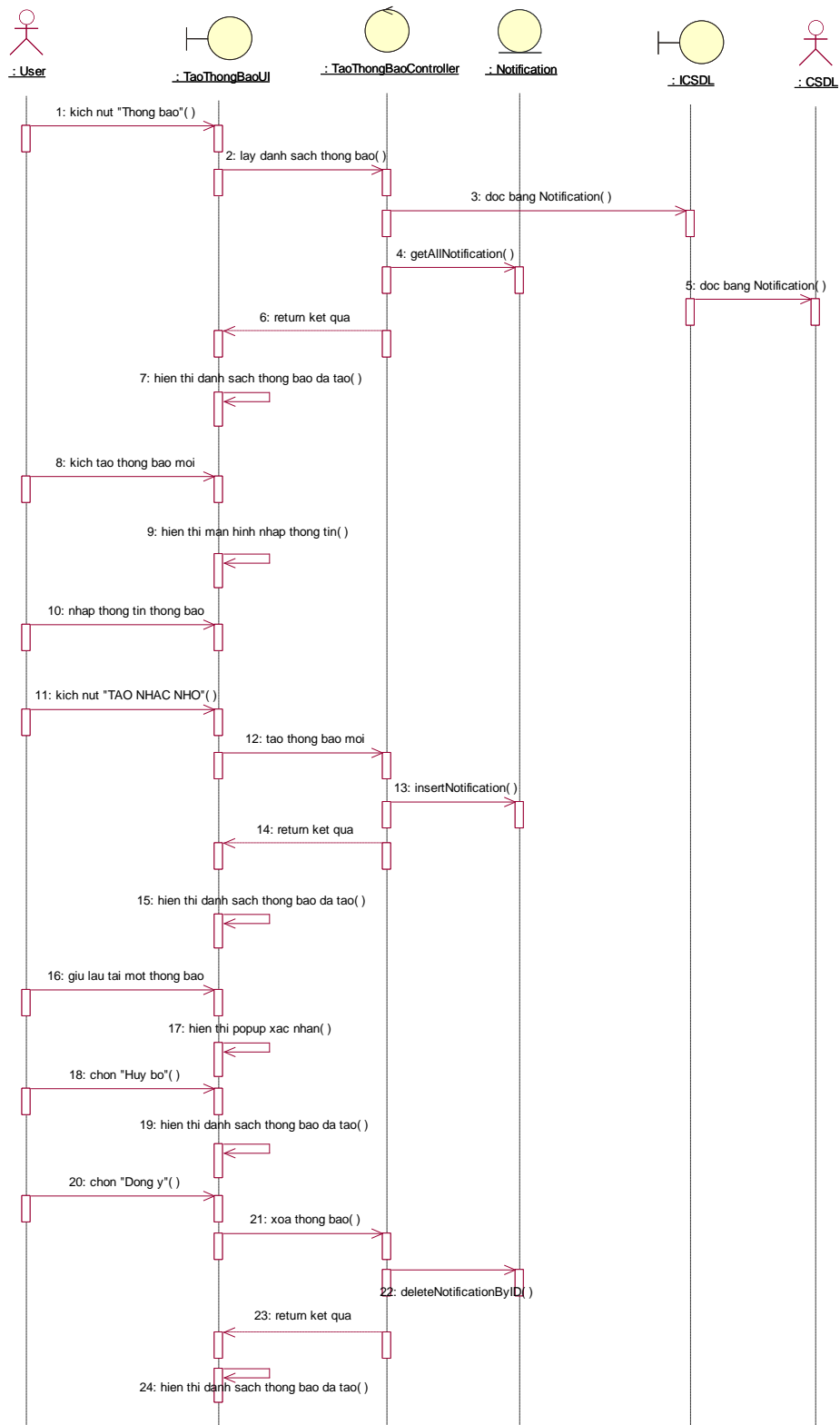
	<b>Luồng rẽ nhánh</b>	<p>1. Tại bước 1 trong luồng cơ bản, nếu người dùng nhập thông tin không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi.</p> <p>2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.</p>
	<b>Các yêu cầu đặc biệt</b>	Không
	<b>Hậu điều kiện</b>	Không
	<b>Điểm mở rộng</b>	Không

#### a, Biểu đồ lớp phân tích



Hình 2. 18. Biểu đồ VOPC Use case Tạo thông báo

## b, Biểu đồ trình tự



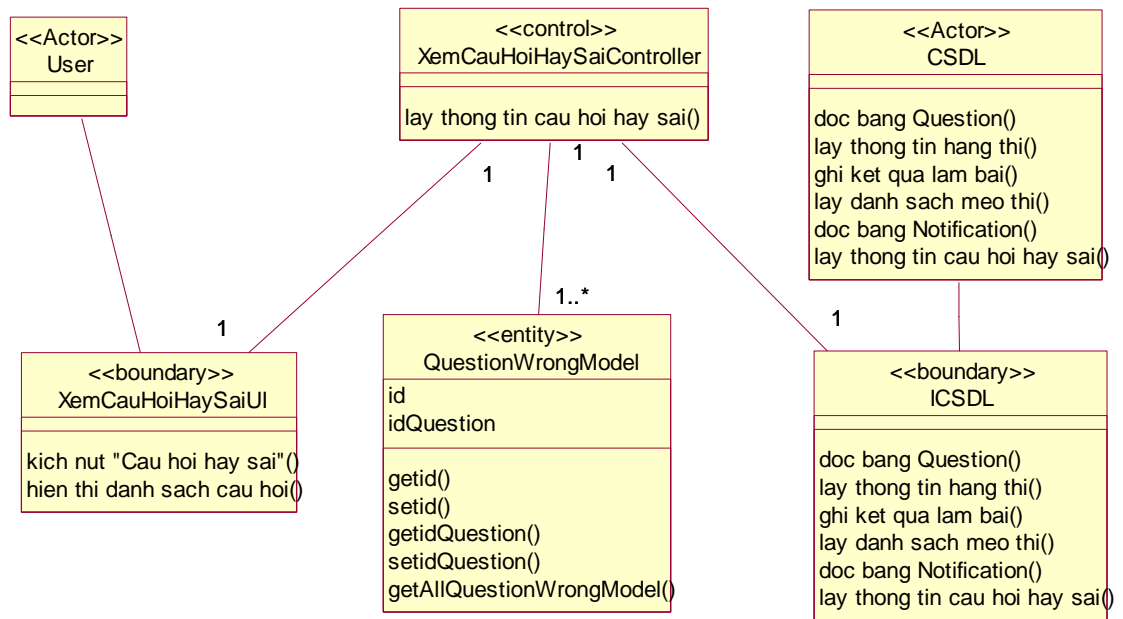
Hình 2. 19. Biểu đồ Basic Flow Use case Tạo thông báo

#### 2.4.2.6. Use case Xem câu hỏi hay sai

*Bảng 1. 6. Mô tả Use case Xem câu hỏi hay sai*

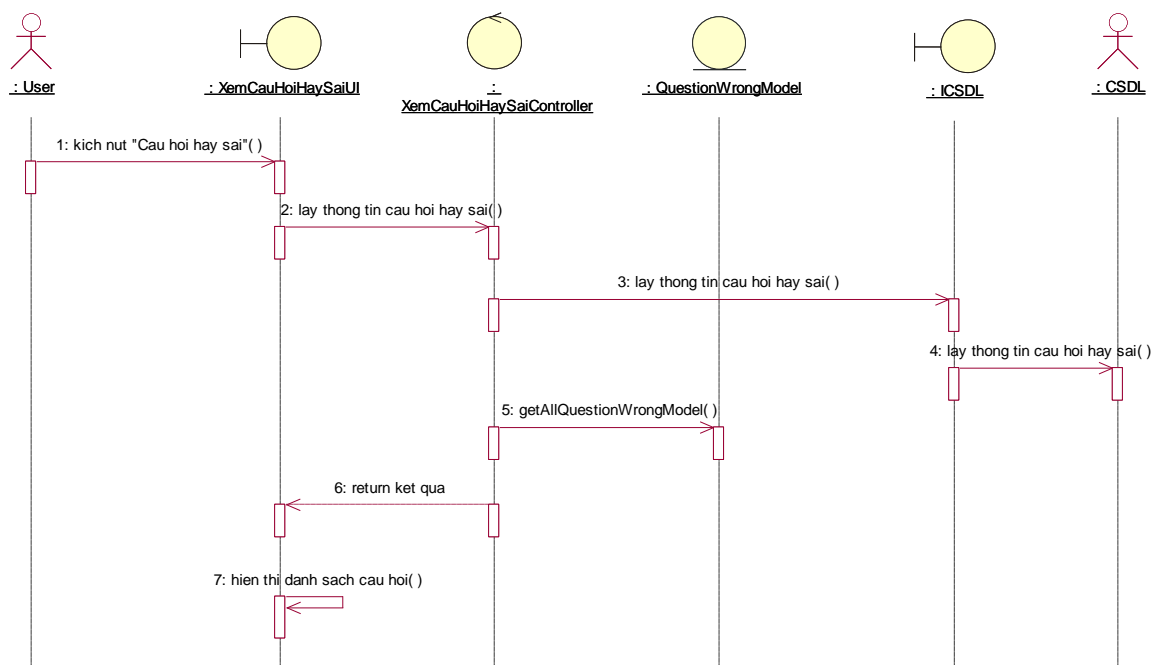
<b>Mô tả</b>		Use case này cho phép người dùng xem những câu hỏi hay sai.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	1. Use case này bắt đầu khi người dùng kích vào nút “Câu hỏi hay sai” trên màn hình giao diện cyra ứng dụng. Hệ thống sẽ lấy thông tin từ bảng QuestionWrongModel và hiển thị danh sách các câu hỏi hay sai lên màn hình. Use case kết thúc.
	<b>Luồng rẽ nhánh</b>	1. Tại bước 1 trong luồng cơ bản, nếu không tìm thấy dữ liệu trong bảng QuestionWrongModel, hệ thống sẽ hiển thị một danh sách trống và use case kết thúc.  2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
<b>Các yêu cầu đặc biệt</b>		Không
<b>Hậu điều kiện</b>		Không
<b>Điểm mở rộng</b>		Không

### a, Biểu đồ lớp phân tích



Hình 2. 20. Biểu đồ VOPC Use case Xem câu hỏi hay sai

### b, Biểu đồ trình tự



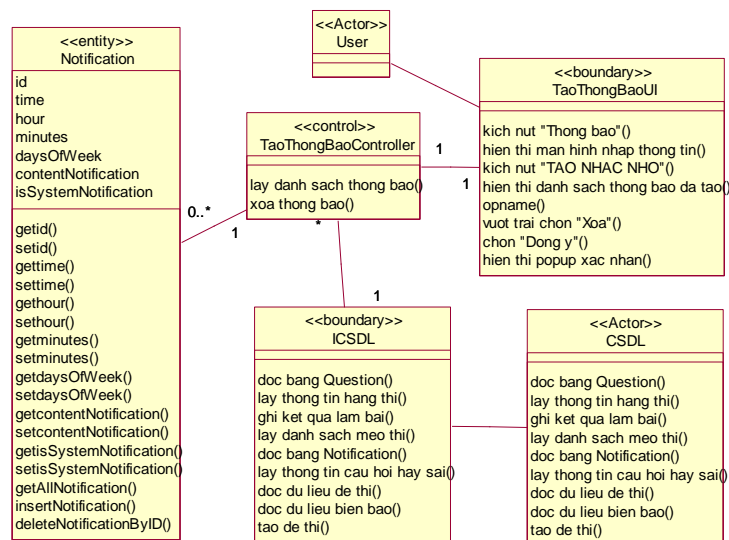
Hình 2. 21. Biểu đồ Basic Flow Use case Xem câu hỏi hay sai

### 2.4.2.7. Use case Học biển báo

Bảng 1. 7. Mô tả Use case Học biển báo

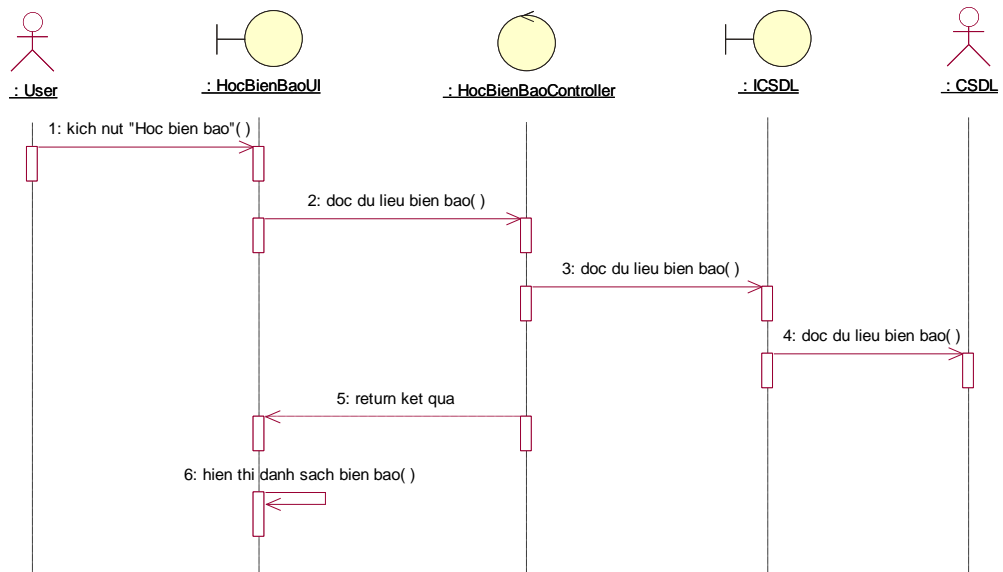
<b>Mô tả</b>		Use case này cho phép người dùng học các biển báo giao thông.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	1. Use case này bắt đầu khi người dùng kích vào nút “Học biển báo” trên màn hình ứng dụng. Hệ thống sẽ đọc dữ liệu từ folder assets của hệ thống Android và hiển thị danh sách biển báo lên màn hình. Use case kết thúc.
	<b>Luồng rẽ nhánh</b>	1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
<b>Các yêu cầu đặc biệt</b>		Không
<b>Hậu điều kiện</b>		Không
<b>Điểm mở rộng</b>		Không

#### a, Biểu đồ lớp phân tích



Hình 2. 22. Biểu đồ VOPC Use case Học biển báo

## b, Biểu đồ trình tự



Hình 2. 23. Biểu đồ Basic Flow Use case Học biển báo

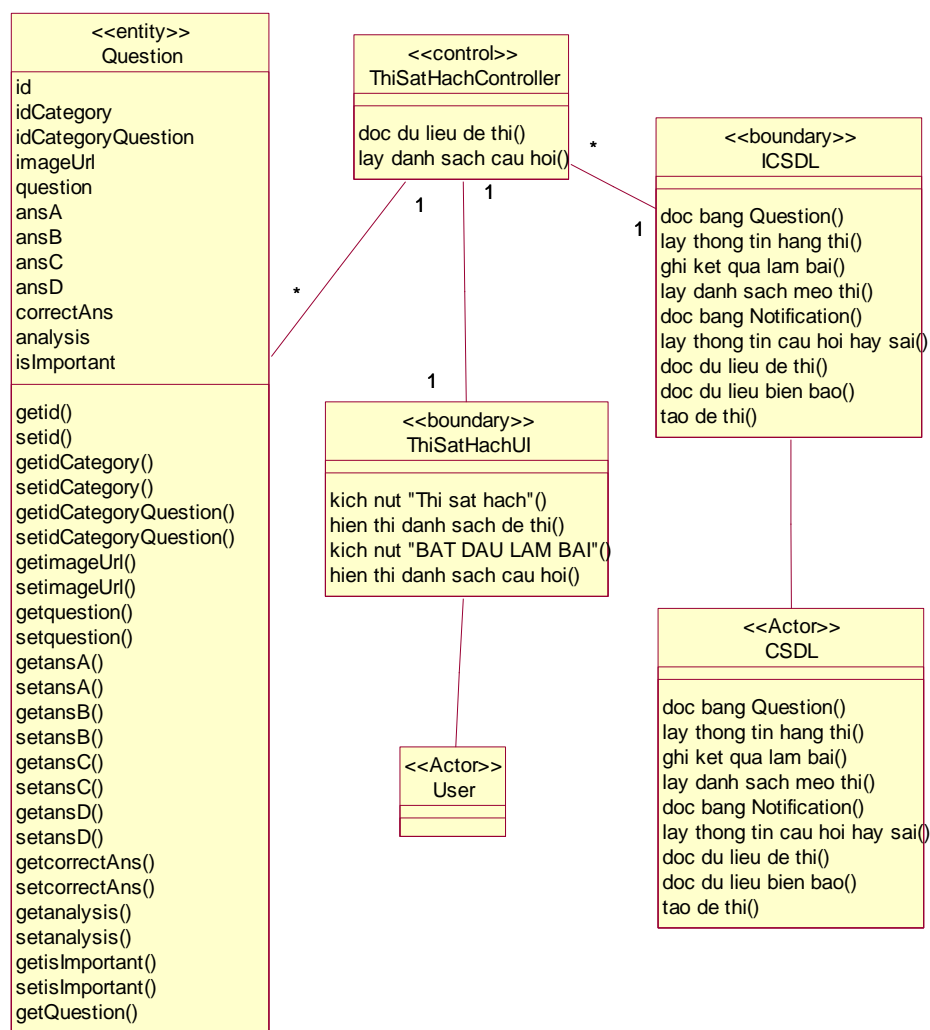
### 2.4.2.8. Use case Thi sát hạch

Bảng 1. 8. Mô tả Use case Thi sát hạch

<b>Mô tả</b>		Use case này cho phép người dùng tham gia thi sát hạch bằng lái xe.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	<p>1. Use case này bắt đầu khi người dùng kích vào nút “Thi sát hạch” trên màn hình ứng dụng. Hệ thống sẽ đọc dữ liệu từ folder assets của hệ thống Android và hiển thị danh sách các đề thi mặc định của ứng dụng lên màn hình.</p> <p>2. Khi người dùng kích vào nút “BẮT ĐẦU LÀM BÀI”. Hệ thống sẽ đọc dữ liệu trong bảng Question và hiển thị danh sách câu hỏi lên màn hình. Use case kết thúc.</p>

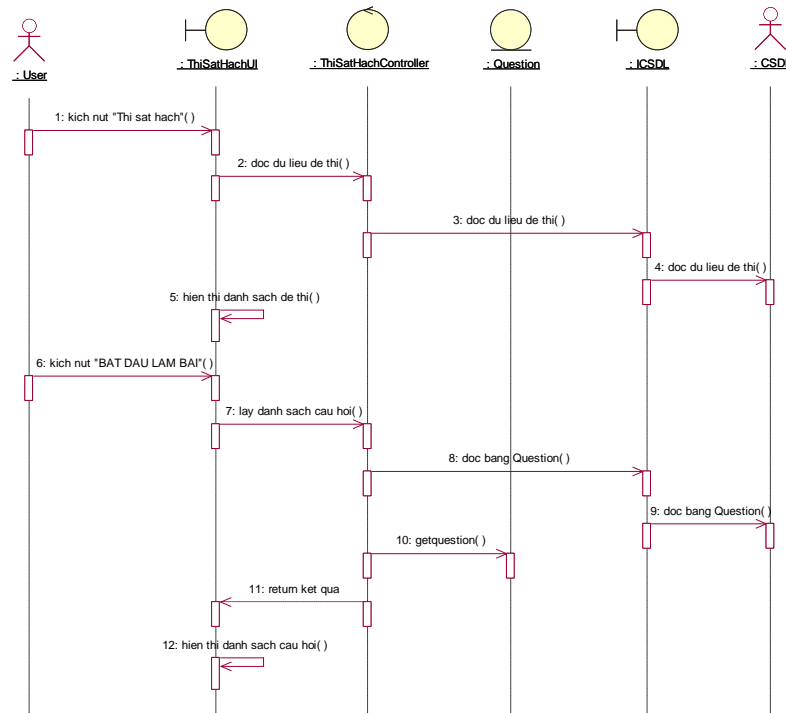
	<b>Luồng rẽ nhánh</b>	<p>1. Tại bước 1 trong luồng cơ bản, nếu tìm thấy dữ liệu, hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p> <p>2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.</p>
	<b>Các yêu cầu đặc biệt</b>	Không
	<b>Hậu điều kiện</b>	Không
	<b>Điểm mở rộng</b>	Không

#### a, Biểu đồ lớp phân tích



Hình 2. 24. Biểu đồ VOPC Use case Thi sát hạch

## b, Biểu đồ trình tự



Hình 2. 25. Biểu đồ Basic Flow Use case Thi sát hạch

### 2.4.2.9. Use case Tạo đề thi thử

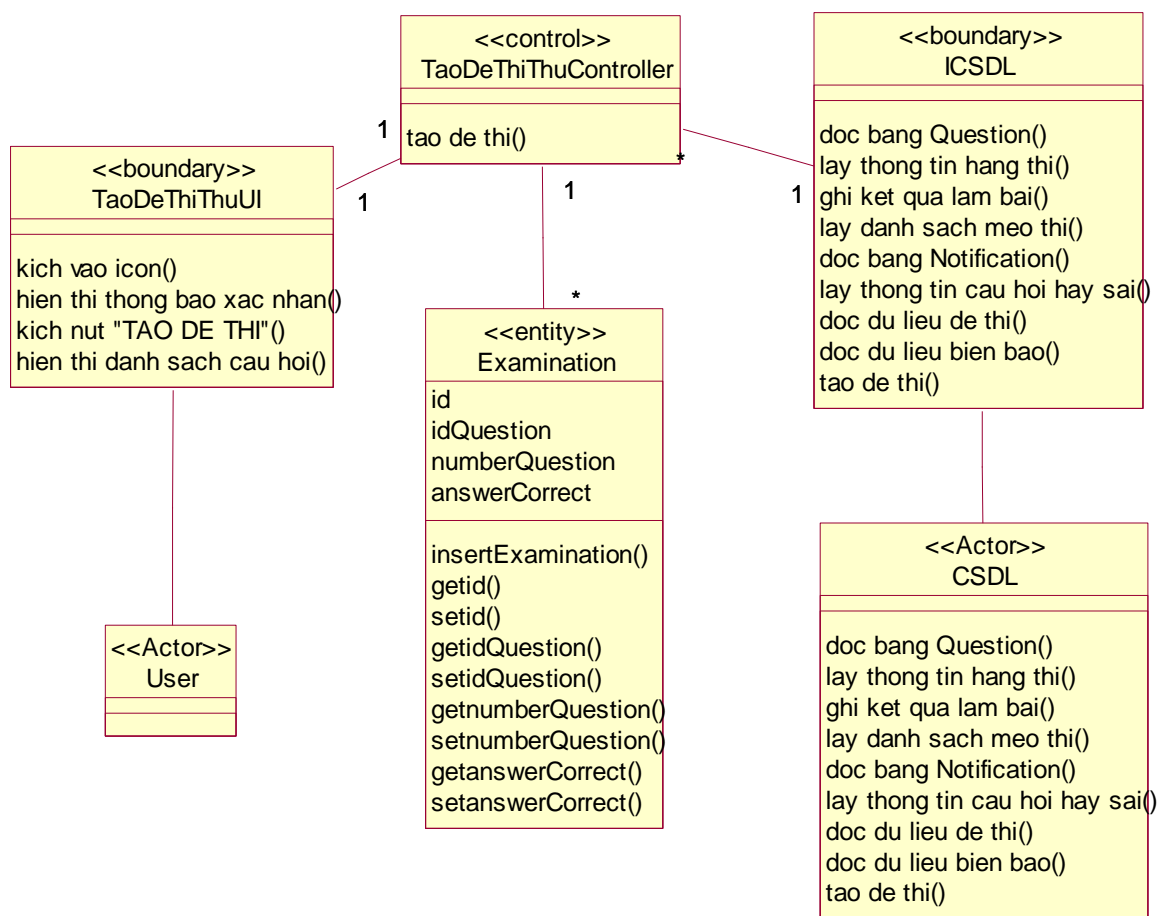
Bảng 1. 9. Mô tả Use case Tạo đề thi thử

<b>Mô tả</b>		Use case này cho phép người dùng tạo các đề thi thử.
<b>Tác nhân</b>		Người dùng
<b>Tiền điều kiện</b>		Không
<b>Luồng sự kiện</b>	<b>Luồng cơ bản</b>	<p>1. Use case này bắt đầu khi người dùng kích vào icon dấu cộng trên thanh menu Thi sát hạch của màn hình ứng dụng. Hệ thống sẽ hiển thị một thông báo xác nhận.</p> <p>2. Khi người dùng kích vào nút “TẠO ĐỀ THI”. Hệ thống sẽ tiến hành tạo một đề thi ngẫu nhiên và lưu vào trong bảng Examination. Use case kết thúc.</p>



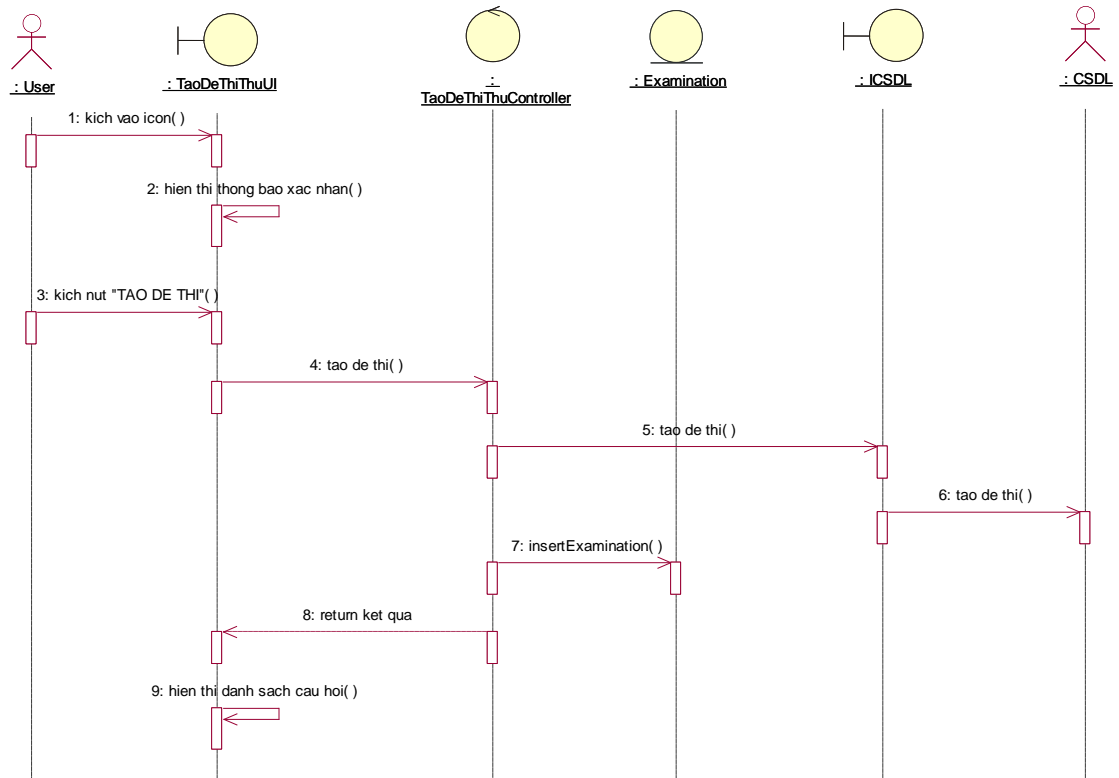
	<b>Luồng rẽ nhánh</b>	1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
	<b>Các yêu cầu đặc biệt</b>	Không
	<b>Hậu điều kiện</b>	Không
	<b>Điểm mở rộng</b>	Không

**a, Biểu đồ lớp phân tích**



*Hình 2. 26. Biểu đồ VOPC Use case Tạo đề thi thử*

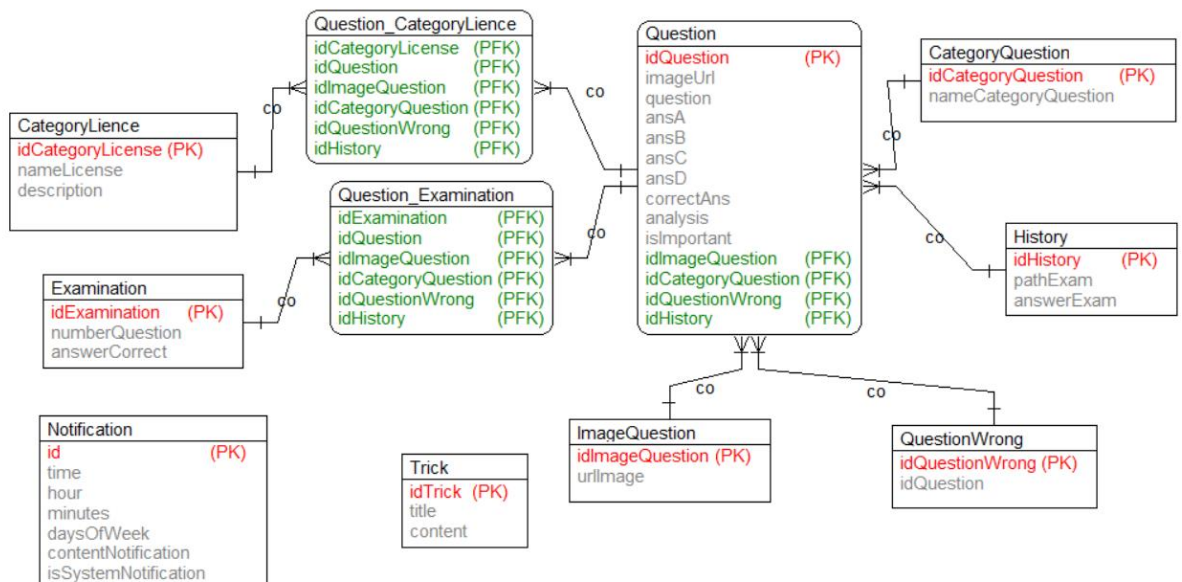
**b, Biểu đồ trình tự**



Hình 2. 27. Biểu đồ Basic Flow Use case tạo đề thi thử

### 2.4.3. Thiết kế cơ sở dữ liệu

#### 2.4.3.1. Biểu đồ thực thể liên kết



Hình 2. 28. Biểu đồ thực thể liên kết

### 2.4.3.2. Thiết kế bảng





#### Bảng CategoryLicense

gplx

Table name: CategoryLicense

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idCategoryLicense	INTEGER								NULL
2	nameLicense	TEXT								NULL
3	description	TEXT								NULL

Bảng 2. 1. Bảng CategoryLicense




#### Bảng CategoryQuestion

gplx

Table name: CategoryQuestion

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idCategoryQuestion	INTEGER								NULL
2	nameCategoryQuestion	TEXT								NULL

Bảng 2. 2. Bảng CategoryQuestion





#### Bảng Examination

gplx

Table name: Examination

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idExamination	INTEGER								NULL
2	numberQuestion	INTEGER								NULL
3	answerCorrect	TEXT								NULL

Bảng 2. 3. Bảng Examination





#### Bảng History

gplx

Table name: History

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idHistory	INTEGER								NULL
2	pathExam	TEXT								NULL
3	answerExam	INTEGER								NULL

Bảng 2. 4. Bảng History




## Bảng ImageQuestion

gplx

Table name: ImageQuestion

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idImageQuestion	INTEGER								NULL
2	urlImage	TEXT								NULL

Bảng 2. 5. Bảng ImageQuestion








## Bảng Notification

gplx

Table name: Notification

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	id	INTEGER								NULL
2	time	INTEGER								NULL
3	hour	INTEGER								NULL
4	minutes	INTEGER								NULL
5	daysOfWeek	INTEGER								NULL
6	contentNotification	TEXT								NULL
7	isSystemNotification	INTEGER								NULL

Bảng 2. 6. Bảng Notification







## Bảng Question

gplx

Table name: Question

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idQuestion	INTEGER								NULL
2	imageUrl	TEXT								NULL
3	question	TEXT								NULL
4	ansA	TEXT								NULL
5	ansB	TEXT								NULL
6	ansC	TEXT								NULL
7	ansD	TEXT								NULL
8	correctAns	INTEGER								NULL
9	analysis	TEXT								NULL
10	isImportant	INTEGER								NULL

Bảng 2. 7. Bảng Question




## Bảng QuestionWrong

gplx

Table name: QuestionWrong

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idQuestionWrong	INTEGER								NULL
2	idQuestion	INTEGER								NULL

*Bảng 2. 8. Bảng QuestionWrong*





## Bảng Trick

gplx

Table name: Trick

☐ WITHOUT ROWID

☐ STRICT

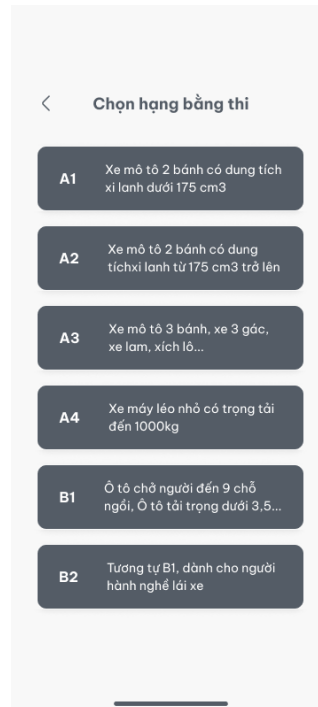
	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	idTrick	INTEGER								NULL
2	title	TEXT								NULL
3	content	TEXT								NULL

*Bảng 2. 9. Bảng Trick*

## CHƯƠNG 3: THIẾT KẾ GIAO DIỆN, CÀI ĐẶT VÀ KIỂM THỬ

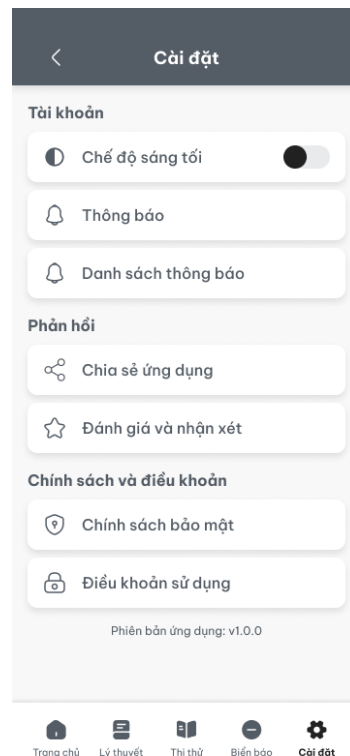
### 3.1. Thiết kế giao diện

#### 3.1.1. Màn hình chọn hạng bằng thi



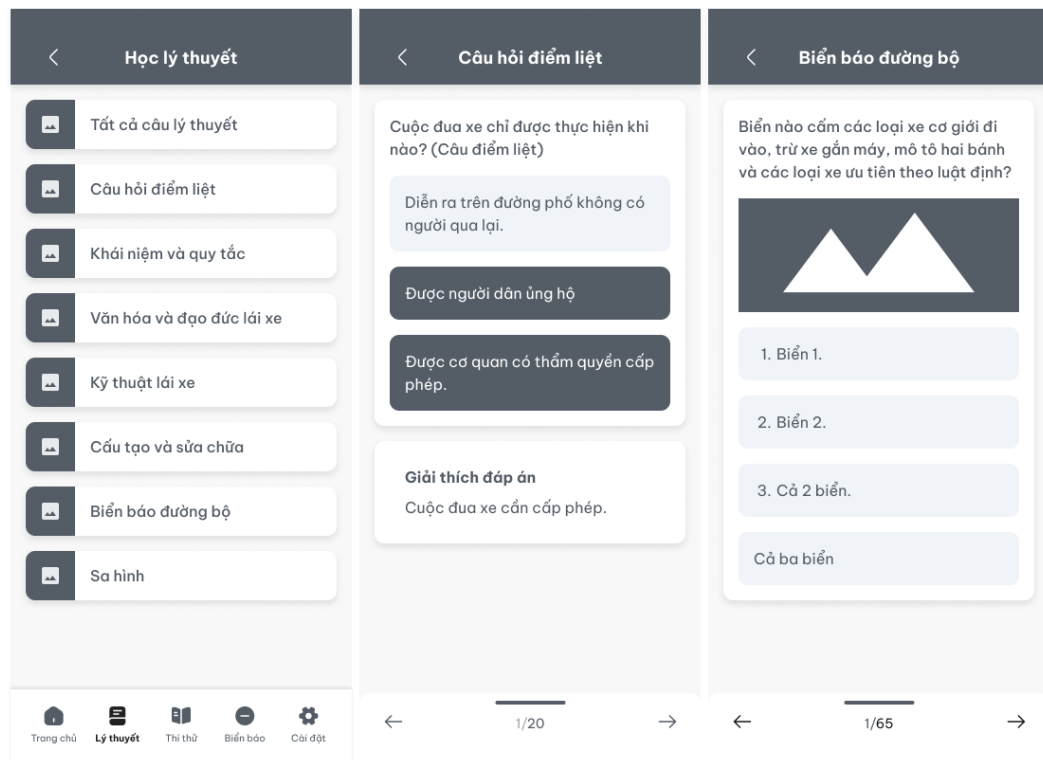
Hình 3. 1. Wireframe màn hình Chọn hạng bằng thi

#### 3.1.2. Màn hình Cài đặt



Hình 3. 2. Wireframe màn hình Cài đặt

### 3.1.3. Màn hình Học lý thuyết



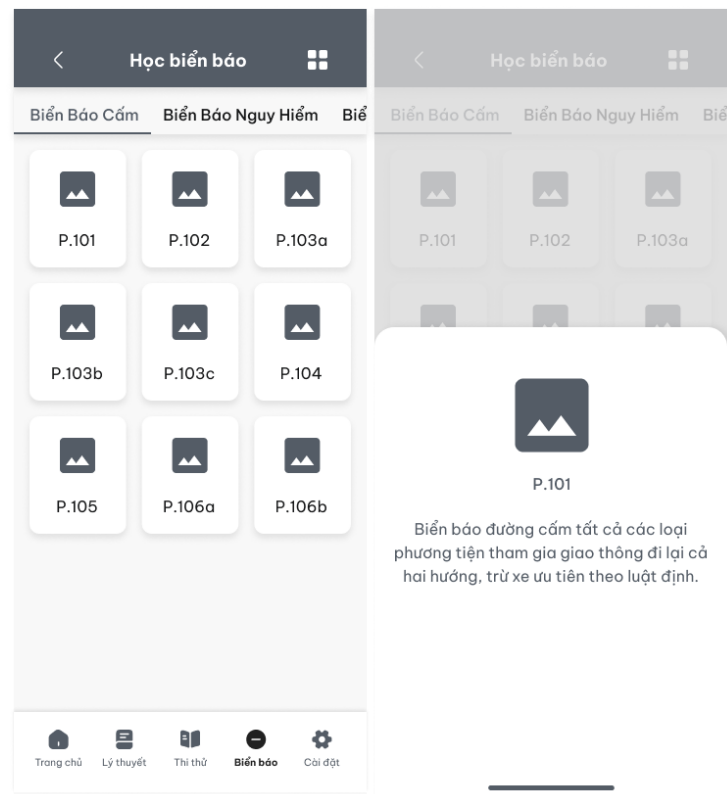
Hình 3. 3. Wireframe màn hình Học lý thuyết

### 3.1.4. Màn hình Xem mẹo thi



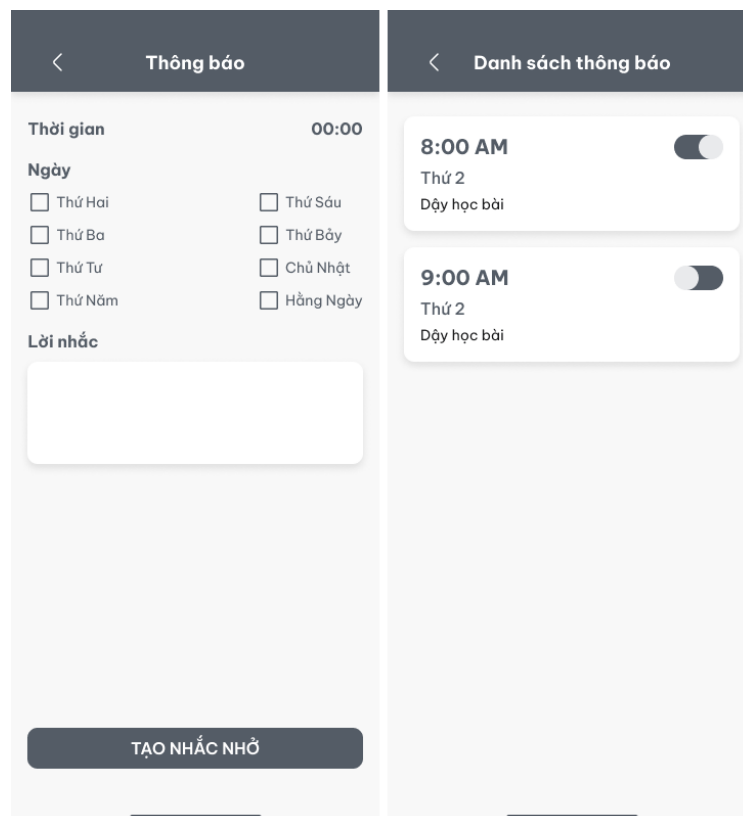
Hình 3. 4. Wireframe màn hình Xem mẹo thi

### 3.1.5. Màn hình Học biển báo



Hình 3. 5. Wireframe màn hình Học biển báo

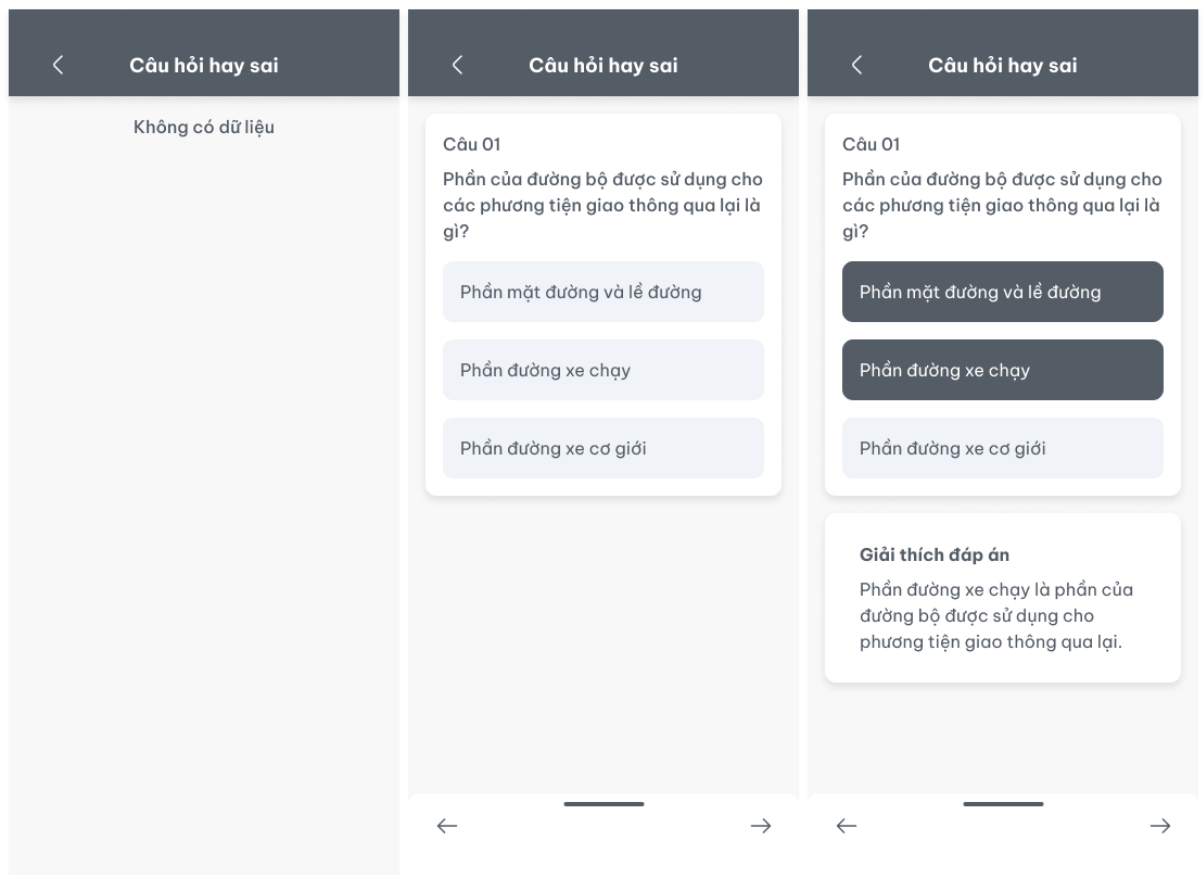
### 3.1.6. Màn hình Tạo thông báo



Hình 3. 6. Wireframe màn hình Tạo thông báo

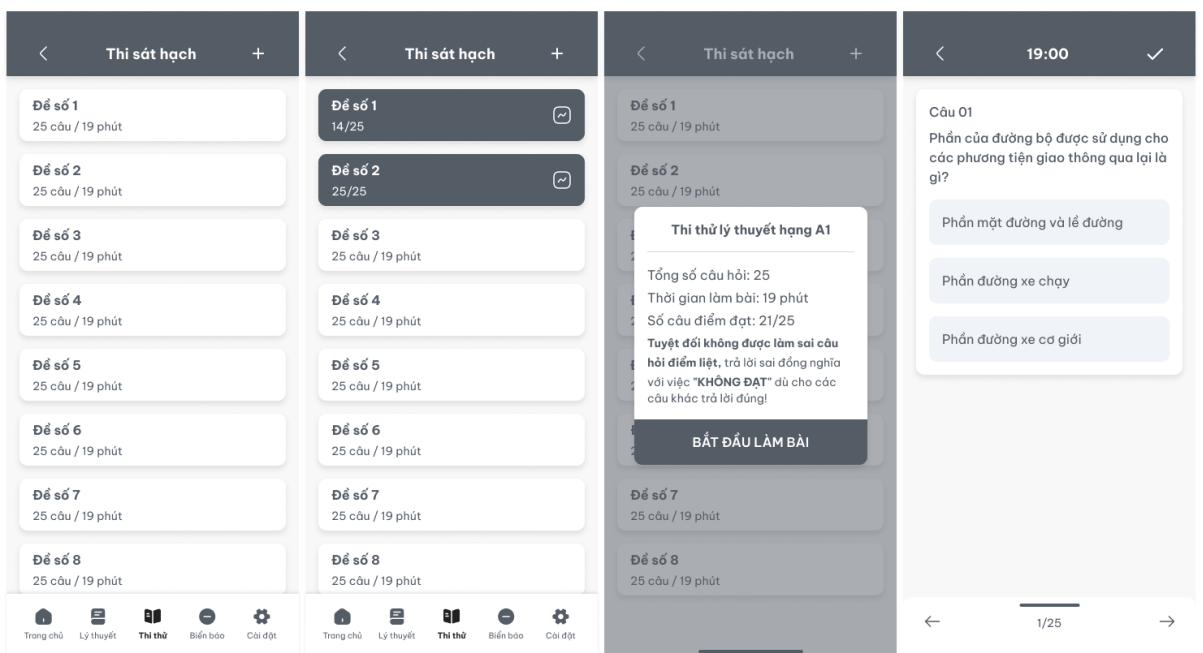


### 3.1.7. Màn hình Xem câu hỏi hay sai



Hình 3. 7. Wireframe màn hình Xem câu hỏi hay sai

### 3.1.8. Màn hình Thi sát hạch



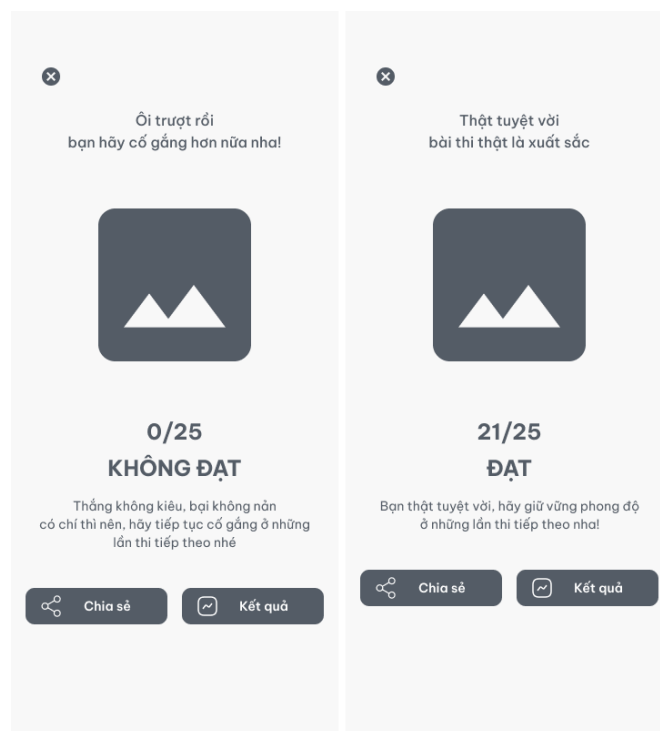
Hình 3. 8. Wireframe màn hình Thi sát hạch

### 3.1.9. Màn hình Kết thúc thi sát hạch



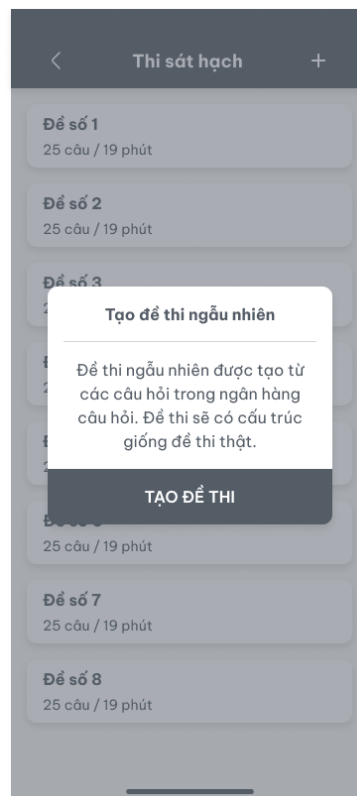
Hình 3. 9. Wireframe màn hình Kết thúc thi sát hạch

### 3.1.10. Màn hình Kết quả



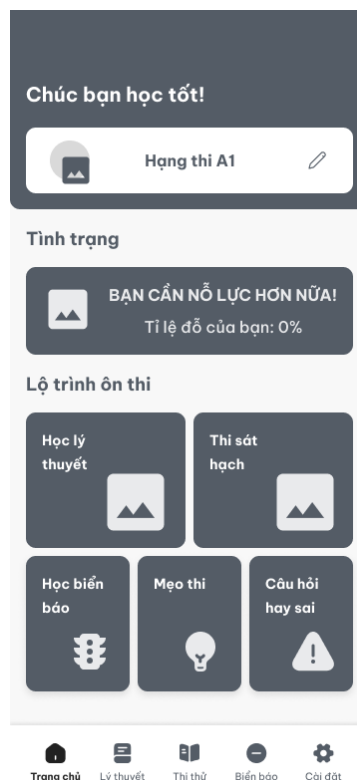
Hình 3. 10. Wireframe màn hình Kết quả

### 3.1.11. Màn hình Tạo đề thi thử



Hình 3. 11. Wireframe màn hình Tạo đề thi thử

### 3.1.12. Màn hình Trang chủ



Hình 3. 12. Wireframe màn hình Trang chủ

## 3.2. Cài đặt chương trình

### 3.2.1. Yêu cầu thiết bị

**Hệ điều hành:** Thiết bị chạy hệ điều hành Android hoặc có khả năng chạy tệp đóng gói đuôi apk.

**Phiên bản hệ điều hành:** Yêu cầu phiên bản Android 11 trở lên để có trải nghiệm tốt nhất.

**Dung lượng lưu trữ:** Cần trống ít nhất 100MB.

**Dung lượng Ram:** Yêu cầu trang bị tối thiểu 4GB RAM.

**Lưu ý:** Một vài chức năng nâng cao có thể không hoạt động đối với các thiết bị đặc biệt hoặc không hỗ trợ phần cứng.

### 3.2.2. Tiến hành cài đặt

**Bước 1:** Tải xuống tệp cài đặt có định dạng đuôi apk vào vùng lưu trữ trong máy.

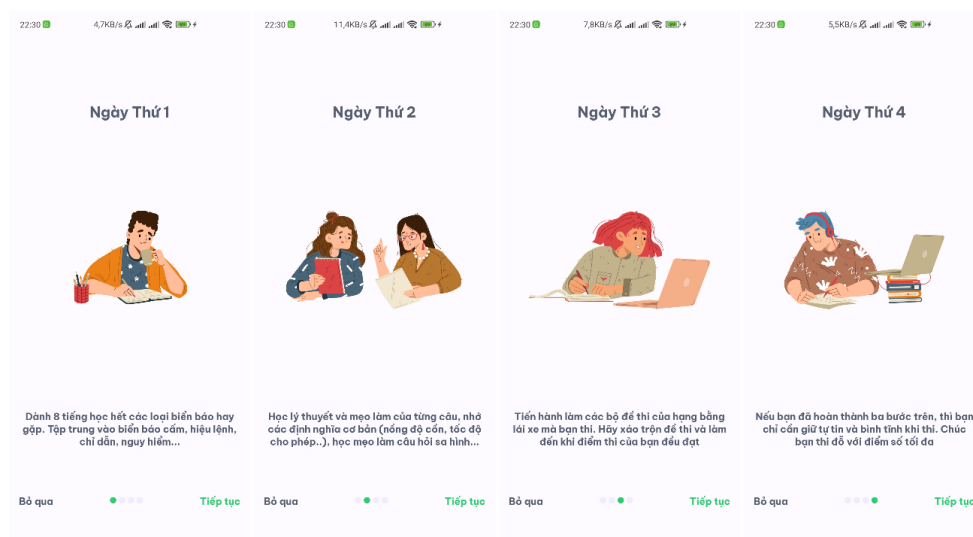
**Bước 2:** Nhấn đúp vào file cài đặt và tiến hành xác nhận cho phép cài đặt ứng dụng từ nguồn cung cấp.

**Bước 3:** Đợi quá trình cài đặt hoàn tất, bạn có thể xóa file cài đặt để tiết kiệm dung lượng bộ nhớ.

**Bước 4:** Ứng dụng tải sẽ hoàn tất và sẵn sàng sử dụng.

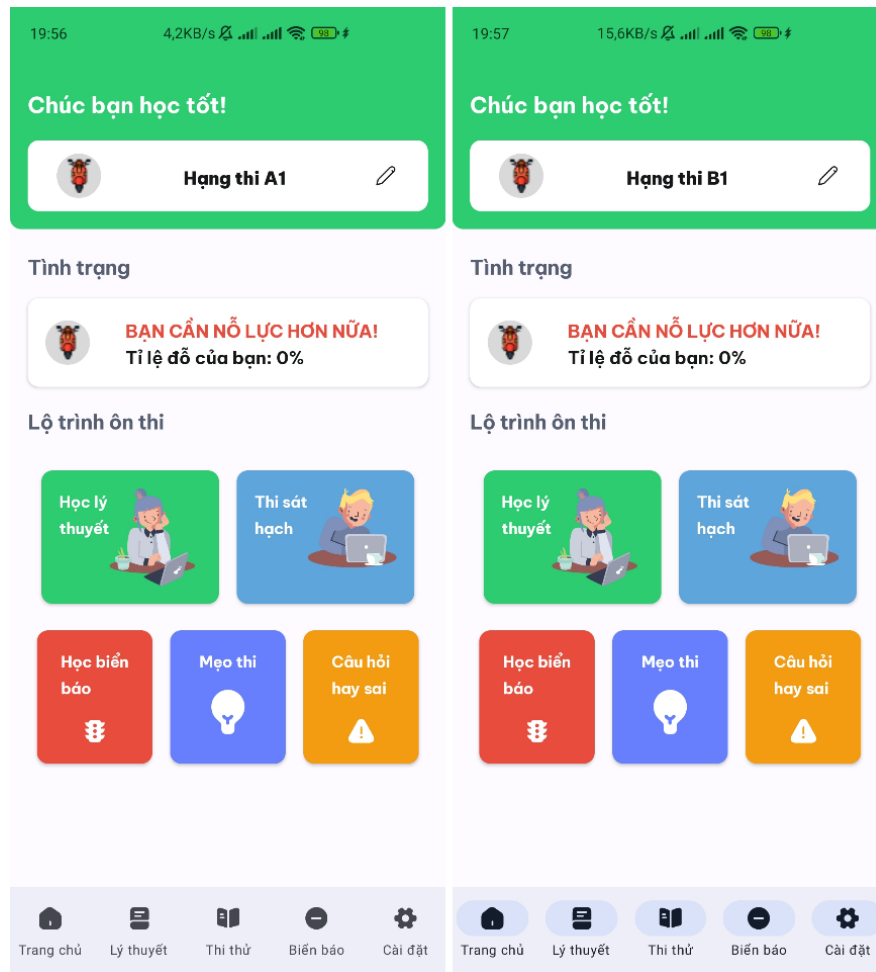
## 3.3. Kết quả đạt được

### 3.3.1. Màn hình Onboarding



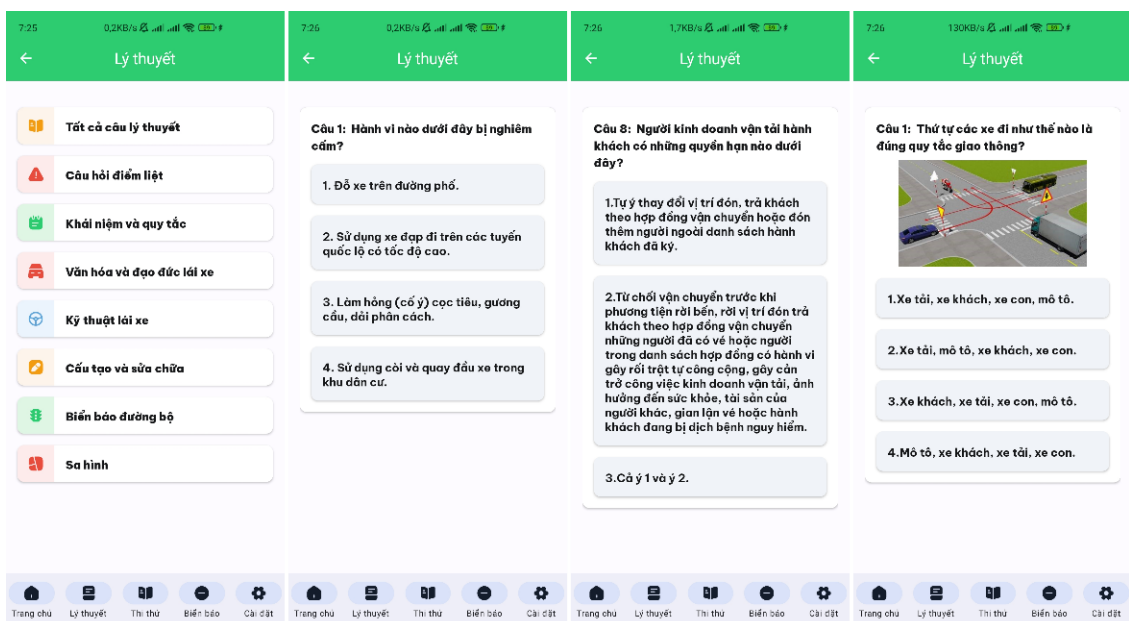
Hình 3. 13. Màn hình giao diện Onboarding

### 3.3.2. Màn hình Trang chủ



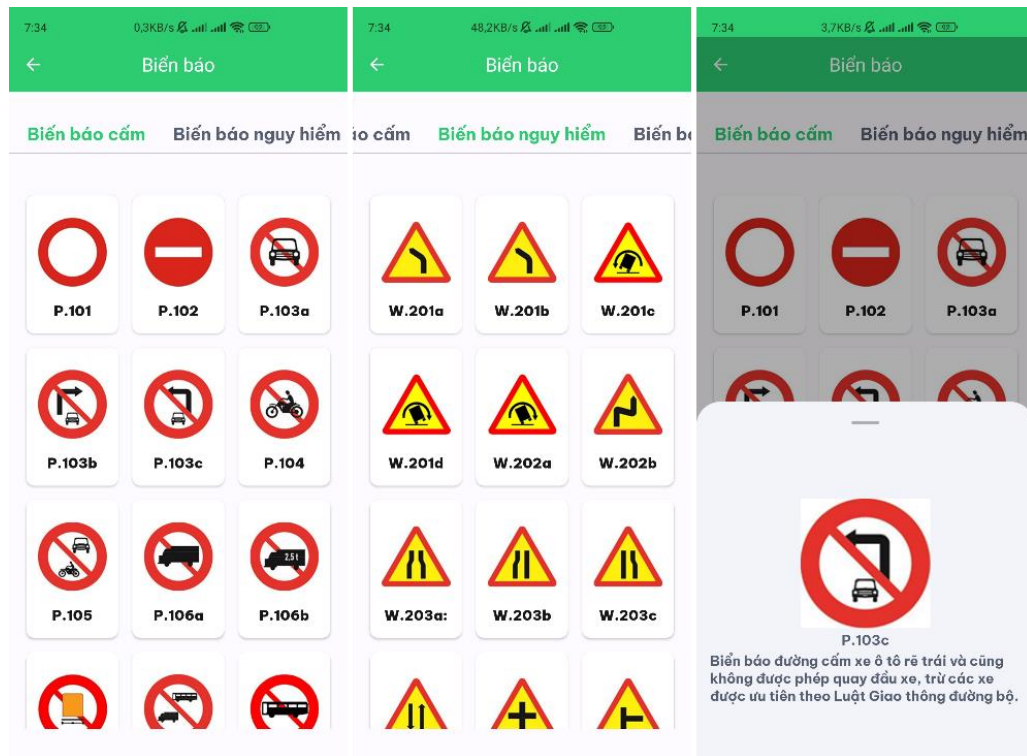
Hình 3. 14. Màn hình giao diện Trang chủ

### 3.3.3. Màn hình Học lý thuyết



Hình 3. 15. Màn hình giao diện Học lý thuyết

### 3.3.4. Màn hình Học biển báo



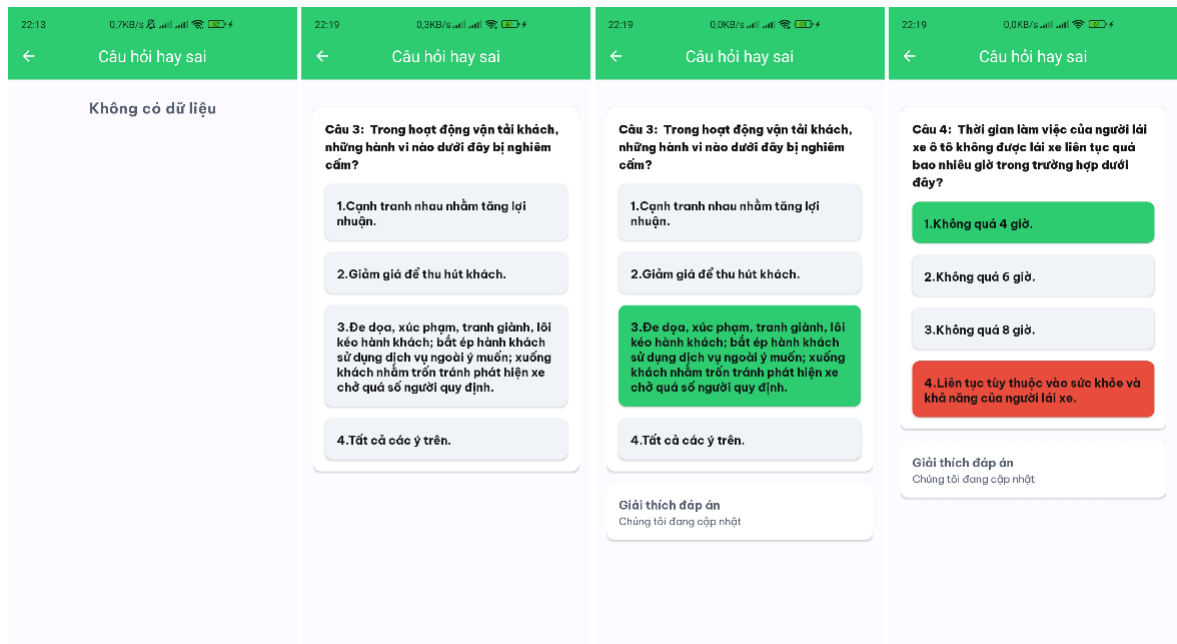
Hình 3. 16. Màn hình giao diện Học biển báo

### 3.3.5. Màn hình Mẹo thi



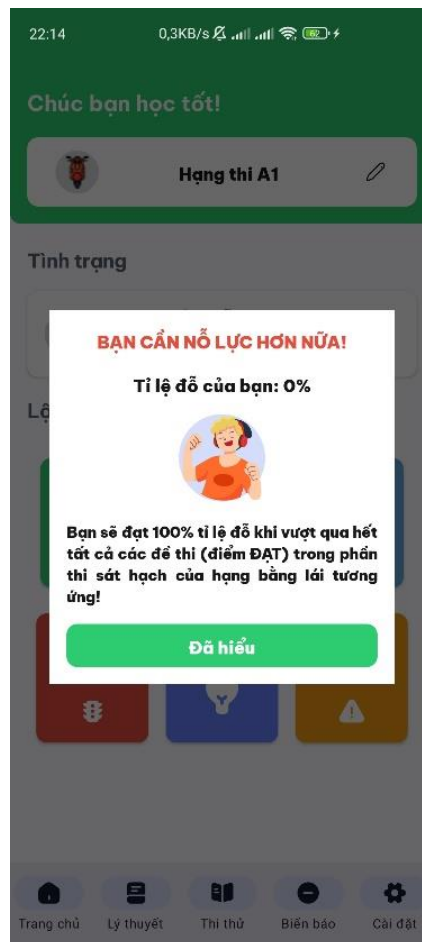
Hình 3. 17. Màn hình giao diện Xem mẹo thi

### 3.3.6. Màn hình Câu hỏi hay sai



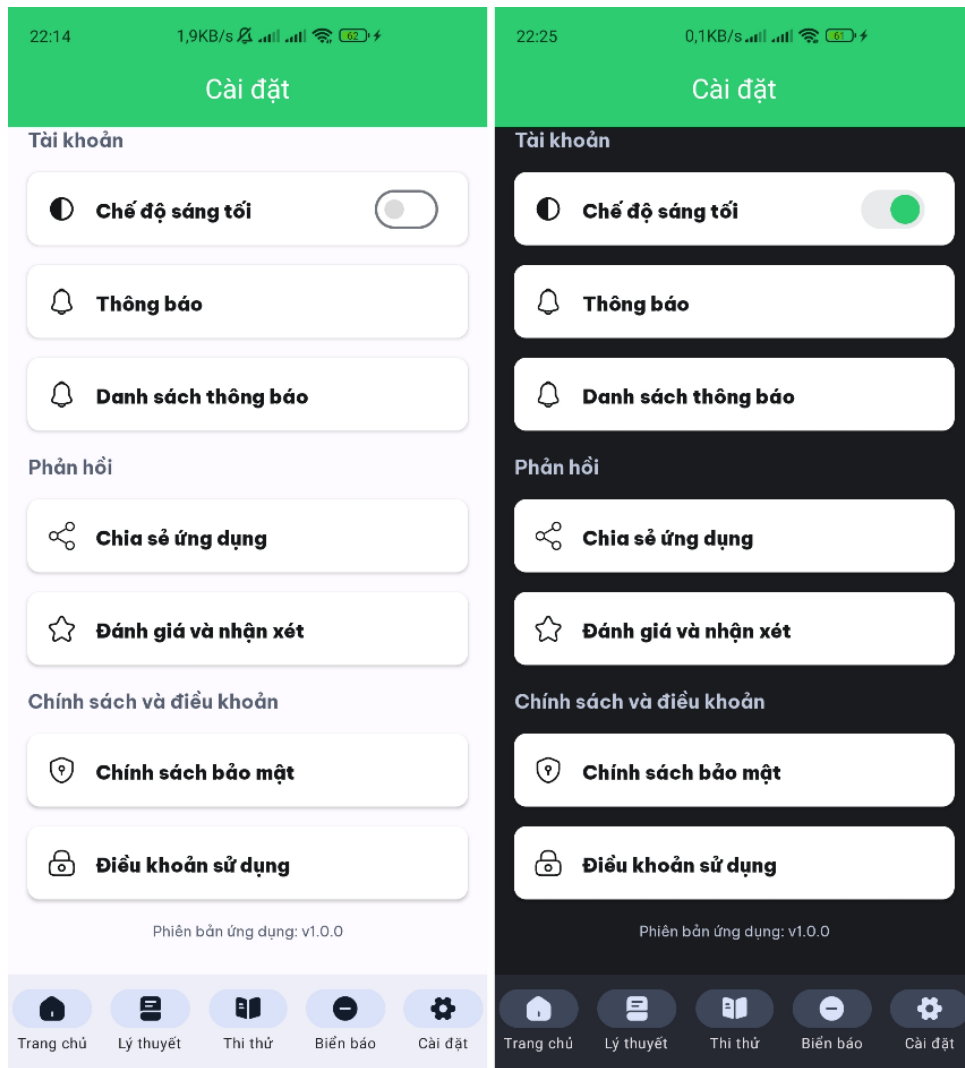
Hình 3. 18. Màn hình giao diện Câu hỏi hay sai

### 3.3.7. Màn hình Xem tỉ lệ đỗ



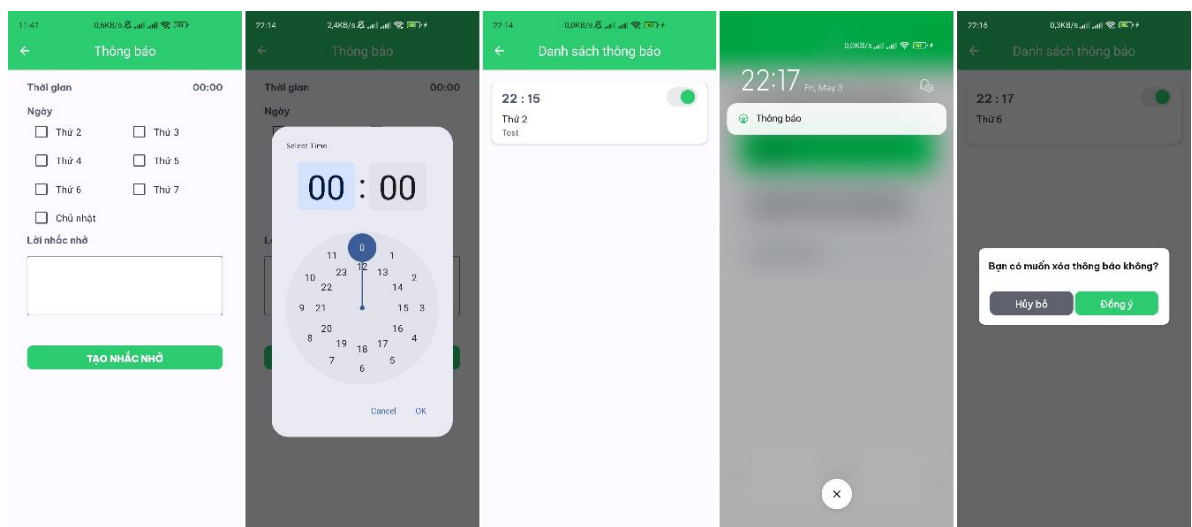
Hình 3. 19. Màn hình giao diện Xem tỉ lệ đỗ

### 3.3.8. Màn hình Cài đặt



Hình 3. 20. Màn hình giao diện Cài đặt

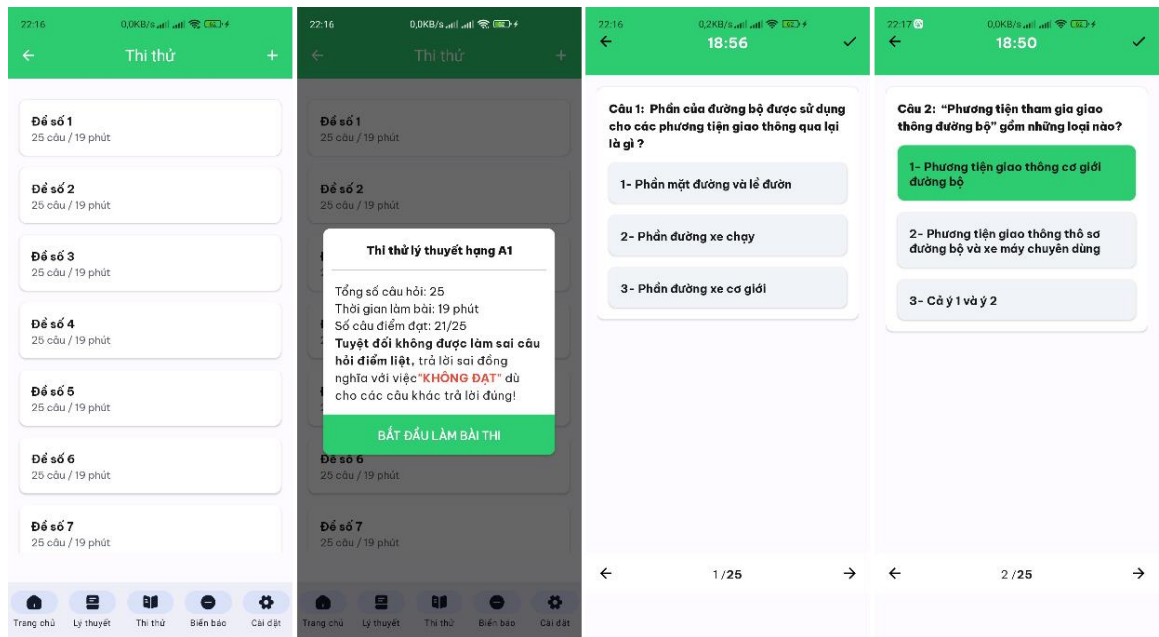
### 3.3.9. Màn hình Thông báo



Hình 3. 21. Màn hình giao diện Thông báo



### 3.3.10. Màn hình Thi sát hạch



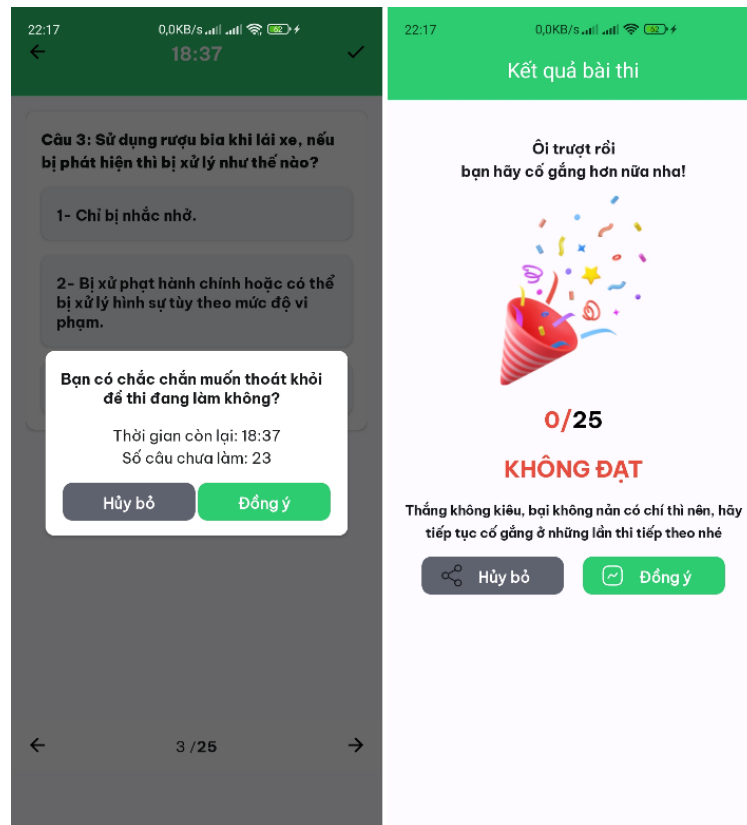
Hình 3. 22. Màn hình giao diện Thi sát hạch

### 3.3.11. Màn hình Tạo đề thi thử



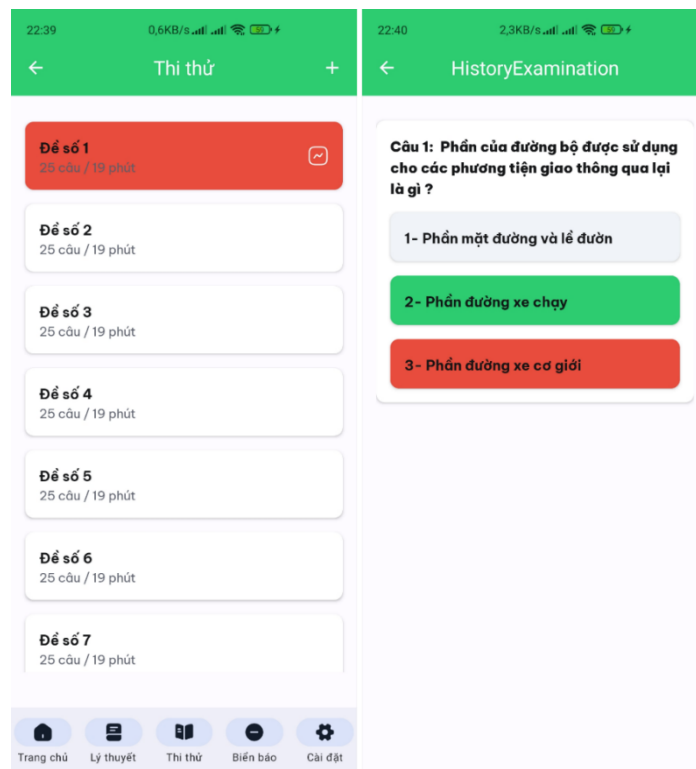
Hình 3. 23. Màn hình giao diện Tạo đề thi thử

### 3.3.12. Màn hình Kết thúc thi sát hạch



Hình 3. 24. Màn hình giao diện Kết thúc thi sát hạch

### 3.3.13. Màn hình Kết quả



Hình 3. 25. Màn hình giao diện Kết quả

### **3.4. Kiểm thử**

#### **3.4.1. Kế hoạch kiểm thử**

##### **3.4.1.1. Mục tiêu**

Kế hoạch kiểm thử giúp xác định xem ứng dụng có hoạt động như mong đợi hay không.

Kiểm thử giúp đảm bảo rằng phần mềm đáp ứng được các yêu cầu về chất lượng, bao gồm hiệu suất, độ tin cậy, bảo mật, và khả năng mở rộng...

Kiểm thử giúp phát hiện lỗi sớm và giảm thiểu số lượng lỗi phát sinh sau khi phần mềm được triển khai, kiểm thử có thể giúp tiết kiệm thời gian và chi phí cho việc sửa lỗi sau này.

##### **3.4.1.2. Các chức năng được đưa vào kiểm thử**

- Chọn hạng bằng thi
- Xem tỉ lệ đỗ
- Học lý thuyết
- Thi sát hạch
- Học biển báo
- Xem mẹo thi
- Xem câu hỏi hay sai
- Tạo đề thi thử
- Kết thúc thi sát hạch
- Tạo thông báo
- Chế độ sáng tối

##### **3.4.1.3. Hướng tiếp cận**

Kiểm thử bằng phương pháp kiểm thử hộp đen. Dưới đây là cách tổ chức kiểm thử và ghi nhận kết quả:

##### **Phân tích chức năng:**

- + Xác định các chức năng chính của ứng dụng cần được kiểm thử.
- + Chia nhỏ các chức năng thành các bước hoạt động cụ thể.

##### **Tạo kịch bản kiểm thử:**

+ Dựa trên phân tích chức năng, tạo ra các kịch bản kiểm thử đơn giản và tổ chức chúng thành các bộ kiểm thử.

+ Mỗi kịch bản tập trung vào một chức năng cụ thể của ứng dụng.

#### **Thực hiện kiểm thử:**

+ Thực hiện các bài kiểm thử theo từng chức năng và ghi lại kết quả.

#### **Ghi nhận kết quả kiểm thử:**

+ Ghi lại kết quả thành công và thất bại qua mỗi bước kiểm thử

+ Đánh giá mức độ ổn định và hiệu suất của ứng dụng trong quá trình kiểm thử.

#### **Phân tích và đánh giá:**

+ Phân tích kết quả kiểm thử để xác định các lỗi và vấn đề gặp phải trong quá trình sử dụng.

+ Đánh giá mức độ đáp ứng của ứng dụng đối với các yêu cầu và mong đợi.

#### **Sửa lỗi và kiểm tra lại:**

+ Sửa các lỗi và vấn đề phát hiện được trong quá trình kiểm thử.

+ Kiểm tra lại tính hoàn thiện và tính ổn định của ứng dụng sau khi sửa lỗi.

#### **Báo cáo và kết luận:**

+ Tạo báo cáo tổng hợp về kết quả kiểm thử, bao gồm các lỗi phát hiện và các biện pháp khắc phục.

+ Kết luận về mức độ hoàn thiện và đáp ứng của ứng dụng đối với yêu cầu kiểm thử.

#### **3.4.1.4. Tiêu chí kiểm thử thành công/thất bại**

Kiểm thử thành công nếu kết quả kiểm thử giống so với kết quả mong muốn đặt ra trước đó.

Kiểm thử thất bại nếu kết quả thực tế của quá trình kiểm thử có sự khác biệt với kết quả mong muốn hoặc không cho ra kết quả.

### 3.4.2. Kịch bản kiểm thử

#### 3.4.2.1. Kiểm thử cơ sở dữ liệu

*Bảng 3. 1. Kiểm thử các bảng trong cơ sở dữ liệu*

ID	Mục đích kiểm thử	Các bước thực hiện	Kết quả mong muốn	Kết quả đạt được
1	Kiểm tra dữ liệu của các trường trong bảng.	1. Kiểm tra các bảng: tên bảng, khóa chính, khóa ngoài, các ràng buộc. 2. Kiểm tra kiểu dữ liệu của các trường trong bảng. 3. Kiểm tra dữ liệu trong các trường.	1. Các bảng có đầy đủ tên, khóa chính, khóa ngoài, các ràng buộc. 2. Dữ liệu trong các bảng đúng, đủ.	PASS
2	Kiểm tra liên kết giữa các bảng	1. Thực hiện kết nối và hiển thị dữ liệu giữa các bảng	Dữ liệu hiển thị đúng và đủ.	PASS

#### 3.4.2.2. Kiểm thử chức năng

*Bảng 3. 2. Kiểm thử chức năng*

ID	Mục đích kiểm thử	Các bước thực hiện	Kết quả mong muốn	Kết quả đạt được
1	Kiểm tra giao diện Onboarding	1. Mở ứng dụng lần đầu tiên, kiểm tra màn hình Onboarding 2. Kiểm tra về thứ tự xuất hiện, màu sắc, thiết kế	1. Thứ tự xuất hiện đúng trình tự 2. Giao diện hiển thị đúng theo thiết kế	FALSE (Giao diện chưa giống so với thiết kế)
2	Kiểm tra giao diện Trang chủ.	1. Mở ứng dụng và dừng tại màn hình Trang chủ. 2. Kiểm tra khả năng bấm vào các mục chức năng tương ứng.	1. Giao diện đúng theo thiết kế 2. Có khả năng chọn vào các mục chức năng ổn định	PASS
3	Kiểm tra chức năng Chọn hạng bằng thi	1. Tại màn hình Trang chủ, kích chọn Hạng thi 2. Kiểm tra sự đầy đủ của các thông tin	1. Các màn hình và nút đều có thể bấm và phản hồi một cách mượt mà 2. Sử dụng	PASS

		trên màn hình như thiết kế. 3. Kiểm tra khả năng kích chọn từng hạng bằng thi tương ứng 4. Kiểm tra hạng bằng thi hiển thị có chính xác với hạng thi được chọn hay không	chung 1 loại font chữ kích thước hợp lí. 3. Kiểm tra không có lỗi chính tả 4. Giao diện giống thiết kế. 5. Hạng bằng thi hiển thị chính xác	
4	Kiểm tra chức năng Học lý thuyết	1. Chọn Học lý thuyết trên màn hình Trang chủ hoặc Lý thuyết trên thanh navigation bar 2. Kiểm tra sự đầy đủ của các thông tin trên màn hình như thiết kế. 3. Kích chọn vào từng danh mục, kiểm tra các trường dữ liệu, số lượng câu hỏi, hình ảnh hiển thị 4. Kiểm tra các màu của câu hỏi tương ứng. Câu sai hiện màu đỏ, câu đúng hiện màu xanh, hiển thị đầy đủ giải thích cho từng câu hỏi.	1. Các màn hình và nút đều có thể bấm và phản hồi một cách mượt mà. 2. Sử dụng chung 1 loại font chữ kích thước hợp lí. 3. Kiểm tra không có lỗi chính tả. 4. Giao diện giống thiết kế. 5. Có đầy đủ các câu hỏi, được hiển thị chính xác theo hạng bằng thi tương ứng. 6. Có đầy đủ giải thích đáp án cho từng câu hỏi. 7. Câu hỏi sa hình hiển thị đúng hình ảnh 8. Khả năng lướt lên xuống một cách mượt mà.	PASS
5	Kiểm tra chức năng Học biển báo	1. Chọn Học biển báo trên màn hình Trang chủ hoặc	1. Danh sách biển báo được hiển thị đầy đủ.	PASS

		Biển báo trên thanh navigation bar. 2. Tiến hành vuốt lên xuống để xem danh sách biển báo. 3. Tiến hành vuốt sang ngang để chuyển đổi giữa các mục. 4. Chọn vào hình ảnh biển báo, kiểm tra thông tin biển báo gồm hình ảnh, tên và mô tả. 5. Kiểm tra giao diện so với thiết kế.	2. Hình ảnh biển báo tương ứng với từng danh mục biển báo. 3. Khả năng vuốt lên xuống, sang trái, sang phải một cách mượt mà. 4. Thông tin biển báo được hiển thị đúng, không có lỗi chính tả. 5. Giao diện giống với thiết kế đã tạo.	
6	Kiểm tra chức năng Xem mẹo thi	1. Chọn Mẹo thi trên giao diện Trang chủ. 2. Tiến hành vuốt lên, xuống để xem danh sách mẹo thi. 3. Kiểm tra thông tin các mẹo thi: chính tả, tiêu đề và mô tả tương ứng.	1. Các mẹo thi được hiển thị đầy đủ và chính xác. 2. Có thể lướt lên, xuống để xem danh sách các mẹo thi.	PASS
7	Kiểm tra chức năng Xem câu hỏi hay sai	1. Chọn Câu hỏi hay sai trên màn hình Trang chủ 2. Kiểm tra giao diện hiển thị khi chưa có câu hỏi hay sai nào 3. Tiến hành học lý thuyết hoặc thi sát hạch và kiểm tra lại câu hỏi hay sai 4. Tiến hành làm lại những câu hỏi hay sai, kiểm tra giao diện hiển thị 5. Kiểm tra về thiết kế	1. Giao diện màn hình được hiển thị đúng thiết kế. 2. Hiển thị danh sách các câu hỏi hay sai	PASS

8	Kiểm tra chức năng Chế độ sáng tối	<ol style="list-style-type: none"> <li>1. Chọn Cài đặt trên thanh navigation bar</li> <li>2. Bật, tắt chế độ sáng tối</li> <li>3. Kiểm tra giao diện hiển thị ở chế độ tối</li> </ol>	<ol style="list-style-type: none"> <li>1. Giao diện và chữ hiển thị phù hợp với nền tối</li> </ol>	FALSE (Một số màn hình màu sắc chữ còn khó nhìn khi hiển thị ở nền tối)
9	Kiểm tra chức năng Tạo đề thi thử	<ol style="list-style-type: none"> <li>1. Chọn Học lý thuyết trên màn hình Trang chủ hoặc Lý thuyết trên thanh navigation bar</li> <li>2. Kích vào icon dấu “+”, chọn Tạo đề thi</li> <li>3. Kiểm tra thứ tự đề thi vừa tạo</li> <li>4. Kiểm tra số lượng câu hỏi, màu sắc, giao diện hiển thị</li> </ol>	<ol style="list-style-type: none"> <li>1. Tạo đề thi thử thành công</li> <li>2. Đề thi có đầy đủ các câu hỏi</li> <li>3. Giao diện hiển thị đúng theo thiết kế</li> </ol>	PASS
10	Kiểm tra chức năng Kết thúc thi sát hạch	<ol style="list-style-type: none"> <li>1. Tiến hành chọn một đề thi ngẫu nhiên và tiến hành thi thử</li> <li>2. Kích vào icon dấu check để kết thúc thi sát hạch</li> <li>3. Kích chọn nút “Hủy bỏ” và “Đồng ý”</li> <li>4. Kích chọn “Đồng ý” và kiểm tra giao diện Kết quả bài thi hiển thị</li> </ol>	<ol style="list-style-type: none"> <li>1. Hiện pop-up xác nhận trước khi kết thúc thi sát hạch</li> <li>2. Khi kích vào nút “Hủy bỏ” quay lại màn hình làm bài</li> <li>3. Khi kích vào nút “Đồng ý” hiển thị giao diện Kết quả gồm số câu trả lời đúng, kết quả...</li> </ol>	PASS
11	Kiểm tra chức năng Thi sát hạch	<ol style="list-style-type: none"> <li>1. Chọn Thi sát hạch trên màn hình Trang chủ hoặc Thi</li> </ol>	<ol style="list-style-type: none"> <li>1. Các màn hình và nút đều có thể bấm và phản hồi</li> </ol>	PASS



		<p>thử trên thanh navigation bar</p> <p>2. Chọn một đề thi ngẫu nhiên</p> <p>3. Chọn “BẮT ĐẦU LÀM BÀI THI” khi pop-up hiển thị</p> <p>4. Kiểm tra sự đầy đủ của các trường thông tin trên màn hình như thiết kế.</p> <p>5. Kiểm tra màn hình làm bài thi, đủ số lượng câu hỏi, thời gian chạy ngược</p> <p>6. Tiến hành vuốt sang trái và phải để chuyển đổi qua lại giữa các câu hỏi</p> <p>7. Kiểm tra về giao diện hiển thị, lỗi chính tả</p>	<p>một cách mượt mà</p> <p>2. Không có lỗi chính tả ở thông tin các câu hỏi và đáp án</p> <p>3. Có đầy đủ hình ảnh ở các câu hỏi biển báo và sa hình</p> <p>4. Có khả năng đếm ngược thời gian về 0 và tự kết thúc bài thi khi hết giờ.</p> <p>5. Nút kết thúc bài làm có tác dụng đối với mọi khoảng thời gian.</p> <p>6. Có khả năng thay đổi câu hỏi dựa vào hai nút điều hướng hoặc vuốt sang trái, sang phải</p> <p>7. Đáp án được chọn có màu xanh, các đáp án còn lại giữ nguyên màu sắc</p> <p>8. Có đầy đủ số lượng câu hỏi so với hạng bằng thi tương ứng</p> <p>9. Giao diện màn hình hiển thị giống so với thiết kế đã tạo</p> <p>10. Hiện thị popup xác nhận trước khi bắt đầu thi sát hạch, tương ứng với</p>	
--	--	--	---	--

			hạng bằng thi đã chọn	
12	Kiểm tra chức năng Tạo thông báo	1. Chọn Cài đặt trên thanh navigation bar 2. Chọn Thông báo 3. Thiết lập thời gian, ngày, lời nhắc nhở và tạo một thông báo mới 4. Kiểm tra các thông tin của thông báo vừa tạo 5. Tiến hành tạo thêm một thông báo mới để xem giao diện danh sách thông báo 6. Xóa một thông báo đã tạo	1. Thông báo được tạo thành công 2. Có khả năng xóa thông báo đã tạo (xóa là nhấn giữ) 3. Thông báo hiển thị rõ ràng và có khả năng đọc được 4. Thông báo được hiển thị đúng với thời gian và ngày đã tạo	PASS
13	Kiểm tra chức năng Xem tỉ lệ đỗ	1. Kiểm tra pop-up xem tỉ lệ đỗ 2. Tiến hành thi thử và kiểm tra lại tỉ lệ đỗ	1. Giao diện hiển thị giống với thiết kế 2. Có tính toán được tỉ lệ đỗ	PASS

### 3.4.3. Kết quả kiểm thử

Tỉ lệ test case đạt (passed): 86,67%

Tỉ lệ test case không đạt (failed): 13,33%

Một số dữ liệu còn chưa được chuẩn xác và đầy đủ, đã tiến hành sửa chữa. Hệ thống hoạt động ổn định các test case manual trên các phiên bản khác nhau.

### 3.5. Định hướng phát triển

Các chức năng cần cải tiến và có thể phát triển thêm:

+ **Giao diện người dùng thân thiện:** Xây dựng một giao diện người dùng thân thiện, dễ dàng sử dụng với người dùng không chỉ trên hệ điều hành Android, mà còn cả trên hệ điều hành iOS.

+ **Nội dung đa dạng và cập nhật:** Cung cấp nội dung ôn thi đa dạng và phong phú, bao gồm các câu hỏi lý thuyết, ví dụ thực tế, và các bài kiểm tra mô

phỏng các kỳ thi thực tế. Đồng thời, cần liên tục cập nhật nội dung mới để phản ánh các thay đổi pháp luật giao thông và bổ sung thông tin mới nhất.

+ **Tích hợp trợ lý ảo:** Sử dụng trí tuệ nhân tạo để cung cấp hỗ trợ và giải đáp câu hỏi của người dùng một cách tự động và nhanh chóng.

+ **Tích hợp các công nghệ mới:** Sử dụng các công nghệ mới như Augmented Reality (AR) hoặc Virtual Reality (VR) để tạo ra một trải nghiệm học tập tương tác và hấp dẫn hơn. Ví dụ, có thể tạo ra các mô phỏng về các tình huống giao thông thực tế trong môi trường VR để người dùng có thể thử nghiệm và rèn luyện kỹ năng quản lý tình huống.

+ **Hỗ trợ đa phương tiện:** Kết hợp nội dung học thông qua nhiều phương tiện như văn bản, hình ảnh, video và âm thanh để tăng cường việc hiểu và nhớ thông tin. Cung cấp giải thích rõ ràng và minh họa bằng hình ảnh hoặc video về các khái niệm phức tạp như quy tắc giao thông hoặc biển báo.

+ **Tích hợp tính năng tùy chỉnh:** Cho phép người dùng tùy chỉnh cài đặt và chế độ hiển thị theo ý muốn của họ, bao gồm cỡ chữ, chế độ ban đêm, màu sắc,... Điều này giúp cải thiện trải nghiệm sử dụng và tạo sự thoải mái cho người dùng.

+ **Liên kết với các đơn vị đào tạo bằng lái xe:** Gợi ý các đơn vị đào tạo bằng lái xe ở gần vị trí của người dùng, là cầu nối kết nối người dùng với các đơn vị đào tạo.

+ **Tích hợp hệ thống hỏi đáp:** Cho phép người dùng đặt câu hỏi và nhận được câu trả lời từ các chuyên gia hoặc cộng đồng người dùng khác thông qua hệ thống hỏi đáp tích hợp trong ứng dụng.

## KẾT LUẬN VÀ KIẾN NGHỊ

Qua thời gian thực hiện đề tài “Xây dựng ứng dụng ôn thi lý thuyết bằng lái xe sử dụng Android Studio” với sự hướng dẫn và góp ý tận tình của cô **ThS. Nguyễn Thị Cẩm Ngoan**, em đã hoàn thành đề tài theo đúng thời gian quy định. Ứng dụng đã xây dựng được những chức năng theo như yêu cầu ban đầu của đề tài, đó là:

- + Chọn hạng bằng thi
- + Xem tỉ lệ đỗ
- + Học lý thuyết
- + Thi sát hạch
- + Học biển báo
- + Xem mẹo thi
- + Xem câu hỏi hay sai
- + Tạo đề thi thử
- + Tạo thông báo
- + Chế độ sáng tối

Do lượng kiến thức còn hạn chế và việc áp dụng công nghệ vào xây dựng ứng dụng vẫn còn nhiều thiếu sót nhưng em sẽ hoàn thiện, sửa chữa và phát triển để có thể đưa ứng dụng vào sử dụng trong thực tế.

### **Kết quả đạt được**

Hoàn thành việc khảo sát, phân tích, thiết kế hệ thống, thiết kế giao diện và hoàn thiện chương trình.

Hoàn thành việc lập trình và cài đặt chương trình.

### **Kiến thức và kinh nghiệm**

Sau thời gian thực hiện đề tài, em đã học hỏi, tiếp thu được rất nhiều kiến thức cả về mặt thực tiễn lẫn kỹ năng lập trình. Thông qua quá trình thực hiện đề tài, em đã có thêm những kiến thức:

- + Hiểu biết thêm về các quy định giao thông.
- + Kỹ năng thiết kế UI/UX.

- + Hiểu biết về lập trình Android bằng ngôn ngữ Kotlin.
- + Kinh nghiệm trong việc sử dụng các công nghệ mới.
- + Khả năng tự học và tự nghiên cứu kiến thức.

### **Ưu điểm**

- + Giao diện người dùng thân thiện, dễ dàng sử dụng.
- + Các câu hỏi có giải thích cụ thể và chi tiết.
- + Hệ thống câu hỏi đa dạng, phong phú.
- + Đa dạng hạng bằng thi để người dùng lựa chọn.
- + Khả năng tính toán tỉ lệ đỗ và tạo thông báo giúp người dùng nâng cao khả năng học tập.

### **Hạn chế**

- + Một số giao diện ở chế độ tối chưa thực sự tối ưu về mặt trải nghiệm.
- + Một số chi tiết trong một vài màn hình chưa thực sự giống so với thiết kế ban đầu.
- + Một số chức năng chưa được tối ưu hóa và còn thiếu một vài khả năng có thể xảy ra trong tương lai.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

- [1]. Thạc Bình Cường (2008), "Bài giảng điện tử môn học: Kiểm thử và đảm bảo chất lượng phần mềm", ĐH Bách Khoa HN.
- [2]. Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan, Giáo trình Phân tích thiết kế hướng đối tượng - Trường Đại học Công nghiệp Hà Nội, NXB Khoa học và Kỹ thuật.
- [3]. Hoàng Quang Huy (2016), "*Giáo trình kiểm thử phần mềm*", Nhà xuất bản Thống kê.
- [4]. Đỗ Ngọc Sơn, Phan Văn Viên, Nguyễn Phương Nga, Giáo trình Hệ quản trị cơ sở dữ liệu - Trường Đại học Công nghiệp Hà Nội, NXB Khoa học và Kỹ thuật.
- [5]. Tổ HTTT Đại học Công Nghiệp Hà Nội, Giáo trình phân tích thiết kế hệ thống.

### Tiếng Anh

- [6]. Jake Knapp, John Zeratsky (2016). "Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days"
- [7]. Steve Krug (2000). "Don't Make Me Think"
- [8]. Ed Burnette, *Hello, Android Introducing Google's Mobile Development Platform*, The Pragmatic Bookshelf, 2010.
- [9]. Donald Norman (1988). "The Design of Everyday Things"
- [10]. <https://developer.android.com/>
- [11]. <https://itnavi.com.vn/blog/kotlin-la-gi-va-uu-diem-cua-ngon-ngu-lap-trinh-kotlin>
- [12]. <https://www.sqlite.org/docs.html>
- [13]. <https://www.behance.net/>