

**BÁO CÁO PHÂN TÍCH VÀ THIẾT KẾ
HỆ THỐNG
PHƯƠNG PHÁP CÓ CẤU TRÚC (SA/SD)**

Hệ thống Báo thức Thông minh với Quiz và QR Code

Ngày 7 tháng 1 năm 2026

Mục lục

1	Phân tích hệ thống	2
1.1	Phương pháp có cấu trúc	2
1.1.1	Biểu đồ phân cấp chức năng	2
1.1.2	Biểu đồ luồng dữ liệu	5
1.1.3	Đặc tả chức năng	12
1.1.4	Phân tích hệ thống về mặt dữ liệu	14
2	Thiết kế	17
2.1	Thiết kế tổng thể	17
2.1.1	Kiến trúc ứng dụng	17
2.1.2	Các module chính	18
2.1.3	Luồng xử lý tổng quát	19
2.2	Giao diện	19
2.2.1	Nhóm màn hình và chức năng	19
2.2.2	Điều hướng (Navigation)	20
2.2.3	Trạng thái (State Management)	21
2.2.4	Validation	21
2.3	Database	22
2.3.1	Thiết kế bảng và quan hệ	22
2.3.2	Ràng buộc toàn vẹn	23
2.3.3	Mapping từ Entity/Room	23
	Giả định/TBD	24

Chương 1

Phân tích hệ thống

1.1 Phương pháp có cấu trúc

Hệ thống Báo thức Thông minh được phân tích theo phương pháp có cấu trúc (Structured Analysis/Design), sử dụng các kỹ thuật: Function Decomposition Diagram (FDD), Data Flow Diagram (DFD), Data Dictionary và Entity-Relationship Diagram (ERD). Phương pháp này tập trung vào việc phân rã chức năng theo cấp bậc và mô tả luồng dữ liệu qua các tiến trình xử lý.

1.1.1 Biểu đồ phân cấp chức năng

Mô tả chung

Biểu đồ phân cấp chức năng (FDD) thể hiện cấu trúc tổng thể của hệ thống bằng cách phân rã các chức năng từ tổng quát đến chi tiết. Hệ thống được chia thành 5 module chính, mỗi module chứa các chức năng con cụ thể.

Sơ đồ FDD

```
1 %% ơS đồ FDD - Function Decomposition Diagram
2 %% ệH ồthng Báo ứthc Thông minh ỡvi Quiz và QR Code
3
4 graph TB
5     Root[ệH ồthng Báo ứthc<br/>Thông minh]
6
7     Root --> M1[1. ảQun lý<br/>Báo ứthc]
8     Root --> M2[2. ảQun lý<br/>ủCh đề & Câu ỏhi]
9     Root --> M3[3. ựThc thi<br/>Báo ứthc]
10    Root --> M4[4. ảQun lý<br/>QR Code]
11    Root --> M5[5. ồThng kê &<br/>Báo cáo]
12
13    %% Module 1: ảQun lý Báo ứthc
14    M1 --> F1_1[1.1 ạTo báo ứthc ỡmi]
15    M1 --> F1_2[1.2 íChnh ửsa báo ứthc]
16    M1 --> F1_3[1.3 Xóa báo ứthc]
17    M1 --> F1_4[1.4 ậBt/ấTt báo ứthc]
18    M1 --> F1_5[1.5 Xem danh sách<br/>báo ứthc]
19    M1 --> F1_6[1.6 ạTo báo ứthc nhanh]
20
21    %% Module 2: ảQun lý ủCh đề & Câu ỏhi
22    M2 --> F2_1[2.1 ảQun lý ủch đề]
23    M2 --> F2_2[2.2 ảQun lý câu ỏhi]
24
25    F2_1 --> F2_1_1[2.1.1 ạTo ủch đềở mi]
26    F2_1 --> F2_1_2[2.1.2 ửSa tên ủch đề]
27    F2_1 --> F2_1_3[2.1.3 Xóa ủch đề]
28    F2_1 --> F2_1_4[2.1.4 Tìm ếkim ủch đề]
29
30    F2_2 --> F2_2_1[2.2.1 Thêm câu ỏhi<br/>vào ủch đề]
31    F2_2 --> F2_2_2[2.2.2 íChnh ửsa<br/>câu ỏhi]
32    F2_2 --> F2_2_3[2.2.3 Xóa câu ỏhi]
33    F2_2 --> F2_2_4[2.2.4 Xem chi ếtít<br/>câu ỏhi]
34
35    %% Module 3: ựThc thi Báo ứthc
36    M3 --> F3_1[3.1 Kích ạhot báo ứthc]
37    M3 --> F3_2[3.2 ểHin ịth giao ệdin<br/>báo ứthc đang reo]
38    M3 --> F3_3[3.3 ựThc ệhin Quiz]
39    M3 --> F3_4[3.4 Quét QR Code<brđể/> ấtt]
40    M3 --> F3_5[3.5 Snooze báo ứthc]
41    M3 --> F3_6[3.6 ấTt báo ứthc]
42
43    F3_3 --> F3_3_1[3.3.1 ọChn câu ỏhi<br/>theo ậthut toán SRS]
44    F3_3 --> F3_3_2[3.3.2 ểHin ịth câu ỏhi<br/>và đếm ờgi]
45    F3_3 --> F3_3_3[3.3.3 ểKim tra đáp án]
46    F3_3 --> F3_3_4[3.3.4 ậCp ậnhт ếtín độ<br/>ọhc ậtp SRS]
47
48    %% Module 4: ảQun lý QR Code
49    M4 --> F4_1[4.1 Quét QR/Barcode]
50    M4 --> F4_2[4.2 ưLu mã QR]
51    M4 --> F4_3[4.3 Xóa mã QR]
52    M4 --> F4_4[4.4 Liên ếkt QR<br/>ỡvi báo ứthc]
53    M4 --> F4_5[4.5 Xác ựthc mã QR<br/>khi ấtt báo ứthc]
54
55    %% Module 5: ồThng kê & Báo cáo
56    M5 --> F5_1[5.1 Xem ồthng kê<brđộ/> chính xác<br/>theo ầtun]
57    M5 --> F5_2[5.2 Xem phân ồphi<br/>ạtrng thái ọhc ậtp]
58    M5 --> F5_3[5.3 Tính điểm<br/>Wake-up Score]
59    M5 --> F5_4[5.4 Theo dõi streak<br/>và điểm ốs]
60    M5 --> F5_5[5.5 Xem ỉlch ửs<br/>báo ứthc]
61
62 style Root fill:#e1f5ff
63 style M1 fill:#fff4e6
```

Phân tích các module chính

Module 1: Quản lý Báo thức

Module này chịu trách nhiệm toàn bộ vòng đời của báo thức, bao gồm 6 chức năng:

- **1.1 Tạo báo thức mới:** Cho phép người dùng tạo báo thức với đầy đủ cấu hình (giờ/phút, nhãn, ngày lặp lại, số câu hỏi, QR code, nhạc chuông, snooze).
- **1.2 Chỉnh sửa báo thức:** Sửa đổi thông tin báo thức đã tồn tại, bao gồm cả việc thay đổi câu hỏi/QR đã liên kết.
- **1.3 Xóa báo thức:** Xóa báo thức khỏi hệ thống và hủy lịch hẹn với Android AlarmManager.
- **1.4 Bật/Tắt báo thức:** Toggle trạng thái `isEnabled`, tự động đặt/hủy lịch hẹn hệ thống.
- **1.5 Xem danh sách báo thức:** Hiển thị tất cả báo thức, hỗ trợ sắp xếp (mặc định hoặc ưu tiên báo thức đang bật).
- **1.6 Tạo báo thức nhanh:** Tạo báo thức đổ chuông sau X phút (5/10/15/30 phút).

Module 2: Quản lý Chủ đề & Câu hỏi

Module này quản lý kho câu hỏi của người dùng theo cấu trúc phân cấp chủ đề:

- **2.1 Quản lý chủ đề:** CRUD chủ đề (tạo, sửa tên, xóa), hỗ trợ tìm kiếm theo tên.
- **2.2 Quản lý câu hỏi:** CRUD câu hỏi trong từng chủ đề, mỗi câu hỏi có 1 đáp án đúng và 3 đáp án sai.

Module 3: Thực thi Báo thức

Module này xử lý toàn bộ luồng báo thức reo, bao gồm 6 chức năng chính và 4 chức năng con của Quiz:

- **3.1 Kích hoạt báo thức:** Nhận broadcast từ AlarmManager, khởi động AlarmService.
- **3.2 Hiển thị giao diện báo thức đang reo:** Full-screen notification với các nút Tắt/Snooze.
- **3.3 Thực hiện Quiz:** Bao gồm:
 - 3.3.1 Chọn câu hỏi theo thuật toán SRS (Spaced Repetition System)
 - 3.3.2 Hiển thị câu hỏi và đếm giờ (15 giây/câu)
 - 3.3.3 Kiểm tra đáp án
 - 3.3.4 Cập nhật tiến độ học tập (`correctStreak`, `easinessFactor`, `interval`)
- **3.4 Quét QR Code để tắt:** Yêu cầu quét mã QR/Barcode trước khi tắt (nếu đã cấu hình).
- **3.5 Snooze báo thức:** Hẹn reo lại sau X phút (cấu hình trong `snoozeDuration`).
- **3.6 Tắt báo thức:** Dừng nhạc, cập nhật lịch sử, lập lại lịch nếu có lặp lại.

Module 4: Quản lý QR Code

Module này quản lý kho mã QR/Barcode (tối đa 5 mã):

- **4.1 Quét QR/Barcode:** Sử dụng camera và ML Kit để phát hiện mã.
- **4.2 Lưu mã QR:** Lưu vào database với tên do người dùng đặt.
- **4.3 Xóa mã QR:** Xóa mã khỏi hệ thống (và tự động hủy liên kết với báo thức).
- **4.4 Liên kết QR với báo thức:** Mỗi báo thức có thể dùng tối đa 3 mã.
- **4.5 Xác thực mã QR khi tắt báo thức:** Kiểm tra mã quét có khớp với mã đã chọn.

Module 5: Thống kê & Báo cáo

Module này cung cấp các báo cáo về hiệu suất học tập và thói quen thức dậy:

- **5.1 Xem thống kê độ chính xác theo tuần:** Biểu đồ đường 7 ngày (tỷ lệ % đúng).
- **5.2 Xem phân phối trạng thái học tập:** Biểu đồ tròn 3 phần (New/Learning/-Mastered).
- **5.3 Tính điểm Wake-up Score:** Điểm 0–100 dựa trên thời gian tắt và số lần snooze.
- **5.4 Theo dõi streak và điểm số:** Chuỗi ngày liên tiếp hoàn thành báo thức, tổng điểm tích lũy.
- **5.5 Xem lịch sử báo thức:** Chi tiết các lần reo/tắt/snooze.

1.1.2 Biểu đồ luồng dữ liệu

DFD Context (Mức ngữ cảnh)

DFD Context mô tả hệ thống như một "hộp đen" duy nhất, tương tác với 2 tác nhân ngoài:

- **E1 - Người dùng:** Gửi thông tin báo thức, chủ đề/câu hỏi, mã QR, lệnh bật/tắt/snooze, đáp án quiz, yêu cầu thống kê. Nhận danh sách báo thức, thông báo reo, câu hỏi quiz, kết quả, báo cáo.
- **E2 - Android AlarmManager:** Nhận lịch hẹn từ hệ thống. Gửi sự kiện đổ chuông (broadcast intent) đúng giờ.

```

1 %% DFD Context Diagram - ơS đồồ Lung ữD ệliu ữNg ắcnh
2 %% ệH ồthng Báo ứthc Thông minh
3
4 graph LR
5     %% External Entities
6     E1[E1<br/>uởNgi dùng]
7     E2[E2<br/>Android<br/>AlarmManager]
8
9     %% System
10    System((ỆH ỒTHNG<br/>BÁO ỨTHC<br/>THÔNG MINH))
11
12    %% Data Flows from User to System
13    E1 -->|Thông tin báo ứthc| System
14    E1 -->|ủCh đề & Câu ỏhi| System
15    E1 -->|Mã QR/Barcode| System
16    E1 -->|ệLnh ậbt/ấtt/snooze| System
17    E1 Đ-->|áp án quiz| System
18
19    %% Data Flows from System to User
20    System -->|Danh sách báo ứthc| E1
21    System -->|Thông báo đở chuông| E1
22    System -->|Câu ỏhi quiz| E1
23    System -->|ểKt ắqu quiz| E1
24    System -->|Báo cáo ồthng kê| E1
25    System -->|Yêu ầcu quét QR| E1
26
27    %% Data Flows with AlarmManager
28    System -->|ịLch ẹhn báo ứthc| E2
29    E2 -->|ựS ẹkin đở chuông| System
30
31    style E1 fill:#ffe6e6
32    style E2 fill:#ffe6e6
33    style System fill:#e1f5ff

```

Listing 1.2: DFD Context - Luồng dữ liệu mức ngữ cảnh

DFD Level 0 (Mức tổng quát)

DFD Level 0 phân rã hệ thống thành 7 tiến trình chính (P1–P7) và 5 kho dữ liệu (DS1–DS5).

```

1 %% DFD Level 0 - ơS đồồ Lung ữD ệliu ứMc 0
2 %% Các ếtín trình chính ủa ệh ốthng
3
4 graph TB
5     %% External Entities
6     E1[E1 ườNgi dùng]
7     E2[E2 Android<br/>AlarmManager]
8
9     %% Processes
10    P1((P1<br/>ảQun lý<br/>Báo ứthc))
11    P2((P2<br/>ảQun lý<br/>ủCh đề &<br/>Câu ỏhi))
12    P3((P3<br/>ậLp ịlch<br/>Báo ứthc))
13    P4((P4<br/>ủX lý<br/>Báo ứthc<br/>Reo))
14    P5((P5<br/>ựThc thi<br/>Quiz))
15    P6((P6<br/>ảQun lý<br/>QR Code))
16    P7((P7<br/>ạTo báo cáo<br/>ốThng kê))
17
18    %% Data Stores
19    DS1[(DS1<br/>Alarms)]
20    DS2[(DS2<br/>Topics &<br/>Questions)]
21    DS3[(DS3<br/>QR Codes)]
22    DS4[(DS4<br/>Progress<br/>& History)]
23    DS5[(DS5<br/>User Stats)]
24
25    %% Flows from E1 (User)
26    E1 -->|Thông tin báo ứthc| P1
27    E1 -->|ủCh đề & Câu ỏhi| P2
28    E1 -->|Mã QR| P6
29    E1 -->|ệLnh ậbt/ấtt| P1
30    E1 Đ-->|áp án quiz| P5
31    E1 -->|Yêu ầc ốthng kê| P7
32
33    %% Flows to E1 (User)
34    P1 -->|Danh sách báo ứthc| E1
35    P4 -->|Thông báo reo| E1
36    P5 -->|Câu ỏhi & ểKt ảqu| E1
37    P6 -->|Danh sách QR| E1
38    P7 -->|Báo cáo ốthng kê| E1
39
40    %% Flows with E2 (AlarmManager)
41    P3 -->|ịLch ẹhn| E2
42    E2 -->|ựS ẹkin đồồ chuông| P4
43
44    %% P1 - ảQun lý Báo ứthc
45    P1 <-->|ữD ệliu báo ứthc| DS1
46    P1 -->|Yêu ầc ậlp ịlch| P3
47
48    %% P2 - ảQun lý ủCh đề & Câu ỏhi
49    P2 <-->|ủCh đề & Câu ỏhi| DS2
50
51    %% P3 - ậLp ịlch Báo ứthc
52    P3 Đọ-->|c ấc ầc hình| DS1
53
54    %% P4 - ủX lý Báo ứthc Reo
55    P4 Đọ-->|c thông tin báo ứthc| DS1
56    P4 -->|ởKhi động quiz| P5
57    P4 -->|Yêu ầc quét QR| P6
58    P4 -->|Ghi ịlch ứs| DS4
59
60    %% P5 - ựThc thi Quiz
61    P5 Đọ-->|c câu ỏhi| DS2
62    P5 Đọ-->|c ếtín độ SRS| DS4
63    P5 -->|ậCp ậnh ếtín độ| DS4
64    P5 -->|ậCp ậnh stats| DS5
65
66    %% P6 - ảQun lý QR Code

```

Các tiến trình chính:

1. **P1 - Quản lý Báo thức:** Nhận thông tin báo thức từ E1, lưu/sửa/xóa vào DS1, gửi yêu cầu lập lịch cho P3, trả về danh sách báo thức cho E1.
2. **P2 - Quản lý Chủ đề & Câu hỏi:** Nhận chủ đề/câu hỏi từ E1, lưu vào DS2 (topics, questions).
3. **P3 - Lập lịch Báo thức:** Đọc cấu hình từ DS1, đặt lịch hẹn với E2 (AlarmManager).
4. **P4 - Xử lý Báo thức Reo:** Nhận sự kiện từ E2, đọc DS1, phát nhạc, hiển thị thông báo reo cho E1, khởi động P5 (Quiz) hoặc P6 (QR), ghi lịch sử vào DS4.
5. **P5 - Thực thi Quiz:** Đọc câu hỏi từ DS2, đọc tiến độ SRS từ DS4, hiển thị câu hỏi cho E1, nhận đáp án, cập nhật tiến độ vào DS4 và DS5.
6. **P6 - Quản lý QR Code:** Nhận mã QR từ E1, lưu/xóa vào DS3, liên kết với báo thức (DS1), xác thực mã khi tắt (gửi kết quả cho P4).
7. **P7 - Tạo báo cáo Thống kê:** Nhận yêu cầu từ E1, đọc DS4 và DS5, tính toán và trả về báo cáo.

Các kho dữ liệu:

1. **DS1 - Alarms:** Lưu trữ thông tin báo thức (giờ/phút, nhãn, daysOfWeek, isEnabled, questionCount, ringtoneUri, snooze...) và các bảng liên kết (alarm_topic_link, alarm_selected_questions, alarm_qr_link).
2. **DS2 - Topics & Questions:** Lưu chủ đề (topicName) và câu hỏi (prompt, options, correctAnswer, ownerTopicId).
3. **DS3 - QR Codes:** Lưu mã QR/Barcode (name, codeValue, codeType) và bảng liên kết alarm_qr_link.
4. **DS4 - Progress & History:** Lưu tiến độ SRS (question_progress), lịch sử trả lời (history), lịch sử báo thức (alarm_history).
5. **DS5 - User Stats:** Lưu thống kê người dùng (totalPoints, currentStreak, bestStreak) và thống kê chủ đề (topic_stats với userEloScore).

DFD Level 1 - Phân rã P5 (Thực thi Quiz)

Tiến trình P5 là phức tạp nhất, được phân rã thành 5 tiến trình con:

```

1 %% DFD Level 1 - Phân rã ếtin trình P5 (ựThc thi Quiz)
2 %% Đây là ếtin trình quan ợtrng ấnhտ ớvi ậthut toán SRS
3
4 graph TB
5     %% External to P5
6     E1[E1 ườNgi dùng]
7     DS2[(DS2<br/>Topics &<br/>Questions)]
8     DS4[(DS4<br/>Progress<br/>& History)]
9     DS5[(DS5<br/>User Stats)]
10    P4((P4<br/>ủX lý<br/>Bảo ứthc<br/>Reo))
11
12    %% Sub-processes of P5
13    P5_1((P5.1<br/>ọChn câu ỏhi<br/>theo SRS))
14    P5_2((P5.2<br/>ểHin ịth<br/>câu ỏhi &<brĐế/>m ởgi))
15    P5_3((P5.3<br/>ểKim tra<brđ/>áp án))
16    P5_4((P5.4<br/>ậCp ậnhտ<br/>ếtin độ SRS))
17    P5_5((P5.5<br/>Tính đểim<br/>ELO Topic))
18
19    %% Internal Data Store
20    DS_TEMP[(DS_TEMP<br/>ộB câu ỏhi<br/>Quiz ệhin ạti)]
21
22    %% Flow from P4
23    P4 -->|Alarm ID & ốS câu ỏhi| P5_1
24
25    %% P5.1 - ọChn câu ỏhi theo SRS
26    P5_1 Độ-->|c câu ỏhi được ọchn| DS2
27    P5_1 Độ-->|c ếtin độ SRS| DS4
28    P5_1 -->|Danh sách câu ỏhi<br/>u tiên| DS_TEMP
29
30    %% P5.2 - ểHin ịth câu ỏhi & Đếm ởgi
31    DS_TEMP -->|Câu ỏhi ếtip theo| P5_2
32    P5_2 -->|Câu ỏhi & Timer| E1
33
34    %% P5.3 - ểKim tra đáp án
35    E1 Đ-->|áp án ườngi dùng| P5_3
36    DS_TEMP Đ-->|áp án đúng| P5_3
37    P5_3 -->|ểKt ắqu đúng/sai| E1
38    P5_3 -->|ểKt ắqu & ởThi gian| P5_4
39
40    %% P5.4 - ậCp ậnhտ ếtin độ SRS
41    P5_4 Độ-->|c ếtin độ c| DS4
42    P5_4 -->|Ghi ếtin độở mi| DS4
43    P5_4 -->|Ghi ịlch ứs ắtr ờli| DS4
44    P5_4 -->|Question ID & ểKt ắqu| P5_5
45
46    %% P5.5 - Tính đểim ELO Topic
47    P5_5 Độ-->|c Topic Stats| DS5
48    P5_5 -->|ậCp ậnhտ đểim ELO| DS5
49
50    %% Completion flow
51    P5_3 -->|Hoàn thành đủ câu| P4
52
53    style E1 fill:#ffe6e6
54    style P4 fill:#ffccbc
55    style P5_1 fill:#c8e6c9
56    style P5_2 fill:#c8e6c9
57    style P5_3 fill:#c8e6c9
58    style P5_4 fill:#c8e6c9
59    style P5_5 fill:#c8e6c9
60    style DS2 fill:#fff9c4
61    style DS4 fill:#fff9c4
62    style DS5 fill:#fff9c4
63    style DS_TEMP fill:#ffe0b2

```

Các tiến trình con:

- **P5.1 - Chọn câu hỏi theo SRS:** Đọc danh sách câu hỏi đã chọn từ DS2 (qua alarm_topic_link và alarm_selected_questions), đọc tiến độ từ DS4, tính điểm ưu tiên (câu chưa học: 500, câu đến hạn: 1000+, câu khác: dùng difficultyScore), sắp xếp và chọn Top N, lưu vào DS_TEMP (bộ nhớ tạm).
- **P5.2 - Hiển thị câu hỏi & Đếm giờ:** Lấy câu hỏi từ DS_TEMP, hiển thị cho E1, bắt đầu timer 15 giây.
- **P5.3 - Kiểm tra đáp án:** Nhận đáp án từ E1, so sánh với đáp án đúng trong DS_TEMP, hiển thị kết quả (đúng/sai màu xanh/đỏ), gửi thông tin (isCorrect, timeSpent) cho P5.4.
- **P5.4 - Cập nhật tiến độ SRS:** Đọc tiến độ cũ từ DS4, tính toán SRS mới theo công thức:
 - Nếu đúng: correctStreak++, easinessFactor += 0.1, interval *= easinessFactor
 - Nếu sai: correctStreak = 0, easinessFactor -= 0.2 (min 1.3), interval = 1Ghi tiến độ mới và lịch sử trả lời vào DS4, gửi questionId cho P5.5.
- **P5.5 - Tính điểm ELO Topic:** Nhận questionId từ P5.4, đọc topic_stats từ DS5, cập nhật điểm ELO (+10 nếu đúng, -5 nếu sai).

DFD Level 1 - Phân rã P4 (Xử lý Báo thức Reo)

Tiến trình P4 xử lý toàn bộ luồng báo thức reo:

```

1 %% DFD Level 1 - Phân rã ếtín trình P4 (ủX lý Báo ứthc Reo)
2 %% ếtín trình ủX lý khi báo ứthc đổ chuông
3
4 graph TB
5     %% External to P4
6     E1[E1 ườNgi dùng]
7     E2[E2 Android<br/>AlarmManager]
8     DS1[(DS1<br/>Alarms)]
9     DS3[(DS3<br/>QR Codes)]
10    DS4[(DS4<br/>Progress<br/>& History)]
11    P5((P5<br/>ựThc thi<br/>Quiz))
12
13    %% Sub-processes of P4
14    P4_1((P4.1<br/>ậNhn ứs ệkin<brđổ/> chuông))
15    P4_2((P4.2<br/>ởKhi ạchy<br/>Alarm Service))
16    P4_3((P4.3<br/>ểHin ịth<br/>màn hình reo))
17    P4_4((P4.4<br/>ủX lý<br/>Snooze))
18    P4_5((P4.5<br/>ểKim tra<brđ/>ềiu ệkin ấtt))
19    P4_6((P4.6<br/>ậLp ặi ịlch<br/>ặlp ặi))
20    P4_7((P4.7<br/>Ghi ịlch ứs<br/>báo ứthc))
21
22    %% Flow from AlarmManager
23    E2 -->|Broadcast Intent| P4_1
24
25    %% P4.1 - ậNhn ứs ệkin đổ chuông
26    P4_1 -->|Alarm ID| P4_2
27    P4_1 Đọ-->|c thông tin báo ứthc| DS1
28
29    %% P4.2 - ởKhi ạchy Alarm Service
30    P4_2 -->|Phát ạnhc chuông| P4_3
31    P4_2 -->|ặTo notification| E1
32
33    %% P4.3 - ểHin ịth màn hình reo
34    P4_3 -->|Giao ệdin báo ứthc| E1
35    P4_3 -->|ặTo Alarm History| P4_7
36
37    %% Flows from User
38    E1 -->|ệLnh Snooze| P4_4
39    E1 -->|ệLnh ấTt| P4_5
40
41    %% P4.4 - ủX lý Snooze
42    P4_4 -->|ậCp ậnhst snooze count| DS4
43    P4_4 Đặ-->|t ịlch reo ặi| E2
44
45    %% P4.5 - ểKim tra ềiu ệkin ấtt
46    P4_5 Đọ-->|c ấcu hình QR| DS3
47    P4_5 -.->|Yêu ầcu quét QR| E1
48    P4_5 -->|ởKhi động Quiz| P5
49    P5 -->|ểKt ắqu hoàn thành| P4_5
50
51    %% P4.5 flows
52    P4_5 Đ-->|ềiu ệkin đủ| P4_6
53    P4_5 -->|ửDng service| P4_2
54    P4_5 -->|ậCp ậnhst dismissal| P4_7
55
56    %% P4.6 - ậLp ặi ịlch ặlp ặi
57    P4_6 Đọ-->|c daysOfWeek| DS1
58    P4_6 Đặ-->|t ịlch ếtíp theo| E2
59    P4_6 -.->|ấTt ếnu 1 ầln| DS1
60
61    %% P4.7 - Ghi ịlch ứs báo ứthc
62    P4_7 -->|Ghi AlarmHistory| DS4
63
64    style E1 fill:#ffe6e6
65    style E2 fill:#ffe6e6
66    style P5 fill:#ffccbc

```

Các tiến trình con:

- **P4.1 - Nhận sự kiện đổ chuông:** Nhận broadcast từ E2 (AlarmManager), đọc thông tin báo thức từ DS1.
- **P4.2 - Khởi chạy Alarm Service:** Phát nhạc chuông (MediaPlayer loop), tạo foreground notification với priority HIGH.
- **P4.3 - Hiển thị màn hình reo:** Hiển thị giao diện full-screen (AlarmRingingActivity) cho E1, tạo bản ghi AlarmHistory vào DS4 (ghi P4.7).
- **P4.4 - Xử lý Snooze:** Nếu E1 nhấn Snooze, cập nhật snoozeCount vào DS4, đặt lịch reo lại với E2.
- **P4.5 - Kiểm tra điều kiện tắt:** Đọc cấu hình QR từ DS3, nếu có QR thì yêu cầu quét (gọi P6), nếu có Quiz thì khởi động P5. Chỉ khi đủ điều kiện mới cho phép tắt, dừng service (P4.2), cập nhật dismissal (P4.7).
- **P4.6 - Lập lại lịch lặp lại:** Đọc daysOfWeek từ DS1, nếu có lặp lại thì đặt lịch tiếp theo với E2, nếu báo thức 1 lần thì cập nhật isEnabled = false vào DS1.
- **P4.7 - Ghi lịch sử báo thức:** Ghi AlarmHistoryEntity vào DS4 (scheduledTime, firstRingTime, dismissalTime, snoozeCount, isDismissed).

1.1.3 Đặc tả chức năng

Tiến trình P5.1 - Chọn câu hỏi theo SRS

Mục đích: Chọn N câu hỏi phù hợp nhất cho Quiz dựa trên thuật toán Spaced Repetition System (SRS).

Input:

- alarmId: Mã báo thức (Int)
- countNeeded: Số câu hỏi cần chọn (Int)
- DS2: Danh sách câu hỏi đã chọn (Topics/Questions)
- DS4: Tiến độ học tập (question_progress)

Output:

- DS_TEMP: Danh sách câu hỏi đã sắp xếp theo độ ưu tiên (List<Question>)

Xử lý:

1. Đọc alarm_selected_questions WHERE alarmId = alarmId
2. Đọc alarm_topic_link WHERE alarmId = alarmId
3. Lấy tất cả câu hỏi từ: (a) Câu hỏi lẻ (manual), (b) Câu hỏi từ các Topic đã chọn full
4. Loại bỏ trùng lặp theo questionId
5. **NẾU** danh sách rỗng: Trả về rỗng (không có câu hỏi)
6. Đọc question_progress cho tất cả questionId
7. **Với mỗi câu hỏi**, tính điểm ưu tiên:
 - **NẾU** chưa có progress (câu mới): priority = 500
 - **NẾU** có progress:
 - **NẾU** nextReviewDate <= now: priority = 1000 + (now - nextReviewDate)
 - **NGƯỢC LẠI:** priority = difficultyScore
8. Sắp xếp danh sách theo priority GIẢM DẦN
9. Thêm yếu tố ngẫu nhiên nhỏ (để tránh lặp lại hoàn toàn)

10. Lấy TOP countNeeded câu hỏi
11. Lưu vào DS_TEMP

Quy tắc nghiệp vụ:

- countNeeded phải > 0 và ≤ 100
- Nếu số câu hỏi có sẵn $< \text{countNeeded}$, chỉ trả về số câu có sẵn
- Câu hỏi mặc định (ID âm) luôn được ưu tiên thấp

Xử lý ngoại lệ:

- **E1:** Không có câu hỏi nào \rightarrow Trả về danh sách rỗng, P5 sẽ tắt báo thức ngay lập tức

Tiến trình P5.4 - Cập nhật tiến độ SRS

Mục đích: Cập nhật trạng thái học tập của câu hỏi sau khi người dùng trả lời.

Input:

- questionId: Mã câu hỏi (Int)
- isCorrect: Đúng hay sai (Boolean)
- timeSpentMs: Thời gian trả lời (Long, milliseconds)
- alarmHistoryId: Mã lịch sử báo thức (Int, nullable)

Output:

- DS4 (question_progress): Cập nhật tiến độ mới
- DS4 (history): Ghi lại lịch sử trả lời

Xử lý:

1. Ghi lại History: INSERT INTO history VALUES (questionId, alarmHistoryId, isCorrect, NOW(), timeSpentMs)
2. Đọc question_progress WHERE questionId = questionId
3. **NẾU** không tồn tại: Tạo mới progress với giá trị mặc định (correctStreak = 0, easinessFactor = 2.5, interval = 0, difficultyScore = 1000)
4. **NẾU** isCorrect = TRUE:
 - correctStreak = correctStreak + 1
 - easinessFactor = MIN(easinessFactor + 0.1, 3.0)
 - **NẾU** interval = 0: interval = 1
 - **NGƯỢC LẠI:** interval = ROUND(interval * easinessFactor)
 - difficultyScore = difficultyScore - 50
5. **NẾU** isCorrect = FALSE:
 - correctStreak = 0
 - easinessFactor = MAX(easinessFactor - 0.2, 1.3)
 - interval = 1
 - difficultyScore = difficultyScore + 100
6. Tính thời gian ôn tập tiếp theo: nextReviewDate = NOW() + interval (ngày), lastReviewedDate = NOW()
7. Cập nhật vào database: UPDATE question_progress SET ...

Quy tắc nghiệp vụ:

- `questionId` phải tồn tại trong bảng `questions` (trừ câu mặc định có ID âm)
- `timeSpentMs` phải ≥ 0
- `easinessFactor` luôn nằm trong khoảng $[1.3, 3.0]$

Xử lý ngoại lệ:

- **E1:** Nếu `questionId < 0` (câu hỏi mặc định): Bỏ qua cập nhật SRS

Tiến trình P3 - Lập lịch Báo thức

Mục đích: Đặt lịch hẹn với Android `AlarmManager` để hệ thống kích hoạt báo thức đúng giờ.

Input:

- `alarmEntity`: Thông tin báo thức (`AlarmEntity`)

Output:

- Lịch hẹn đã đặt trong hệ thống Android (`PendingIntent`)

Xử lý:

1. Tạo `Calendar` instance
2. `SET` `hour = alarmEntity.hour, minute = alarmEntity.minute, second = 0, millisecond = 0`
3. **NẾU** `calendar.timeInMillis <= currentTimeMillis`: `calendar.add(Calendar.DAY_OF_YEAR, 1)`
4. **NẾU** `daysOfWeek.isEmpty()`: Sử dụng thời gian vừa tính (báo thức 1 lần)
5. **NGƯỢC LẠI** (có lặp lại): Tìm ngày lặp gần nhất trong tương lai (vòng lặp `FOR` từ 0 đến 7, kiểm tra `dayCode` có trong `daysOfWeek` không)
6. Tạo `PendingIntent`:
 - `Intent target = AlarmReceiver`
 - `requestCode = alarmId` (Quan trọng: mỗi alarm có code riêng)
 - `extras: ALARM_ID, ALARM_LABEL, RINGTONE_URI`
 - `flags: FLAG_UPDATE_CURRENT | FLAG_IMMUTABLE`
7. Gọi `AlarmManager.setAlarmClock(triggerTime, pendingIntent)`

Quy tắc nghiệp vụ:

- Kiểm tra quyền `SCHEDULE_EXACT_ALARM` (Android 12+)
- Dùng `setAlarmClock` để đảm bảo đánh thức máy ngay cả khi Doze mode

Xử lý ngoại lệ:

- **E1:** Thiếu quyền `SCHEDULE_EXACT_ALARM` → Hiển thị dialog yêu cầu người dùng cấp quyền

1.1.4 Phân tích hệ thống về mặt dữ liệu

Mô hình ER hạn chế

Mô hình Entity-Relationship (ER) mô tả cấu trúc dữ liệu và các quan hệ giữa các thực thể.

```

1 %% ERD - Entity Relationship Diagram
2 %% Mô hình ểliu ủa ệh ốthng Báo ứthc Thông minh
3
4 erDiagram
5     ALARMS ||--o{ ALARM_TOPIC_LINK : "có"
6     ALARMS ||--o{ ALARM_SELECTED_QUESTIONS : "ọchn"
7     ALARMS ||--o{ ALARM_QR_LINK : "ứs ụngng"
8     ALARMS ||--o{ ALARM_HISTORY : "ạto ra"
9
10    TOPICS ||--o{ ALARM_TOPIC_LINK : đượ"c ọchn ởbi"
11    TOPICS ||--o{ QUESTIONS : "ứcha"
12    TOPICS ||--o{ TOPIC_STATS : "có"
13
14    QUESTIONS ||--o{ ALARM_SELECTED_QUESTIONS : đượ"c ọchn"
15    QUESTIONS ||--o{ QUESTION_PROGRESS : "có"
16    QUESTIONS ||--o{ HISTORY : "ạto ịlch ứs"
17
18    QR_CODES ||--o{ ALARM_QR_LINK : đượ"c dùng ởbi"
19
20    ALARM_HISTORY ||--o{ HISTORY : "ứcha"
21
22    USER_STATS ||--|| USER : "ộthuc ềv"
23
24    ALARMS {
25        int alarmId PK
26        int hour
27        int minute
28        string label
29        set daysOfWeek
30        int questionCount
31        boolean isEnabled
32        string ringtoneUri
33        int snoozeDuration
34        boolean snoozeEnabled
35    }
36
37    TOPICS {
38        int topicId PK
39        string topicName
40    }
41
42    QUESTIONS {
43        int questionId PK
44        int ownerTopicId FK
45        string prompt
46        list options
47        string correctAnswer
48    }
49
50    ALARM_TOPIC_LINK {
51        int alarmId PK_FK
52        int topicId PK_FK
53    }
54
55    ALARM_SELECTED_QUESTIONS {
56        int selectionId PK
57        int alarmId FK
58        int questionId
59        int topicId
60    }
61
62    QR_CODES {
63        int qrId PK
64        string name
65        string codeValue
66        string codeType

```

Các thực thể chính:

1. **ALARMS (Báo thức):** Lưu thông tin báo thức (alarmId PK, hour, minute, label, daysOfWeek, questionCount, isEnabled, ringtoneUri, snoozeDuration, snoozeEnabled).
2. **TOPICS (Chủ đề):** Lưu chủ đề câu hỏi (topicId PK, topicName).
3. **QUESTIONS (Câu hỏi):** Lưu câu hỏi (questionId PK, ownerTopicId FK, prompt, options, correctAnswer).
4. **QR_CODES (Mã QR):** Lưu mã QR/Barcode (qrId PK, name, codeValue, codeType, createdAt).
5. **QUESTION_PROGRESS (Tiến độ học tập):** Lưu tiến độ SRS (questionId PK_FK, correctStreak, lastReviewedDate, nextReviewDate, difficultyScore, easinessFactor, interval).
6. **TOPIC_STATS (Thống kê chủ đề):** Lưu điểm ELO (topicId PK_FK, userEloScore).
7. **HISTORY (Lịch sử trả lời):** Lưu từng câu trả lời (historyId PK, questionId FK, alarmHistoryId FK, isCorrect, answeredAt, timeToAnswerMs).
8. **ALARM_HISTORY (Lịch sử báo thức):** Lưu thông tin reo/tắt (historyId PK, alarmId FK, snoozeCount, scheduledTime, firstRingTime, dismissalTime, isDismissed).
9. **USER_STATS (Thống kê người dùng):** Lưu thống kê tổng hợp (userId PK, totalPoints, currentStreak, bestStreak, totalAlarmsDismissed, lastActiveDate).

Các bảng liên kết (Junction Tables):

1. **ALARM_TOPIC_LINK:** Quan hệ N-N giữa ALARMS và TOPICS (alarmId PK_FK, topicId PK_FK). Cascade delete.
2. **ALARM_SELECTED_QUESTIONS:** Báo thức chọn câu hỏi lẻ (selectionId PK, alarmId FK, questionId, topicId). Lưu ý: questionId có thể âm (câu mặc định).
3. **ALARM_QR_LINK:** Quan hệ N-N giữa ALARMS và QR_CODES (alarmId PK_FK, qrId PK_FK). Cascade delete.

Quan hệ chính:

- ALARMS (1) (N) ALARM_TOPIC_LINK (N) TOPICS (1)
- ALARMS (1) (N) ALARM_SELECTED_QUESTIONS
- ALARMS (1) (N) ALARM_QR_LINK (N) QR_CODES (1)
- ALARMS (1) (N) ALARM_HISTORY
- TOPICS (1) (N) QUESTIONS
- TOPICS (1) (1) TOPIC_STATS
- QUESTIONS (1) (1) QUESTION_PROGRESS
- QUESTIONS (1) (N) HISTORY
- ALARM_HISTORY (1) (N) HISTORY
- USER (1) (1) USER_STATS

Ràng buộc toàn vẹn:

- **PK:** Tất cả bảng có Primary Key (auto-increment hoặc composite).
- **FK:** Foreign Key với ON DELETE CASCADE (xóa parent tự động xóa child).
- **NOT NULL:** Các trường quan trọng (topicName, prompt, codeValue...).
- **UNIQUE:** topicName (trong TOPICS), codeValue (trong QR_CODES).
- **CHECK:** hour (0–23), minute (0–59), snoozeDuration (1–60).

Chương 2

Thiết kế

2.1 Thiết kế tổng thể

2.1.1 Kiến trúc ứng dụng

Hệ thống sử dụng kiến trúc **MVVM (Model-View-ViewModel)** kết hợp với **Clean Architecture**, phù hợp với Android Jetpack và Kotlin Coroutines.

Phân tầng

Ứng dụng được phân thành 4 tầng chính:

1. UI Layer (Presentation)

Tầng này sử dụng Jetpack Compose để xây dựng giao diện:

- **Screens:** AlarmScreen, AlarmSettingsScreen, QuizScreen, TopicScreen, TopicDetailScreen, StatsScreen, QRCodeScannerScreen, AlarmRingScreen.
- **Components:** AlarmCard, QuestionCard, TopicCard, PermissionItem, PremiumPurchaseDialog, SearchableTopBar.
- **Navigation:** MainScreen với Bottom Navigation (3 tabs: Alarm, Topic, Stats), NavHost quản lý stack navigation.

2. ViewModel Layer

Tầng này quản lý UI state và xử lý user interaction:

- **AlarmViewModel:** Quản lý danh sách báo thức, toggle bật/tắt, xóa, tính thời gian chờ chuông tiếp theo.
- **AlarmSettingsViewModel:** Quản lý state cấu hình báo thức (giờ/phút, label, daysOfWeek, mission, QR, snooze).
- **QuizViewModel:** Quản lý quiz state (câu hỏi hiện tại, timer progress, correctCount, đáp án đã chọn).
- **TopicViewModel:** Quản lý danh sách chủ đề, tìm kiếm.
- **TopicDetailViewModel:** Quản lý danh sách câu hỏi trong chủ đề.
- **StatsViewModel:** Tính toán và cung cấp dữ liệu thống kê (weekly accuracy, SRS distribution, wake-up score).
- **QRCodeViewModel:** Quản lý danh sách QR, quét/lưu/xóa mã.

3. Logic Layer (Domain/Business Logic)

Tầng này chứa logic nghiệp vụ:

- **AlarmScheduler:** Đặt/hủy lịch hẹn với Android AlarmManager.

- **AlarmReceiver (BroadcastReceiver):** Nhận sự kiện đổ chuông từ AlarmManager, khởi động AlarmService.
- **AlarmService (Foreground Service):** Phát nhạc chuông (MediaPlayer loop), hiển thị notification với full-screen intent.
- **QuestionAlgorithmManager:** Thuật toán SRS (chọn câu hỏi ưu tiên, cập nhật tiến độ, tính điểm ELO).

4. Data Layer (Repository/DAO)

Tầng này quản lý truy cập dữ liệu:

- **AppDatabase (Room):** Singleton database (SQLite), version 2, chứa 12 bảng.
- **AppDao:** CRUD operations cho Alarms, Topics, Questions, QR Codes, liên kết, tiến độ SRS.
- **StatisticsDao:** Query phức tạp cho thống kê (GROUP BY, JOIN, aggregation).
- **Entities:** AlarmEntity, TopicEntity, QuestionEntity, QRCodeEntity, QuestionProgressEntity, HistoryEntity, AlarmHistoryEntity, UserStatsEntity, TopicStatsEntity.
- **Converters:** TypeConverter cho Set<String> (daysOfWeek), List<String> (options), Date (timestamp).

Luồng dữ liệu và state management

Luồng dữ liệu (Data Flow):

1. **UI → ViewModel:** User actions (clicks, text input) thông qua event handlers (onClick, onChange).
2. **ViewModel → Logic:** Gọi business logic (VD: AlarmViewModel gọi AlarmScheduler.schedule).
3. **Logic → Data:** CRUD operations qua Dao (VD: alarmDao.insertAlarm()).
4. **Data → ViewModel:** Emit Flow/StateFlow khi data thay đổi (reactive).
5. **ViewModel → UI:** Collect Flow trong Composable, UI tự động recompose.

State Management:

- **StateFlow:** ViewModel expose UI state dạng StateFlow<T> (VD: alarms: StateFlow<List<T>>).
- **Flow:** Data layer trả về Flow<T> từ Room database (auto-update khi DB thay đổi).
- **MutableStateFlow:** ViewModel quản lý internal state (VD: _uiState: MutableStateFlow<T>).
- **Coroutines:** Tất cả async operations dùng viewModelScope.launch, tự động cancel khi ViewModel clear.

2.1.2 Các module chính

Dựa trên cấu trúc package và phân tích FDD, hệ thống có 5 module chính:

1. **Module Alarm:** ui/theme/alarm/, alarm_logic/. Bao gồm: AlarmScreen, AlarmSettingsScreen, AlarmRingingScreen, QuizScreen, AlarmViewModel, AlarmSettingsViewModel, QuizViewModel, AlarmScheduler, AlarmReceiver, AlarmService.
2. **Module Topic:** ui/theme/topic/. Bao gồm: TopicScreen, TopicDetailScreen, TopicViewModel, TopicDetailViewModel.
3. **Module Stats:** ui/theme/stats/. Bao gồm: StatsScreen, StatsViewModel, StatisticsDao.
4. **Module QR Code:** ui/theme/qrcode/. Bao gồm: QRCodeScannerScreen, QRCodeManagementDialog, QRCodeSelectionDialog, QRCodeViewModel.
5. **Module Logic:** logic/. Bao gồm: QuestionAlgorithmManager (thuật toán SRS).

2.1.3 Luồng xử lý tổng quát

Luồng tạo báo thức

1. User nhấn "Thêm báo thức" → AlarmScreen navigate → AlarmSettingsScreen
2. AlarmSettingsViewModel điền giá trị mặc định → hiển thị form
3. User chỉnh sửa thông tin (giờ/phút/label/days), chọn Mission/QR (optional)
4. User nhấn "Lưu" → AlarmSettingsViewModel validate → lưu AlarmEntity vào DB (qua AppDao)
5. ViewModel lưu các liên kết (alarm__topic__link, alarm__selected__questions, alarm__qr__link)
6. ViewModel gọi AlarmScheduler.schedule() → đặt lịch với AlarmManager
7. Navigate back → AlarmScreen hiển thị báo thức mới

Luồng báo thức reo và Quiz

1. AlarmManager gửi broadcast đúng giờ → AlarmReceiver.onReceive()
2. AlarmReceiver đọc alarm từ DB, kiểm tra daysOfWeek (lặp lại hoặc 1 lần)
3. AlarmReceiver khởi động AlarmService (foreground service)
4. AlarmService phát nhạc (MediaPlayer loop), tạo notification priority HIGH, khởi động AlarmRingActivity (full-screen intent)
5. AlarmRingActivity hiển thị màn hình reo, user nhấn "Tắt"
6. Nếu có QR: navigate → QRCodeScannerScreen, user quét mã, xác thực
7. Nếu có Quiz: navigate → QuizScreen
8. QuizViewModel gọi QuestionAlgorithmManager.generateMissionQuestions() → chọn câu hỏi theo SRS
9. Loop: hiển thị câu hỏi, timer 15s, user chọn đáp án, kiểm tra, cập nhật SRS (processAnswer), tăng correctCount
10. Đủ số câu đúng → cập nhật AlarmHistory (dismissalTime), cập nhật UserStats, dừng AlarmService, finish activity

2.2 Giao diện

2.2.1 Nhóm màn hình và chức năng

Ứng dụng có 3 màn hình chính (tabs) và nhiều màn hình phụ:

Màn hình chính (Bottom Navigation):

1. **AlarmScreen (Tab 1 - Báo thức):** Hiển thị danh sách báo thức dạng card, mỗi card có:
 - Thời gian (giờ:phút) lớn
 - Nhãn (label)
 - Ngày lặp lại (T2, T3... hoặc "Một lần")
 - Switch bật/tắt
 - Icon sửa/xóaHeader hiển thị "Báo thức tiếp theo sau X giờ Y phút", FAB "Thêm mới".
2. **TopicScreen (Tab 2 - Chủ đề):** Hiển thị danh sách chủ đề dạng card, mỗi card có:
 - Tên chủ đề
 - Số câu hỏi (VD: "15 câu")
 - Icon xem chi tiết

Top bar có search field, FAB "Thêm chủ đề".

3. **StatsScreen (Tab 3 - Thống kê):** Hiển thị:

- Line chart: Độ chính xác 7 ngày (trục X: ngày, trục Y: % đúng)
- Pie chart: Phân phối SRS (New/Learning/Mastered)
- Card: Wake-up Score (điểm 0–100)
- Card: User Stats (Tổng điểm, Streak hiện tại, Streak tốt nhất, Tổng báo thức đã tắt)

Màn hình phụ:

1. **AlarmSettingsScreen:** Form cấu hình báo thức với:

- Time Picker (giờ/phút)
- TextField (label)
- Chip Selector (ngày lặp lại: T2–CN)
- Button "Chọn nhạc chuông"(Ringtone Picker)
- Slider "Số câu hỏi"(0–10)
- Button "Chọn câu hỏi" → MissionSelectionDialog
- Button "Chọn QR Code" → QRCodeManagementDialog
- Switch + Slider "Snooze"(1–60 phút)
- Text "Đổ chuông sau X giờ Y phút"(tự động tính)
- Button "Lưu" và "Hủy"

2. **QuizScreen:** Hiển thị quiz với:

- Header: Progress bar (X/Y câu đúng)
- Circular timer progress (15s)
- Text: Nội dung câu hỏi
- 4 nút đáp án (card), highlight màu xanh (đúng) hoặc đỏ (sai) sau khi chọn
- Button "Quay lại"(disabled khi quiz đang chạy)

3. **AlarmRingScreen:** Full-screen activity với:

- Icon báo thức lớn
- Text: Label báo thức
- Text: Thời gian hiện tại
- Button "Tắt"(chính)
- Button "Snooze"(nếu enabled)

4. **TopicDetailScreen:** Hiển thị danh sách câu hỏi trong chủ đề dạng card (prompt + số đáp án), FAB "Thêm câu hỏi".

5. **QRCodeScannerScreen:** Camera preview toàn màn hình, khung hình quét, text hướng dẫn "Đưa mã QR vào khung hình", button "Quay lại".

2.2.2 Điều hướng (Navigation)

Bottom Navigation (3 tabs):

- Tab 1: Báo thức (route: "alarm")
- Tab 2: Chủ đề (route: "topic")
- Tab 3: Thống kê (route: "stats")

Stack Navigation:

- alarm → alarm_settings/alarmId (sửa báo thức)

- alarm → alarm_settings/-1 (tạo mới)
- alarm → alarm_ringing (báo thức reo)
- alarm_ringing → quiz_screen (làm quiz)
- alarm_settings → mission_selection (dialog)
- alarm_settings → qr_management (dialog)
- topic → topic_detail/topicId (xem chi tiết chủ đề)
- qr_management → qr_scanner (quét mã)

Deep Link:

Khi AlarmService khởi động, nó gửi intent với extra `NAVIGATE_TO = "QUIZ_SCREEN"`, MainActivity nhận và navigate tới QuizScreen.

2.2.3 Trạng thái (State Management)

UI State:

Mỗi screen có data class state tương ứng (VD: QuizUiStateData, AlarmSettingData), chứa:

- Dữ liệu hiển thị (questions, alarms, topics...)
- Trạng thái loading/error
- Trạng thái tương tác (isAnswered, selectedAnswerId, timerProgress...)

State Flow:

- ViewModel expose `StateFlow<UiState>` hoặc `Flow<List<Data>>`
- UI collect trong `LaunchedEffect` hoặc `collectAsState()`
- Khi state thay đổi, UI tự động recompose

2.2.4 Validation

Validation Input:

- **Giờ/Phút:** Kiểm tra khoảng hợp lệ (0–23, 0–59), `TimePicker` tự validate.
- **Label:** Không bắt buộc, tối đa 100 ký tự.
- **Số câu hỏi:** Slider (0–10), không thể vượt quá.
- **Snooze duration:** Slider (1–60 phút).
- **Topic name:** Không được rỗng, tối đa 100 ký tự.
- **Question prompt:** Không được rỗng, tối đa 500 ký tự.
- **QR name:** Không được rỗng, tối đa 50 ký tự.

Validation nghiệp vụ:

- **Số mã QR:** Tối đa 5 mã trong hệ thống, kiểm tra bằng `dao.getQRCodeCount()`.
- **QR per alarm:** Tối đa 3 mã/báo thức, kiểm tra bằng `dao.getQRLinkCountForAlarm()`.
- **Trùng lặp QR:** Kiểm tra `codeValue` bằng `dao.getQRCodeByValue()`.
- **Trùng lặp Topic:** Room tự động check (UNIQUE constraint trên `topicName`).

Error Handling:

- Hiển thị Snackbar hoặc AlertDialog khi validation fail
- VD: "Bạn chỉ có thể lưu tối đa 5 mã QR. Vui lòng xóa bớt mã cũ."
- Try-catch trong ViewModel, log lỗi, cập nhật UI state (`isLoading = false`, `errorMessage = ...`)

2.3 Database

2.3.1 Thiết kế bảng và quan hệ

Hệ thống sử dụng Room Database (SQLite) với 12 bảng chính. Database name: `app_database`, version: 2.

Bảng ALARMS:

Tên cột	Kiểu	Ràng buộc	Mô tả
alarmId	INTEGER	PK, Auto-increment	Mã báo thức
hour	INTEGER	NOT NULL, CHECK (0–23)	Giờ
minute	INTEGER	NOT NULL, CHECK (0–59)	Phút
label	TEXT	Nullable	Nhãn báo thức
daysOfWeek	TEXT	NOT NULL	Set<String> (VD: "T2,T3,T4")
questionCount	INTEGER	NOT NULL, DEFAULT 0	Số câu hỏi bắt buộc
isEnabled	INTEGER	NOT NULL, DEFAULT 1	Trạng thái bật/tắt (Boolean)
ringtoneUri	TEXT	Nullable	URI nhạc chuông
snoozeDuration	INTEGER	NOT NULL, DEFAULT 5	Thời gian snooze (phút)
snoozeEnabled	INTEGER	NOT NULL, DEFAULT 0	Cho phép snooze (Boolean)

Bảng TOPICS:

Tên cột	Kiểu	Ràng buộc	Mô tả
topicId	INTEGER	PK, Auto-increment	Mã chủ đề
topicName	TEXT	NOT NULL, UNIQUE	Tên chủ đề

Bảng QUESTIONS:

Tên cột	Kiểu	Ràng buộc	Mô tả
questionId	INTEGER	PK, Auto-increment	Mã câu hỏi
ownerTopicId	INTEGER	FK → topics(topicId) ON DELETE CASCADE	Chủ đề sở hữu
prompt	TEXT	NOT NULL	Nội dung câu hỏi
options	TEXT	NOT NULL	List<String> (JSC án sai)
correctAnswer	TEXT	NOT NULL	Đáp án đúng

Bảng QR_CODES:

Tên cột	Kiểu	Ràng buộc	Mô tả
qrId	INTEGER	PK, Auto-increment	Mã QR
name	TEXT	NOT NULL	Tên do người dùng đặt
codeValue	TEXT	NOT NULL, UNIQUE	Giá trị mã QR/Barcode
codeType	TEXT	NOT NULL	"QR" hoặc "BARCODE"
createdAt	INTEGER	NOT NULL	Timestamp (Long)

Bảng ALARM_TOPIC_LINK (Junction Table):

Tên cột	Kiểu	Ràng buộc	Mô tả
alarmId	INTEGER	PK, FK → alarms(alarmId) ON DELETE CASCADE	Mã báo thức

topicId	INTEGER	PK, FK → topics(topicId) ON DELETE CASCADE	Mã chủ đề
---------	---------	--	-----------

Các bảng khác: alarm_selected_questions, alarm_qr_link, question_progress, topic_stats, history, alarm_history, UserStats (cấu trúc tương tự, đã mô tả trong ERD).

2.3.2 Ràng buộc toàn vẹn

Primary Key (PK):

Tất cả bảng có PK (auto-increment hoặc composite). VD: alarmId, topicId, hoặc (alarmId, topicId).

Foreign Key (FK):

Tất cả FK có ON DELETE CASCADE, đảm bảo xóa parent tự động xóa child. VD: Xóa Alarm → xóa alarm_topic_link, alarm_selected_questions, alarm_qr_link, alarm_history.

NOT NULL:

Các trường quan trọng không được null: topicName, prompt, codeValue, hour, minute.

UNIQUE:

topicName (trong TOPICS), codeValue (trong QR_CODES).

CHECK Constraint:

hour (0–23), minute (0–59), snoozeDuration (1–60). Room không hỗ trợ CHECK trực tiếp, kiểm tra trong code.

Index:

Tự động tạo index cho FK (Room auto-index). Thêm index thủ công nếu cần: @Index("ownerTopic").

2.3.3 Mapping từ Entity/Room

Entity → Table: Mapping 1:1, mỗi data class có @Entity(tableName = "...") ánh xạ trực tiếp tới bảng.

TypeConverter:

Room cần TypeConverter cho các kiểu phức tạp:

- **Set<String> String:** daysOfWeek lưu dạng "T2,T3,T4". Converter: joinToString(",") và split(",").toSet().
- **List<String> String JSON:** options lưu dạng JSON array. Converter: dùng Gson (toJson() và fromJson()).
- **Date Long:** answeredAt, nextReviewDate lưu dạng timestamp (milliseconds). Converter: date.time và Date(timestamp).

ViewModel → Entity:

ViewModel quản lý UI state (ephemeral data), khi save thì convert sang Entity. VD: AlarmSettingData (UI state) → AlarmEntity (DB entity).

```
val alarmEntity = AlarmEntity(
    alarmId = if (state.id == -1) 0 else state.id,
    hour = state.hour,
    minute = state.minute,
    label = state.label,
    daysOfWeek = state.daysOfWeek,
    ...
)
alarmDao.insertAlarm(alarmEntity)
```

Giả định/TBD

- **Giả định 1:** Người dùng luôn cấp quyền `CAMERA`, `POST_NOTIFICATIONS`, `SCHEDULE_EXACT_ALARM`. Trên thực tế, cần xử lý trường hợp từ chối quyền chi tiết hơn.
- **Giả định 2:** Chỉ có 1 người dùng (`userId = 1`). Không hỗ trợ multi-profile.
- **TBD 1:** Chưa có backup/restore dữ liệu. Cần implement Export/Import (JSON/CSV) trong phiên bản tiếp theo.
- **TBD 2:** Thuật toán SRS chưa tính forgetting curve (đường cong quên). Có thể cải tiến bằng cách tích hợp thuật toán Anki.
- **TBD 3:** Chưa hỗ trợ câu hỏi dạng hình ảnh. Cần thêm `imageUri` vào `QuestionEntity` và `ImagePicker` trong UI.