|  |  |  |
| --- | --- | --- |
|  | | |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **---------------------------------------** | | |
| **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH KỸ THUẬT PHẦN MỀM** | | |
|  | | |
| **XÂY DỰNG ỨNG DỤNG VIẾT NHẬT KÝ ĐỜI SỐNG HÀNG NGÀY CHO SINH VIÊN** | | |
|  | | |
|  | | |
| **CBHD: TS. *Nguyễn Văn Tỉnh*** | |  |
| **Sinh viên: Ngô Thị Mai – K15** |  | |
| **Mã số sinh viên: 2020604962** |  | |
|  | | |
|  | | |
|  | | |
|  | | |
| Hà Nội – Năm 2024 | | |

# **LỜI CẢM ƠN**

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến trường Đại học Công Nghiệp Hà Nội, thầy cô giáo khoa Công nghệ thông tin đã cung cấp cho em một môi trường học tập chuyên nghiệp, tạo điều kiện thuận lợi và truyền dạy kiến thức cho em trong thời gian qua để em có thể hoàn thành đồ án tốt nghiệp một cách tốt nhất. Và hơn hết em xin bày tỏ lòng biết ơn sâu sắc nhất tới thầy giáo hướng dẫn **TS. Nguyễn Văn Tỉnh** đã tận tình giúp đỡ, định hướng cho em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và xây dựng đồ án tốt nghiệp.

Khi thực hiện nghiên cứu đề tài, em đã cố gắng hoàn thành tốt nhất báo cáo đồ án tốt nghiệp trong khả năng của bản thân. Em rất mong nhận được sự đóng góp của tất cả các thầy cô giáo để đồ án của em được đầy đủ và hoàn chỉnh hơn.

Lời cuối cùng, em xin kính chúc quý thầy cô luôn mạnh khỏe, hoàn thành tốt công việc của mình, chúc khoa Công nghệ thông tin, nhà trường ngày càng phát triển và trở thành một môi trường học tập tuyệt vời cho các thế hệ sinh viên tiếp theo.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI CẢM ƠN** 2](#_Toc167093539)

[**DANH MỤC HÌNH ẢNH** v](#_Toc167093540)

[**DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT** vii](#_Toc167093541)

[**DANH MỤC BẢNG BIỂU** viii](#_Toc167093542)

[**Mở đầu** 1](#_Toc167093543)

[**1.** **Lý do chọn đề tài** 1](#_Toc167093544)

[**2.** **Mục tiêu của đề tài** 1](#_Toc167093545)

[**3.** **Đối tượng và phạm vi** 2](#_Toc167093546)

[**4.** **Phương pháp nghiên cứu** 2](#_Toc167093547)

[**Chương 1 TỔNG QUAN** 3](#_Toc167093548)

[1.1 Cơ sở lý thuyết 3](#_Toc167093549)

[1.1.1 Xác định yêu cầu 3](#_Toc167093550)

[1.1.2 Tổng quan về lập trình ứng dụng di động trên nền tảng Android 3](#_Toc167093551)

[1.1.2.1 Giới thiệu về lập trình ứng dụng di động 3](#_Toc167093552)

[1.1.2.2 Ngôn ngữ sử dụng 3](#_Toc167093553)

[1.1.2.3 Giới thiệu về hệ điều hành Android 4](#_Toc167093554)

[1.1.3 Công cụ lập trình Android Studio 4](#_Toc167093555)

[1.1.3.1 Giới thiệu về Android Studio 4](#_Toc167093556)

[1.1.3.2 Các tính năng hàng đầu của Android Studio 5](#_Toc167093557)

[1.1.4 Ngôn ngữ lập trình Kotlin 8](#_Toc167093558)

[1.1.4.1 Giới thiệu về ngôn ngữ lập trình Kotlin 8](#_Toc167093559)

[1.1.4.2 Tính năng của ngôn ngữ lập trình Kotlin 8](#_Toc167093560)

[1.1.4.3 Ưu nhược điểm của Kotlin 9](#_Toc167093561)

[1.1.5 Mô hình MVVM 9](#_Toc167093562)

[1.1.6 Room Database 12](#_Toc167093563)

[1.2 Khảo sát 14](#_Toc167093564)

[1.2.1 Phát biểu bài toán 14](#_Toc167093565)

[1.2.2 Mục đích 15](#_Toc167093566)

[1.2.3 Mô tả các yêu cầu 15](#_Toc167093567)

[1.2.3.1 Yêu cầu chức năng 15](#_Toc167093568)

[1.2.3.2 Các yêu cầu khác 17](#_Toc167093569)

[**Chương 2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 18](#_Toc167093570)

[2.1. Mô hình use cases 18](#_Toc167093571)

[2.1.1. Xác định các tác nhân và các ca sử dụng 18](#_Toc167093572)

[2.1.2. Biểu đồ Use case 19](#_Toc167093573)

[2.1.3. Các đặc tả use case 19](#_Toc167093574)

[**2.1.3.1.** **Use case Tạo profile** 19](#_Toc167093575)

[**2.1.3.2.** **Use case Tạo mật khẩu** 20](#_Toc167093576)

[**2.1.3.3.** **Use case Tạo nhật ký** 22](#_Toc167093577)

[**2.1.3.4.** **Use case Sửa nhật ký** 23](#_Toc167093578)

[**2.1.3.5.** **Use case Xóa nhật ký** 24](#_Toc167093579)

[**2.1.3.6.** **Use case Tìm kiếm nhật ký** 25](#_Toc167093580)

[**2.1.3.7.** **Use case Thống kê cảm xúc** 25](#_Toc167093581)

[**2.1.3.8.** **Use case Tạo thời khóa biểu** 27](#_Toc167093582)

[**2.1.3.9.** **Use case Sửa thời khóa biểu** 28](#_Toc167093583)

[**2.1.3.10.** **Use case Xóa thời khóa biểu** 29](#_Toc167093584)

[**2.1.3.11.** **Use case Đổi ngôn ngữ** 30](#_Toc167093585)

[**2.1.3.12.** **Use case Đổi chủ đề** 31](#_Toc167093586)

[2.2. Biểu đồ lớp 32](#_Toc167093587)

[2.3. Phân tích use case 32](#_Toc167093588)

[2.3.1. Phân tích Use case Tạo Profile 32](#_Toc167093589)

[**2.3.1.1.** **Biểu đồ lớp phân tích** 32](#_Toc167093590)

[**2.3.1.2.** **Biểu đồ trình tự** 33](#_Toc167093591)

[2.3.2. Phân tích Use case Tạo mật khẩu 33](#_Toc167093592)

[**2.3.2.1.** **Biểu đồ lớp phân tích** 33](#_Toc167093593)

[**2.3.2.2.** **Biểu đồ trình tự** 34](#_Toc167093594)

[2.3.3. Phân tích Use case Tạo nhật ký 34](#_Toc167093595)

[**2.3.3.1.** **Biểu đồ lớp phân tích** 34](#_Toc167093596)

[**2.3.3.2.** **Biểu đồ trình tự** 35](#_Toc167093597)

[2.3.4. Phân tích Use case Thống kê cảm xúc 35](#_Toc167093598)

[**2.3.4.1.** **Biểu đồ lớp phân tích** 35](#_Toc167093599)

[**2.3.4.2.** **Biểu đồ trình tự** 36](#_Toc167093600)

[2.3.5. Phân tích Use case Sửa nhật ký 36](#_Toc167093601)

[**2.3.5.1.** **Biểu đồ lớp phân tích** 36](#_Toc167093602)

[**2.3.5.2.** **Biểu đồ trình tự** 37](#_Toc167093603)

[2.3.6. Phân tích Use case Xóa Nhật ký 37](#_Toc167093604)

[**2.3.6.1.** **Biểu đồ lớp phân tích** 37](#_Toc167093605)

[**2.3.6.2.** **Biểu đồ trình tự** 38](#_Toc167093606)

[2.3.7. Phân tích Use case Tìm kiếm nhật ký 38](#_Toc167093607)

[**2.3.7.1.** **Biểu đồ lớp phân tích** 38](#_Toc167093608)

[**2.3.7.2.** **Biểu đồ trình tự** 39](#_Toc167093609)

[2.3.8. Phân tích Use case Tạo thời khóa biểu 39](#_Toc167093610)

[**2.3.8.1.** **Biểu đồ lớp phân tích** 39](#_Toc167093611)

[**2.3.8.2.** **Biểu đồ trình tự** 40](#_Toc167093612)

[2.3.9. Phân tích Use case Sửa thời khóa biểu 40](#_Toc167093613)

[**2.3.9.1.** **Biểu đồ lớp phân tích** 40](#_Toc167093614)

[**2.3.9.2.** **Biểu đồ trình tự** 41](#_Toc167093615)

[2.3.10. Phân tích Use case Xóa thời khóa biểu 41](#_Toc167093616)

[**2.3.10.1.** **Biểu đồ lớp phân tích** 41](#_Toc167093617)

[**2.3.10.2.** **Biểu đồ trình tự** 42](#_Toc167093618)

[2.3.11. Phân tích Use case Đổi ngôn ngữ 42](#_Toc167093619)

[**2.3.11.1.** **Biểu đồ lớp phân tích** 42](#_Toc167093620)

[**2.3.11.2.** **Biểu đồ trình tự** 43](#_Toc167093621)

[2.3.12. Phân tích Use case Đổi chủ đề 43](#_Toc167093622)

[**2.3.12.1.** **Biểu đồ lớp phân tích** 43](#_Toc167093623)

[**2.3.12.2.** **Biểu đồ trình tự** 44](#_Toc167093624)

[2.4. Thiết kế cơ sở dữ liệu 44](#_Toc167093625)

[2.4.1. Mô hình thực thể liên kết 44](#_Toc167093626)

[2.4.2. Thiết kế bảng cơ sở dữ liệu 45](#_Toc167093627)

[2.5. Thiết kế giao diện 47](#_Toc167093628)

[2.5.1. Thiết kế giao diện giả lập luồng tạo nhật ký 47](#_Toc167093629)

[2.5.2. Thiết kế giao diện giả lập luồng thao tác tại màn Cài đặt 48](#_Toc167093630)

[2.5.3. Thiết kế giao diện giả lập luồng tạo thời khóa biểu 48](#_Toc167093631)

[**Chương 3 KẾT QUẢ ĐẠT ĐƯỢC** 49](#_Toc167093632)

[3.1. Cài đặt 49](#_Toc167093633)

[3.2. Kết quả đạt được 51](#_Toc167093634)

[3.2.1. Giao diện các màn hình trên ứng dụng 51](#_Toc167093635)

[3.2.1.1. Giao diện màn Splash 51](#_Toc167093636)

[3.2.1.2. Giao diện màn Intro 51](#_Toc167093637)

[3.2.1.3. Giao diện màn Language 52](#_Toc167093638)

[3.2.1.4. Giao diện màn hình chính 52](#_Toc167093639)

[3.2.1.5. Giao diện màn Tạo nhật ký 53](#_Toc167093640)

[3.2.1.6. Giao diện màn Tạo nhắc hẹn 53](#_Toc167093641)

[3.2.1.7. Giao diện màn Thời khóa biểu 54](#_Toc167093642)

[3.2.1.8. Giao diện màn Thống kê cảm xúc 54](#_Toc167093643)

[3.2.1.9. Giao diện màn Tạo Profile 55](#_Toc167093644)

[3.2.1.10. Giao diện màn Tạo mật khẩu 55](#_Toc167093645)

[3.2.1.11. Giao diện màn Cài đặt 56](#_Toc167093646)

[3.2.1.12. Giao diện màn Chọn chủ đề 56](#_Toc167093647)

[3.2.2. Kết quả kiểm thử 57](#_Toc167093648)

[3.3. Đánh giá kết quả 59](#_Toc167093649)

[3.4. Hướng phát triển 59](#_Toc167093650)

[**KẾT LUẬN** 60](#_Toc167093651)

[**TÀI LIỆU THAM KHẢO** 61](#_Toc167093652)

# **DANH MỤC HÌNH ẢNH**

[Hình 1.1. Mô hình MVVM 10](#_Toc166756092)

[Hình 1.2. Thành phần Room Database 13](#_Toc166756093)

[Hình 2. 1. Biểu đồ Use case 19](#_Toc167091770)

[Hình 2. 2 Biểu đồ lớp 32](#_Toc167091771)

[Hình 2. 3. Biểu đồ lớp phân tích Use case Tạo profile 32](#_Toc167091772)

[Hình 2. 4. Biểu đồ trình tự Use case Tạo profile 33](#_Toc167091773)

[Hình 2. 5. Biểu đồ lớp phân tích Use case Tạo mật khẩu 33](#_Toc167091774)

[Hình 2. 6. Biểu đồ trình tự Use case Tạo mật khẩu 34](#_Toc167091775)

[Hình 2. 7. Biểu đồ lớp phân tích Use case Tạo nhật ký 34](#_Toc167091776)

[Hình 2. 8. Biểu đồ trình tự Use case Tạo nhật ký 35](#_Toc167091777)

[Hình 2. 9. Biểu đồ lớp Use case Thống kê cảm xúc 35](#_Toc167091778)

[Hình 2. 10. Biểu đồ trình tự Use case Thống kê cảm xúc 36](#_Toc167091779)

[Hình 2. 11. Biểu đồ lớp phân tích Use case Sửa nhật ký 36](#_Toc167091780)

[Hình 2. 12. Biểu đồ trình tự Use case Sửa nhật ký 37](#_Toc167091781)

[Hình 2. 13. Biểu đồ lớp phân tích Use case Xóa nhật ký 37](#_Toc167091782)

[Hình 2. 14. Biểu đồ trình tự Use case Xóa nhật ký 38](#_Toc167091783)

[Hình 2. 15. Biểu đồ lớp phân tích Use case Tìm kiếm nhật ký 38](#_Toc167091784)

[Hình 2. 16. Biểu đồ trình tự Use case Tìm kiếm nhật ký 39](#_Toc167091785)

[Hình 2. 17. Biểu đồ lớp phân tích Use case Tạo thời khóa biểu 39](#_Toc167091786)

[Hình 2. 18. Biểu đồ trình tự Use case Tạo thời khóa biểu 40](#_Toc167091787)

[Hình 2. 19. Biểu đồ lớp Use case Sửa thời khóa biểu 40](#_Toc167091788)

[Hình 2. 20. Biểu đồ trình tự Use case Sửa thời khóa biểu 41](#_Toc167091789)

[Hình 2. 21. Biểu đồ lớp Use case Xóa thời khóa biểu 41](#_Toc167091790)

[Hình 2. 22. Biểu đồ trình tự Use case Xóa thời khóa biểu 42](#_Toc167091791)

[Hình 2. 23. Biểu đồ lớp phân tích Use case Đổi ngôn ngữ 42](#_Toc167091792)

[Hình 2. 24. Biểu đồ trình tự Use case Đổi ngôn ngữ 43](#_Toc167091793)

[Hình 2. 25. Biểu đồ trình tự Use case Đổi chủ đề 43](#_Toc167091794)

[Hình 2. 26. Biểu đồ trình tự Use case Đổi chủ đề 44](#_Toc167091795)

[Hình 2. 27. Biểu đồ mô tả cấu trúc cơ sở dữ liệu 44](#_Toc167091796)

[Hình 2. 28. Giao diện luồng tạo nhật ký 47](#_Toc167091797)

[Hình 2. 29. Giao diện luồng thao tác tại màn Cài đặt 48](#_Toc167091798)

[Hình 2. 30. Giao diện luồng tạo thời khóa biểu 48](#_Toc167091799)

[Hình 3. 1 Bật chế độ gỡ lỗi qua USB trên thiết bị Android 49](#_Toc167091818)

[Hình 3. 2 Chọn thiết bị Android để chạy 49](#_Toc167091819)

[Hình 3. 3 Tạo file APK 50](#_Toc167091820)

[Hình 3. 4 File APK được lưu trong thư mục máy 50](#_Toc167091821)

[Hình 3. 5 Giao diện màn Splash 51](#_Toc167091822)

[Hình 3. 6 Giao diện màn Intro 51](#_Toc167091823)

[Hình 3. 7 Giao diện màn Ngôn ngữ 52](#_Toc167091824)

[Hình 3. 8 Giao diện màn hình chính 52](#_Toc167091825)

[Hình 3. 9 Giao diện màn hình Tạo nhật ký 53](#_Toc167091826)

[Hình 3. 10 Giao diện màn hình Tạo nhắc hẹn 53](#_Toc167091827)

[Hình 3. 11 Giao diện màn hình Tạo thời khóa biểu 54](#_Toc167091828)

[Hình 3. 12 Giao diện màn hình Thống kê cảm xúc 54](#_Toc167091829)

[Hình 3. 13 Giao diện màn Tạo Profile 55](#_Toc167091830)

[Hình 3. 14 Giao diện màn hình Tạo mật khẩu bảo mật 55](#_Toc167091831)

[Hình 3. 15 Giao diện màn hình Cài đặt 56](#_Toc167091832)

[Hình 3. 16 Giao diện màn hình chọn chủ đề 56](#_Toc167091833)

# **DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Viết tắt** | ***Dịch nghĩa*** |
| 1 | MVVM | *Model – View – ViewModel* |
| 2 | SQL | *Structured Query Language* |
| 3 | API | *Application Programming Interface* |
| 4 | XML | *Extensible Markup Language* |
| 5 | IDE | *Integrated Development Environment* |
| 6 | APK | *Android Package Kit* |
| 7 | GPS | *Global Positioning System* |
| 8 | MVC | *Model-View-Controller* |
| 9 | SDK | *Software Development Kit* |
| 10 | NDK | *Native Development Kit* |
| 11 | JVM | *Java Virtual Machine* |
| 12 | LLVM | *Low Level Virtual Machine* |
| 13 | MVP | *Model – View – Presenter* |
| 14 | XAML | *Xtensible Application Markup Language* |
| 15 | UI | *User Interface* |
| 16 | JDK | *Java Development Kit* |
| 17 | USB | *Universal Serial Bus* |

# **DANH MỤC BẢNG BIỂU**

[Bảng 2. 1 Các ca sử dụng theo từng tác nhân của hệ thống 18](#_Toc167093653)

[Bảng 2. 2 Mô tả cơ sở dữ liệu bảng NhatKy 45](#_Toc167093654)

[Bảng 2. 3 Mô tả cơ sở dữ liệu bảng ThoiKhoaBieu 45](#_Toc167093655)

[Bảng 2. 4 Mô tả cơ sở dữ liệu bảng NhacHen 46](#_Toc167093656)

[Bảng 2. 5 Mô tả cơ sở dữ liệu bảng ChuDe 46](#_Toc167093657)

[Bảng 2. 6 Mô tả cơ sở dữ liệu bảng NgonNgu 46](#_Toc167093658)

[Bảng 2. 7 Mô tả cơ sở dữ liệu bảng MatKhau 47](#_Toc167093659)

[Bảng 2. 8 Mô tả cơ sở dữ liệu bảng CauHoiBaoMat 47](#_Toc167093660)

[Bảng 3. 1. Bảng kết quả kiểm thử ứng dụng Viết nhật ký hàng ngày cho sinh viên 57](#_Toc166760230)

# 

# **MỞ ĐẦU**

1. **Lý do chọn đề tài**

Cuộc sống của sinh viên đầy rẫy những bận rộn, áp lực và đôi khi là sự lạc lõng giữa các môn học, hoạt động xã hội và công việc bán thời gian. Trong tình hình này, việc duy trì sự cân bằng giữa học tập và cuộc sống cá nhân trở nên cực kỳ quan trọng. Chính vì lý do đó, em đã quyết định chọn đề tài “**XÂY DỰNG ỨNG DỤNG VIẾT NHẬT KÝ ĐỜI SỐNG HÀNG NGÀY CHO SINH VIÊN”** cho đồ án tốt nghiệp của mình để củng cố kiến thức học được tại trường cũng như tạo một ứng dụng hữu ích cho các bạn sinh viên.

Một trong những lý do chính là để giúp sinh viên ghi lại những trải nghiệm, suy nghĩ và cảm xúc hàng ngày một cách dễ dàng và bảo mật. Ứng dụng này sẽ hoạt động hoàn toàn offline, không chia sẻ thông tin với bất kỳ ai khác, mang lại sự yên tâm tuyệt đối cho người dùng.

Cuối cùng, ứng dụng này không chỉ là một công cụ hữu ích mà còn là người bạn đồng hành đáng tin cậy của sinh viên trong hành trình học tập và phát triển cá nhân của họ. Em hy vọng rằng ứng dụng này sẽ mang lại những lợi ích thực sự và giúp sinh viên trải qua những ngày học tập một cách suôn sẻ và hiệu quả hơn.

1. **Mục tiêu của đề tài**

Đề tài: Xây dựng ứng dụng viết nhật ký hàng ngày cho sinh viên đáp ứng được những mục tiêu sau:

* Hiểu những kiến thức nền tảng cần thiết về xây dựng giao diện App Mobile, phân tích thiết kế hệ thống, cơ sở dữ liệu.
* Hiểu và sử dụng ngôn ngữ lập trình để phát triển ứng dụng di động.
* Phát triển một ứng dụng di động cho sinh viên viết nhật ký hàng ngày, cung cấp cho người dùng khả năng ghi lại những trải nghiệm, suy nghĩ và cảm xúc hàng ngày một cách dễ dàng và tiện lợi.
* Cung cấp một công cụ hữu ích cho việc học tập và phát triển cá nhân.

1. **Đối tượng và phạm vi**

* Đối tượng:
  + Ngôn ngữ lập trình Kotlin
  + Framework Spring Boot
  + Công cụ lập trình Android Studio
  + Mô hình MVVM
  + Cơ sở dữ liệu Room Database
* Phạm vi:
  + Ghi chép nhật ký hàng ngày
  + Tạo thời khóa biểu và nhắc hẹn

1. **Phương pháp nghiên cứu**

Phương pháp nghiên cứu về mặt lý thuyết

* Nghiên cứu và tiến hành thu thập các tài liệu có liên quan đến đề tài.
* Tổng hợp các tài liệu đã thu thập và tiến hành phân tích.
* Chọn lọc các tài liệu nghiên cứu để báo cáo đồ án tốt nghiệp.

Phương pháp nghiên cứu trong thực nghiệm

* Tìm hiểu cách thức, tiêu chí kiểm thử chức năng.
* Phân tích các ứng dụng có chức năng tương tự.
* Phân tích và tiến hành xây dựng ứng dụng theo phân tích và yêu cầu thực tế.
* Tìm hiểu ngôn ngữ lập trình Kotlin, framework Spring Boot, công cụ lập trình Android Studio, mô hình MVVM, cơ sở dữ liệu Room Database
* Áp dụng công nghệ vào xây dựng hệ thống.
* Tổng hợp các kiến thức đã học và hoàn thành báo cáo.

**Chương 1 TỔNG QUAN**

* 1. Cơ sở lý thuyết
     1. Xác định yêu cầu
* Khảo sát, phân tích, thiết kế, xây dựng ứng dụng
* Tìm hiểu ngôn ngữ lập trình
* Xây dựng ứng dụng
  + 1. Tổng quan về lập trình ứng dụng di động trên nền tảng Android
       1. Giới thiệu về lập trình ứng dụng di động

Lập trình ứng dụng di động (Mobile App) là việc liên quan đến quá trình thiết kế, xây dựng và phát triển các phần mềm hoạt động trên các thiết bị di động.

Mỗi chương trình sẽ được thiết kế với những chức năng riêng biệt, nhằm đáp ứng tốt nhất các mục đích sử dụng của người dùng trên thị trường, có thể là phục vụ cho việc học, giải trí hay mua sắm online, …

Trong quá trình lập trình ứng dụng di động, các lập trình viên sẽ phải nhờ đến sự hỗ trợ của các ngôn ngữ chuyên dùng, để có thể dễ dàng tạo ra được một chương trình thỏa mãn nhu cầu sử dụng của người dùng.

Hiện nay, lập trình ứng dụng di động thường tập trung vào quá trình phát triển phần mềm hoạt động trên hai hệ điều hành chính, đó là Android – các phần mềm sẽ được cung ứng thông qua CH Play.

* + - 1. Ngôn ngữ sử dụng

Lập trình ứng dụng di động cho Android

* Ngôn ngữ lập trình Java: Là ngôn ngữ được sử dụng rộng rãi, phục vụ chủ yếu cho việc xây dựng và phát triển ứng dụng trên Android. Ngoài ra, Java còn có thể hỗ trợ lập trình bất kỳ loại ứng dụng di động nào, mang nhiều tính năng hiện đại và đồng thời cho phép thực hiện chỉnh sửa, cập nhật ứng dụng dễ dàng.
* Ngôn ngữ lập trình Kotlin: Là một ngôn ngữ luôn nhận được sự ưu tiên của các lập trình viên khi xây dựng một ứng dụng vận hành trên Android. Kotlin nổi tiếng là ngôn ngữ dễ học, dễ tương tác và trực quan hơn, nhờ tính đơn giản và ngắn gọn. Kotlin được tạo ra với mục đích giải quyết những hạn chế còn tồn tại khi làm việc với Java.
  + - 1. Giới thiệu về hệ điều hành Android

Android là hệ điều hành mã nguồn mở, dựa trên Linux Kernel, dành cho các thiết bị di động có màn hình cảm ứng như: điện thoại, máy tính bảng, đồng hồ thông minh, máy nghe nhạc, …

* Ưu điểm
* Kho ứng dụng đa dạng
* Mẫu mã đa dạng
* Có thể mở rộng bộ nhớ bằng thẻ
* Khả năng tùy biến cao có thể chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google.
* Được nhiều người dùng ưa chuộng
* Nhược điểm
* Nhiều ứng dụng chạy ngầm
* Một số ứng dụng chưa được tối ưu hóa tốt
* Một số ứng dụng có chất lượng kém
* Viruss dễ xâm nhập
  + 1. Công cụ lập trình Android Studio
       1. Giới thiệu về Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) các ứng dụng Android. Phần mềm này đã dựa trên IntelliJ IDEA – môi trường phát triển của phần mềm Java.

Để giúp phát triển được ứng dụng bên trong hệ điều hành Android thì cần phải có phần mềm hệ thống công nghệ được xây dựng dựa trên Gradle, trình giả lập, tích hợp Github, mẫu mã. Mỗi một dự án ở trong Android Studio có 1 hoặc nhiều các phương thức, mã nguồn và các tệp tài nguyên. Những phương thức này sẽ bao gồm mô–đun cho ứng dụng, thư viện và Google App Engine.

Công cụ này dùng Instant Push để có thể giúp đẩy những thay đổi về mã, tài nguyên và các ứng dụng đang chạy. Trình soạn thảo sẽ hỗ trợ các nhà phát triển ứng dụng viết mã, cung cấp các khả năng hoàn thành, khúc xạ và tiến hành phân tích mã. Các ứng dụng được xây dựng trong công cụ này sẽ được phiên dịch ở định dạng APK để có thể gửi lên cửa hàng Google Play.

* + - 1. Các tính năng hàng đầu của Android Studio
* Cung cấp công cụ nhanh nhất để tạo ứng dụng trên nền tảng Android

Android Studio là một trong những**IDE**chính thức của hệ điều hành Android. Nó được tạo ra nhằm mục đích đẩy nhanh sự phát triển và xây dựng các ứng dụng chất lượng cao trên các thiết bị Android. Với công cụ hỗ trợ nhanh và tốt nhất, thì sẽ giúp cho các nhà phát triển rất nhiều trong quá trình họ tạo ra được một ứng dụng dành cho Android hoàn chỉnh.

* Trình giả lập, mô phỏng nhanh chóng với nhiều tính năng hữu ích

Trên Android Studio sẽ hỗ trợ **trình giả lập Android** cho người dùng có thể**cài đặt và khởi động**ứng dụng nhanh hơn trên thiết bị thực tế. Đồng thời sẽ cho phép tạo ra được các mẫu và thử nghiệm ứng dụng này trên các thiết bị Android khác như là điện thoại, máy tính bảng, Android Wear, TV, …

Có thể mô phỏng được một loạt những tính năng phần cứng như là GPS, độ trễ mạng, các cảm biến chuyển động, thông qua các tính năng trong công cụ này.

* Trình chỉnh sửa mã thông minh, hỗ trợ nhiều ngôn ngữ lập trình

Trình chỉnh sửa mã trong công cụ Android Studio sẽ giúp bạn có thể viết mã được tốt hơn. Đẩy nhanh tốc độ và làm việc năng xuất hơn bằng cách cung cấp các tính năng hoàn thành, tái cấu trúc và phân tích mã ở dạng nâng cao. Ngoài ra công cụ này còn hỗ trợ cả **C++ và NDK**, đi kèm theo là các tác vụ chỉnh sửa tệp.

* Hệ thống xây dựng mạnh mẽ và linh hoạt

Android Studio sẽ cung cấp cho người dùng tính năng tự động hóa bản dựng và có thể quản lý sự phụ thuộc và cấu hình bản dựng có thể tùy chỉnh được. Bạn sẽ có thể định được cấu hình dự án của mình để đảm bảo rằng các thư viện cục bộ đều được lưu trữ.

Đồng thời sẽ xác định các biến thể bản dựng bao gồm các mã và tài nguyên khác nhau. Đồng thời bạn cũng sẽ có thể áp dụng cấu hình ký ứng dụng và thu nhỏ đi các mã khác nhau.

* Cung cấp mẫu dự án và mã giúp xây dựng tính năng thông qua GitHub

Android Studio sẽ bao gồm tất cả các mẫu dự án và mã giúp dễ dàng thêm được các mã mẫu đã được thiết lập tốt như là ngăn điều hướng hay là xem máy nhắn tin.

Người dùng có thể bắt đầu với một mẫu mã hay thậm chí có thể nhấp chuột phải vào API trong tình chỉnh sửa và chọn tìm lại Mã mẫu để tìm kiếm các ví dụ. Bạn cũng có thể nhập các ứng dụng đầy đủ chức năng vào công cụ từ GitHub.

* Trình chỉnh sửa bố cục trực quan

Khi làm việc có các tệp bố cục XML, thì công cụ Android Studio sẽ cung cấp **trình chỉnh sửa trực quan**để người dùng kéo thả, tạo bố cục mới cực dễ dàng. Trình chỉnh sửa bố cục sẽ được xây dựng đồng thời cùng với cả **API ConstraintLayout.**

Người dùng có thể nhanh chóng tạo ra được một bố cục thích ứng với các kích thước màn hình khác nhau. Chỉ cần kéo các chế độ xem vào vị trí và thêm các ràng buộc bố cục trong công cụ là xong.

* Trình phân tích APK một cách tổng thể

Người dùng sẽ có thể sử dụng trình phân tích APK, để tìm ra được nội dung APK phù hợp. Nó sẽ cho biết kích thước của các thành phần để người dùng có thể xác định cách giảm kích thước tổng thể.

Công cụ cho phép xem trước được các nội dung đã được đóng gói, kiểm tra lại các tệp DEX. Để có thể khắc phục kịp thời các sự cố Multidex và có sự so sánh khác biệt giữa 2 APK với nhau.

* Cung cấp khung phân tích và kiểm tra ứng dụng toàn diện

Android Studio sẽ cung cấp thêm các công cụ mở rộng để người dùng có thể kiểm tra các ứng dụng của mình với JUnit 4 và khung kiểm tra giao diện chức năng. Đồng thời **trình ghi thử nghiệm Espresso**, sẽ cho phép người dùng có thể tạo mã kiểm tra giao diện thông qua việc ghi lại tương tác của bạn với ứng dụng ở trên thiết bị, trình mô phỏng. Bạn sẽ có thể chạy thử được thí nghiệm của mình trên các thiết bị hoặc trình mô phỏng liên tục.

* Khả năng đồng bộ hóa dữ liệu cao

Android Studio còn trang bị tính năng **Áp dụng thay đổi**, cho phép người dùng có thể thay đổi mã, tài nguyên vào ứng dụng đang chạy mà không cần khởi động lại. Tính linh hoạt sẽ giúp người dùng kiểm soát mức độ ứng dụng khởi động lại và muốn triển khai, chạy thử nghiệm lại các thay đổi nhỏ. Đồng thời vẫn duy trì được trạng thái hiện tại của thiết bị.

* Môi trường tối ưu hóa cho tất cả các thiết bị Android

Android Studio sẽ cung cấp cho người dùng một môi trường thống nhất, nơi bạn có thể tạo được các ứng dụng điện thoại, máy tính bảng, …. Các mô đun mã đã có mã cấu trúc sẽ cho phép bạn chia dự án của mình thành những đơn vị chức năng khác nhau mà bạn có thể xây dựng, kiểm tra, gỡ lỗi chúng một cách độc lập.

* Kết nối và phân tích ứng dụng theo thời gian thực với Firebase

**Trợ lý Firebas**e sẽ giúp cho người dùng có thể kết nối ứng dụng của mình với Firebase. Có thể thêm được những dịch vụ như là phân tích, xác thực, thông báo, … thông qua các quy trình từng ở ngay trên công cụ Android Studio.

* Tích hợp Google Cloud Platform và Google App Engine

Những công cụ tích hợp cho **Google Cloud Platform** cũng sẽ giúp người dùng tích hợp thêm vào ứng dụng của mình. Ví dụ như là dịch vụ Google Cloud Endpoints, các mô đun dự án đã được thiết kế đặc biệt dành cho Google App Engine. 

* + 1. Ngôn ngữ lập trình Kotlin
       1. Giới thiệu về ngôn ngữ lập trình Kotlin

Kotlin là một ngôn ngữ lập trình kiểu tĩnh chạy trên máy ảo Java (JVM) và có thể được biên dịch sang mã nguồn Java hay sử dụng cơ sở hạ tầng trình biên dịch LLVM.

Kotlin được xây dựng và phát triển bởi JetBrains và xuất hiện lần đầu vào năm 2011.

Kể từ Android Studio 3.0 (phát hành vào tháng 10 năm 2017), Kotlin được Google hỗ trợ đầy đủ để sử dụng cho việc lập trình ứng dụng cho hệ điều hành Android của họ, và được nhúng trực tiếp vào trong gói cài đặt của IDE đó để thay thế cho trình biên dịch Java tiêu chuẩn.

* + - 1. Tính năng của ngôn ngữ lập trình Kotlin
* Cung cấp chức năng Thư viện Chuẩn có thể giúp các nhà phát triển Kotlin theo nhiều cách, đặc biệt các hàm trong ngôn ngữ này cũng dễ dàng mở rộng để phục vụ tốt hơn cho các công việc trong lập trình.
* Cung cấp tính năng Lazy Loading, hỗ trợ tải những nội dung cần thiết ban đầu cho ứng dụng và giúp làm giảm khá nhiều thời gian tải và khởi chạy cho chương trình.
* Hoạt động tốt và hiệu quả hơn với Null Pointer Exception, mang đến nhiều lợi ích sử dụng cho nhà lập trình trong giai đoạn đầu quá trình phát triển.
* Biết cách xử lý các tập hợp bất biến, có sẵn các API với chức năng phong phú sẽ tự động trả về dưới dạng bộ sưu tập và được đặc trưng bởi chức năng giống nhau.
  + - 1. Ưu nhược điểm của Kotlin
* Ưu điểm
* Hoạt động với mã nguồn mở, sở hữu cú pháp ngắn gọn và mang lại sự thuận tiện trong quá trình debug.
* Các đoạn mã code của Kotlin vô cùng đơn giản, dễ học và đọc hiểu.
* Hoạt động tương thích gần như 100% với Java, mọi việc thực hiện trong Java đều dễ dàng ứng dụng vào Kotlin.
* Kotlin là một ngôn ngữ an toàn, sở hữu cơ chế null-safety, sẽ không còn khái niệm Null Pointer Exceptions xuất hiện.
* Dễ dàng sử dụng toàn bộ những Framework và thư viện sẵn trong của Java. Đặc biệt, mang lại sự thuận tiện khi kết hợp với Maven, Gradle cùng các hệ thống build khác.
* Là công cụ hỗ trợ tuyệt vời cho các công cụ và IDE.
* Nhược điểm
* Phần hàm của Kotlin vẫn chưa được cập nhật kiểu Aliases, nên khi bắt buộc làm việc với hàm này các lập trình viên phải viết thủ công, khiến cho phần mã nguồn trở nên dư thừa.
* Còn đối với Class, nó luôn được mặc định là Class final và khi muốn nó trở thành định dạng Class thông thường, phải thực thi thêm từ khóa Open.
* Kotlin không thể tự mặc định kiểu dữ liệu với các định dạng nguyên thủy.
* Cộng đồng hỗ trợ người dùng còn khá hạn chế.
* Các pattern và các mẫu code sẽ khá khó đọc và hiểu khi mới làm quen.
  + 1. Mô hình MVVM

MVVM là viết tắt của Model - View - ViewModel, đây là mô hình hỗ trợ two-way data binding giữa View và View Model. Cụ thể mô hình MVVM được trình bày như sau:

* View

Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn, nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.

* Model

Cũng tương tự như trong mô hình MVC, Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.

* View Model

Là lớp trung gian giữa View và Model. View Model có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh thực hiện Data Binding, Command. Một điểm cần lưu ý là trong mô hình MVVM, các tầng bên dưới sẽ không biết được các thông tin gì về các tầng trên của nó.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu tượng

Mô tả được tạo tự động

Hình 1.1. Mô hình MVVM

* Một định nghĩa khác về ViewModel trong MVVM:

ViewModel sẽ đảm nhận công việc đồng bộ dữ liệu từ Model lên View. Mối quan hệ giữa View và View-Model là View sẽ được ánh xạ tới View Model nhưng ViewModel lại không biết thông tin gì về View. nó được ẩn giấu qua cách sử dụng Data-binding và cơ chế của mô hình Observer, một ViewModel có thể được ánh xạ từ nhiều View.

Thông thường khi sử dụng với MVVM chúng ta nên tạo 3 thư mục chính chứa các file code liên quan.

* Views

Tại đây chứa các file giao diện và mỗi file giao diện đều có class codebehind đi kèm. Đặc biệt file code-behind ta sẽ không sử dụng đến, mọi điều cần làm sẽ chuyển xuống class ViewModel. Tất nhiên là bạn có thể code trong file code-behind của XAML nhưng đồng nghĩa điều đó sẽ phá vỡ quy ước của MVVM. Bạn có thể khai báo thuộc tính datacontext hoặc vài thiết lập khác nhưng nên hạn chế tối thiểu code ở đây. Views được sử dụng để kết hợp với các mô hình MVVM,... Nó dùng để cung cấp một sự chia tách gọn gàng của khái niệm giữa UI và presentation logic và data.

* Models

Trong thư mục Models tạo các tầng chứa dữ liệu và bất kỳ liên kết validation, logic nghiệp vụ để chắc chắn tính toàn vẹn của data, bạn có thể tách ra như một Repositories khác, chúng được dùng như một phần của mô hình MVVM.

* ViewModels

Thông thường trong một file giao diện thì ta tạo ra một class View Models tương ứng (có đôi lúc ta tạo nhiều class phụ giúp tinh giản file code và gọi chúng trong class ViewModels chính).

ViewModels sẽ sử dụng các models nếu cần định nghĩa dữ liệu. Sự liên kết giữa View-ViewModel giúp chúng gửi và nhận dữ liệu, để hiểu rõ ta cần tìm hiểu các khái niệm về Binding, DataContext, Behaviors SDK, nhờ đó ta tách code-behind của View và đưa xuống View Model.

Ngoài ra một lớp ViewModels chứa presentation logic và state của ứng dụng. ViewModel cần chứa các chức năng của ứng dụng. ViewModels định nghĩa properties, commands và events để chuyển đổi controls trong view cần databind.

* Data binding

Data binding trong MVVM là điều không bắt buộc, một số implement chỉ đơn giản làm ViewModel như một lớp trung gian giữa Model-View, lớp này giữ nhiệm vụ format data hoặc mapping trạng thái của View. Tuy nhiên cách này theo mình khiến cho ViewModel trở thành Presenter và đưa kiến trúc này về MVP.

* Ưu điểm
* Thực hiện Unit testing bây giờ sẽ rất dễ dàng, vì bạn thực sự không phụ thuộc vào view.
* MVVM sẽ tạo sự tương tác hiệu quả giữa designer và developer.
* Tăng khả năng sử dụng lại các thành phần hay việc thay đổi giao diện chương trình mà không cần phải viết lại code quá nhiều.
* Phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì…
* Nhược điểm
* Khả năng duy trì khi view có thể gán cả biến và biểu thức, các logic không liên quan sẽ tăng dần theo thời gian, ảnh hưởng đến việc thêm code vào XML.
* Đối với dự án nhỏ việc áp dụng mô hình MVVM gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.
* Đối với dự án lớn hơn, nó gây khó khăn và mất thời gian để thiết kế các ViewModel.
* Việc liên kết dữ liệu cho tất cả các thành phần gây khó khăn trong việc debug khi cơ sở dữ liệu phức tạp.
  + 1. Room Database

Room database được phát triển và cải tiến từ SQLite. Room database giúp đơn giản hóa việc code và giảm thiểu các công đoạn liên quan tới cơ sở dữ liệu

Bản chất Room database là abstract layer gồm cơ sở dữ liệu chuẩn SQLite được Android thông qua.

Với 3 thành phần chính là:

* Database: đóng vai trò là điểm truy cập chính cho kết nối với cơ sở dữ liệu.
* Enity: Như là một table trong database, đại diện của một class Enity.
* DAO (Data Access Object): Chứa các method dùng để truy cập database, cơ bản sẽ là CRUD data.

A diagram of data access objects

Description automatically generated

Hình 1.2. Thành phần Room Database

* Ưu điểm
* Tóm tắt và dễ sử dụng
* Room database giúp đơn giản hóa việc tương tác với SQLite trên Android bằng cách cung cấp một lớp trừu tượng và chú thích. Nó loại bỏ sự cần thiết phải viết mã SQL thô, làm cho việc truy cập và thao tác dữ liệu trở nên dễ dàng và ít lỗi hơn.
* An toàn cho kiểu dữ liệu
* Room database giúp đảm bảo an toàn cho kiểu dữ liệu bằng cách kiểm tra biên dịch các truy vấn của bạn và tự động chuyển đổi các kiểu dữ liệu Java sang SQL. Điều này giúp giảm thiểu lỗi và làm cho mã của bạn dễ đọc và dễ bảo trì hơn.
* Hiệu suất
* Room database được tối ưu hóa cho hiệu suất và tận dụng tối đa sức mạnh của SQLite. Nó giúp cung cấp các tính năng như truy vấn bù đắp, bộ nhớ đệm và truy vấn bất đồng bộ để giúp bạn viết mã truy cập dữ liệu hiệu quả.
* Khả năng bảo trì
* Room database giúp giữ cho mã của bạn dễ bảo trì bằng cách cung cấp một API rõ ràng và nhất quán để truy cập dữ liệu. Điều này giúp bạn dễ dàng hiểu và sửa đổi mã của mình, ngay cả khi nó được viết bởi người khác.
* Tích hợp Livedata
* Room database tích hợp liền mạch với Livedata, một lớp khung Android để quản lý dữ liệu thay đổi. Điều này giúp bạn dễ dàng xây dựng các UI phản hồi dữ liệu thay đổi từ cơ sở dữ liệu.
* Nhược điểm
* Khả năng tùy chỉnh hạn chế
* Room database là một lớp trừu tượng trên SQLite, vì vậy nó có khả năng tùy chỉnh hạn chế hơn sơ với việc sử dụng SQLite trực tiếp. Điều này có thể gây ra vấn đề nếu bạn cần thực hiện các truy vấn phúc tạp hoặc truy cập các tính năng nâng cao của SQLite.
* Hiệu suất bổ sung
* Lớp trừu tượng của Room database có thể thêm một chút chi phí hiệu suất so với sử dụng SQLite trực tiếp. Tuy nhiên, trong hầu hết trường hợp, hiệu suất này không đáng kể.
* Sự phụ thuộc
* Room database là một thư viện của bên thứ ba, vì vậy bạn phụ thuộc và sự phát triển và bảo trì liên tục của nó. Nếu dự án Room database ngừng hoạt động, bạn có thể cần phải chuyển sang giải pháp thay thế khác.
  1. Khảo sát
     1. Phát biểu bài toán

Xây dựng ứng dụng viết nhật ký đời sống hàng ngày cho sinh viên cần phải đảm bảo tính thân thiện và dễ sử dụng, giúp sinh viên ghi lại những khoảnh khắc đáng nhớ trong cuộc sống học đường một cách dễ dàng và linh hoạt. Đồng thời, ứng dụng cần cung cấp các tính năng tiện ích như tạo và quản lý bài viết, chia sẻ trải nghiệm, gắn kết cộng đồng sinh viên. Thiết kế phải được tối ưu hóa để phản ánh phong cách trẻ trung, hiện đại và hấp dẫn, từ giao diện đến trải nghiệm người dùng, nhằm thu hút và giữ chân người dùng trong thế giới sống động của sinh viên.

* + 1. Mục đích

Cuộc sống của sinh viên đầy rẫy những bận rộn, áp lực và đôi khi là sự lạc lõng giữa các môn học, hoạt động xã hội và công việc bán thời gian. Trong tình hình này, việc duy trì sự cân bằng giữa học tập và cuộc sống cá nhân trở nên cực kỳ quan trọng. Chính vì lý do đó, em đã quyết định chọn đề tài “XÂY DỰNG ỨNG DỤNG VIẾT NHẬT KÝ ĐỜI SỐNG HÀNG NGÀY CHO SINH VIÊN”.

* + 1. Mô tả các yêu cầu
       1. Yêu cầu chức năng
* Tạo profile:
  + Mô tả: Người dùng có thể tạo UserName hiển thị ở trong ứng dụng.
  + Yêu cầu: Form tạo profile với trường thông tin là là UserName
* Tạo mật khẩu bảo mật:
  + Mô tả: Cho phép người dùng tạo mật khẩu mỗi khi vào ứng dụng nhằm tăng tính bảo mật.
  + Yêu cầu: Tính năng thiết lập và thay đổi mật khẩu, có câu hỏi bảo mật khi người dùng quên mật khẩu.
* Quản lý nhật ký:
  + Mô tả: Người dùng có thể tạo, chỉnh sửa, và xóa các mục nhật ký cá nhân. Mỗi mục nhật ký có thể bao gồm cảm xúc, tiêu đề, nội dung văn bản và ảnh đính kèm.
  + Yêu cầu:
    - Tạo nhật ký: Form để người dùng nhập tiêu đề, chọn cảm xúc (hệ thống cảm xúc có sẵn), viết nội dung và đính kèm ảnh.
    - Chỉnh sửa nhật ký: Tính năng cho phép người dùng thay đổi thông tin của nhật ký đã tạo.
    - Xóa nhật ký: Tính năng cho phép người dùng xóa nhật ký không còn cần thiết.
* Thống kê cảm xúc:
  + Mô tả: Cung cấp cho người dùng bảng thống kê phần trăm các cảm xúc đã ghi lại trong các mục nhật ký.
  + Yêu cầu:
    - Thu thập dữ liệu cảm xúc từ các mục nhật ký.
    - Hiển thị biểu đồ hoặc bảng thống kê phần trăm từng loại cảm xúc.
* Tạo nhắc hẹn:
  + Mô tả: Người dùng có thể thiết lập các nhắc nhở để viết nhật ký vào những thời điểm cụ thể.
  + Yêu cầu: Tính năng tạo nhắc hẹn với các tùy chọn về thời gian và tần suất (hàng ngày, hàng tuần, v.v.), cùng với thông báo nhắc nhở.
* Quản lý thời khóa biểu:
  + Mô tả: Người dùng có thể tạo và quản lý thời khóa biểu cá nhân, bao gồm tiêu đề, nội dung chi tiết và thời gian nhắc hẹn.
  + Yêu cầu:
    - Tạo thời khóa biểu: Form để người dùng nhập tiêu đề, nội dung và thời gian.
    - Chỉnh sửa thời khóa biểu: Tính năng cho phép người dùng thay đổi thông tin của thời khóa biểu.
    - Xóa thời khóa biểu: Tính năng cho phép người dùng xóa các mục thời khóa biểu không còn cần thiết.
* Thay đổi ngôn ngữ:
  + Mô tả: Ứng dụng hỗ trợ đa ngôn ngữ, cho phép người dùng thay đổi ngôn ngữ hiển thị trong ứng dụng theo ý thích.
  + Yêu cầu:
    - Danh sách ngôn ngữ có sẵn trong ứng dụng.
    - Tính năng chuyển đổi ngôn ngữ và lưu lại tùy chọn ngôn ngữ của người dùng.
* Thay đổi chủ đề:
  + Mô tả: Người dùng có thể thay đổi giao diện của màn hình Home theo các chủ đề có sẵn trong ứng dụng để cá nhân hóa trải nghiệm.
  + Yêu cầu:
    - Danh sách các chủ đề có sẵn trong ứng dụng.
    - Tính năng áp dụng chủ đề mới và lưu lại tùy chọn chủ đề của người dùng.
      1. Các yêu cầu khác
* Ứng dụng có nội dung hấp dẫn, đầy đủ, phù hợp với mọi đối tượng sử dụng, đặc biệt là đối tượng các bạn sinh viên.
* Được sắp xếp bố cục một cách hợp lý, tạo điều kiện cho người dùng dễ định hướng trong ứng dụng.
* Hình thức đẹp phù hợp với chủ đề, nội dung, thân thiện với người dùng
* Hoàn thiện các module chức năng cần thiết.

**Chương 2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

* 1. Mô hình use cases
     1. Xác định các tác nhân và các ca sử dụng

1. Nhóm tác nhân chính của hệ thống:

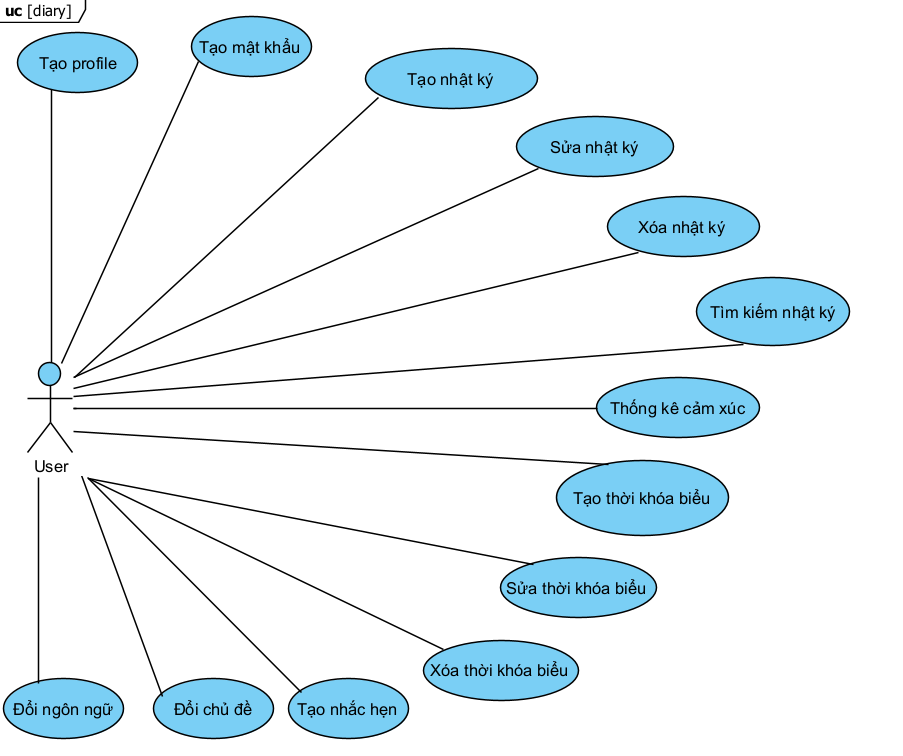
* Người dùng: Là người có nhu cầu sử dụng ứng dụng để ghi chú hoặc sử dụng những tính năng có trong ứng dụng đặc biệt là sinh viên. Người dùng có thể vào ứng dụng để ghi chú, tạo lịch hẹn, thay đổi ngôn ngữ, thay đổi chủ đề, tạo thời khóa biểu.

1. Xác định các các ca sử dụng theo từng tác nhân của hệ thống:

Bảng 2. 1 Các ca sử dụng theo từng tác nhân của hệ thống

|  |  |  |
| --- | --- | --- |
| Tác nhân | Ca sử dụng | Mô tả |
| Người dùng | Tạo profile | Cho phép người dùng đặt usename để hiển thị trong ứng dụng |
| Tạo mật khẩu | Cho phép người dùng đặt mã bảo vệ khi vào ứng dụng, tăng cường tính bảo mật |
| Quản lý nhật ký | Cho phép người dùng viết nhật ký hàng ngày, có thể kèm thêm ảnh và thêm cảm xúc, thêm, xóa nhật ký |
| Tạo thời khóa biểu | Cho phép người dùng tạo thời khóa biểu phục vụ cho việc học tập, làm việc kết hợp với tính năng nhắc hẹn |
| Tạo nhắc hẹn | Cho phép người dùng đặt lịch hẹn để viết nhật ký |
| Đổi ngôn ngữ | Cho phép người dùng đổi ngôn ngữ hiển thị trong ứng dụng |
| Đổi chủ đề | Cho phép người dùng đổi chủ đề hiển thị trong ứng dụng |

* + 1. Biểu đồ Use case



Hình 2. 1. Biểu đồ Use case

* + 1. Các đặc tả use case
       1. **Use case Tạo profile**
* Use case mô tả quy trình khi người dùng mới vào ứng dụng và cần tạo một hồ sơ mới bằng cách nhập tên người dùng.
* Luồng sự kiện

**Luồng cơ bản:**

* 1. Use case này bắt đầu khi người dùng mở ứng dụng lần đầu tiên.
  2. Hệ thống hiển thị giao diện yêu cầu người dùng nhập tên người dùng.
  3. Người dùng nhập tên của họ vào trường cần thiết trên giao diện.
  4. Người dùng xác nhận thông tin nhập và chọn "Tiếp tục"
  5. Hệ thống lưu tên người dùng và chuyển hướng đến giao diện chính của ứng dụng.
  6. Use case kết thúc

**Luồng rẽ nhánh:**

* 1. Nếu người dùng không nhập tên người dùng hoặc nhập một tên không hợp lệ:
     + Hệ thống hiển thị thông báo lỗi yêu cầu người dùng nhập tên người dùng hợp lệ.
     + Quay lại bước nhập tên người dùng.
  2. Nếu người dùng chọn "Hủy":
     + Hệ thống thoát khỏi quy trình tạo hồ sơ và đóng ứng dụng.
     + Use case kết thúc.
* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện: Người dùng vào ứng dụng lần đầu
* Hậu điều kiện: Hồ sơ mới được tạo và lưu trong cơ sở dữ liệu của ứng dụng, và người dùng được chuyển hướng đến giao diện chính của ứng dụng.
* Điểm mở rộng: Không có.
  + - 1. **Use case Tạo mật khẩu**
* Use case này mô tả quy trình khi người dùng tạo một mật khẩu bảo mật và đặt một câu hỏi bảo mật để cải thiện tính bảo mật của ứng dụng.
* Luồng sự kiện

**Luồng cơ bản:**

* 1. Use case này bắt đầu khi người dùng mở ứng dụng và chưa thiết lập mật khẩu bảo mật và câu hỏi bảo mật trước đó.
  2. Hệ thống hiển thị giao diện yêu cầu người dùng tạo một mật khẩu bảo mật.
  3. Người dùng nhập mật khẩu bảo mật của họ vào trường cần thiết trên giao diện.
  4. Hệ thống yêu cầu người dùng đặt một câu hỏi bảo mật (ví dụ: "Tên thú cưng đầu tiên của bạn là gì?").
  5. Người dùng nhập câu trả lời cho câu hỏi bảo mật và xác nhận mật khẩu.
  6. Hệ thống lưu mật khẩu bảo mật và câu hỏi bảo mật vào hồ sơ của người dùng và chuyển hướng đến giao diện chính của ứng dụng.
  7. Từ nay, mỗi khi người dùng mở ứng dụng, họ sẽ được yêu cầu nhập mật khẩu bảo mật trước khi truy cập được vào giao diện chính của ứng dụng.
  8. Use case kết thúc.

**Luồng rẽ nhánh:**

* 1. Nếu người dùng chọn bỏ qua việc tạo mật khẩu bảo mật và câu hỏi bảo mật và quay lại sau:
     + Hệ thống không yêu cầu người dùng nhập mật khẩu bảo mật và câu hỏi bảo mật và chuyển hướng đến giao diện chính của ứng dụng.
     + Use case kết thúc.
* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện: Người dùng đã cài đặt và mở ứng dụng
* Hậu điều kiện: Mật khẩu bảo mật mới và câu hỏi bảo mật được thiết lập và liên kết với tài khoản của người dùng.
* Điểm mở rộng: Không có.
  + - 1. **Use case Tạo nhật ký**
* Use case mô tả quy trình khi người dùng tạo một bài nhật ký mới trong ứng dụng.
* Luồng sự kiện

**Luồng cơ bản:**

* 1. Use case này bắt đầu khi người dùng vào màn hình chính (Home) của ứng dụng, nơi hiển thị danh sách các nhật ký đã được tạo.
  2. Người dùng chọn biểu tượng tạo mới.
  3. Hệ thống chuyển hướng người dùng đến màn hình thêm nhật ký.
  4. Màn hình thêm nhật ký hiển thị tùy chọn "Thêm cảm xúc".
  5. Người dùng chọn cảm xúc mong muốn từ danh sách.
  6. Sau khi chọn cảm xúc, người dùng chọn tiếp tục để sang màn Viết nhật ký.
  7. Tại màn Viết nhật ký, người dùng nhập các thông tin vào các trường tương ứng để tạo nhật ký.
  8. Người dùng chọn "Lưu".
  9. Hệ thống lưu nhật ký mới vào cơ sở dữ liệu và hiển thị thông báo xác nhận cho người dùng.
  10. Use case kết thúc và người dùng được chuyển về màn hình chính với nhật ký mới được hiển thị.

**Luồng rẽ nhánh:**

1. Tại bước số 8 ở luồng cơ bản, nếu người dùng không chọn "Lưu" mà thoát khỏi màn hình thêm nhật ký:
   * + Hệ thống không lưu nhật ký và không hiển thị thông báo xác nhận.
     + Người dùng được chuyển về màn hình chính.
     + Use case kết thúc.

* Các yêu cầu đặc biệt: Người dùng bắt buộc phải thêm cảm xúc trước khi viết nội dung nhật ký.
* Tiền điều kiện: Người dùng đã vào hệ thống và đang ở trong màn hình chính của ứng dụng.
* Hậu điều kiện: Nhật ký mới có thể được tạo và lưu vào cơ sở dữ liệu của ứng dụng, tùy thuộc vào lựa chọn của người dùng.
* Điểm mở rộng: Không có.
  + - 1. **Use case Sửa nhật ký**
* Use case mô tả quy trình khi người dùng sửa một bài nhật ký trong ứng dụng.
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi Người dùng ấn vào một Nhật ký đã có ở màn Home. Hệ thống sẽ điều hướng sang màn chi tiết nhật ký.
2. Người dùng chọn icon chỉnh sửa ở góc trên tại màn chi tiết nhật ký rồi lựa chọn Chỉnh sửa nhật ký.
3. Người dùng thao tác sửa các thông tin trong trường nhật ký và chọn “Tiếp tục” sau khi chỉnh sửa xong.
4. Hệ thống sẽ kiểm tra thông tin và cập nhật thông tin vào cơ sở dữ liệu.
5. Use case kết thúc và người dùng được điều hướng về màn hình Home của ứng dụng.

**Luồng rẽ nhánh:**

1. Ở bước 2 tại luồng cơ bản, người dùng hủy bỏ sửa nhật ký bằng cách quay lại màn trước.

* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện: Trên màn hình phải có nhật ký đã tạo.
* Hậu điều kiện: Hệ thống cập nhật các thông tin nhật ký vừa sửa lên màn hình chính.
* Điểm mở rộng: Không có.
  + - 1. **Use case Xóa nhật ký**
* Use case mô tả quy trình khi người dùng xóa một bài nhật ký trong ứng dụng.
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi Người dùng ấn vào một Nhật ký đã có ở màn Home. Hệ thống sẽ điều hướng sang màn chi tiết nhật ký.
2. Người dùng chọn icon Xóa ở góc trên tại màn chi tiết nhật ký.
3. Hệ thống sẽ hiện lên một thông báo xác nhận Xóa nhật ký.
4. Người dùng lựa chọn xác nhận Xóa nhật ký, hệ thống sẽ kiểm tra thông tin và cập nhật thông tin vào cơ sở dữ liệu.
5. Use case kết thúc và người dùng được điều hướng về màn hình Home của ứng dụng.

**Luồng rẽ nhánh:**

1. Ở bước 2 tại luồng cơ bản, người dùng lựa chọn không xóa nhật ký, thông báo mất đi và hiển thị vẫn ở màn xem chi tiết nhật ký.

* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện: Trên màn hình phải có nhật ký đã tạo.
* Hậu điều kiện: Hệ thống xóa nhật ký vừa chọn trên màn hình chính.
* Điểm mở rộng: Không có.
  + - 1. **Use case Tìm kiếm nhật ký**
* Use case mô tả quy trình khi người dùng tìm kiếm một bài nhật ký trong ứng dụng.
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi Người dùng ấn vào thanh tìm kiếm ở màn hình chính ứng dụng
2. Hệ thống hiển thị bàn phím, người dùng nhập tiêu đề nhật ký muốn tìm kiếm
3. Hệ thống sẽ hiển thị lên màn hình chính các nhật ký có tiêu đề tương ứng với tiêu đề được tìm kiếm
4. Use case kết thúc

**Luồng rẽ nhánh:**

1. Ở bước 2 tại luồng cơ bản, nếungười dùng nhập tiêu đề không trùng khớp với các tiêu đề trong nhật ký, hệ thống sẽ không trả về bản nhật ký nào.

* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện: Người dùng phải vào đến màn hình chính
* Hậu điều kiện: Hệ thống trả về những nhật ký có tiêu đề tương ứng với tiêu đề đã tìm kiếm
* Điểm mở rộng: Không có.
  + - 1. **Use case Thống kê cảm xúc**
* Use case mô tả quy trình khi người dùng muốn xem thống kê về các cảm xúc đã được thêm vào nhật ký trong ứng dụng.
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng đang ở trong màn hình chính (Home) của ứng dụng.
2. Người dùng chọn chức năng "Thống kê cảm xúc" hoặc biểu tượng thống kê trên giao diện.
3. Hệ thống truy xuất và tính toán số lượng các cảm xúc từ nhật ký đã được tạo trong cơ sở dữ liệu.
4. Hệ thống hiển thị các số liệu thống kê về các cảm xúc, chẳng hạn như số lượng cảm xúc tích lũy theo từng loại cảm xúc.
5. Người dùng có thể xem thông tin thống kê và quay lại màn hình chính bất cứ lúc nào.
6. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Nếu người dùng chưa thêm nhật ký:
   * + Hệ thống hiển thị thông báo cho người dùng rằng không có dữ liệu cảm xúc để thống kê:
     + Người dùng có thể quay lại màn hình chính.
     + Use case kết thúc.

* Các yêu cầu đặc biệt: Kiểm tra xem người dùng đã thêm ít nhất một nhật ký hay chưa để quyết định hiển thị thông báo hoặc tiếp tục quá trình thống kê cảm xúc.
* Tiền điều kiện: Người dùng đã vào hệ thống và đang ở trong màn hình chính của ứng dụng.
* Hậu điều kiện: Thống kê cảm xúc được hiển thị và người dùng có thể quay lại màn hình chính hoặc tiếp tục thêm nhật ký.
* Điểm mở rộng: Không có
  + - 1. **Use case Tạo thời khóa biểu**
* Use case này mô tả quy trình khi người dùng muốn xem thời khóa biểu của mình và có thể đặt báo thức nhắc nhở cho các sự kiện trong thời khóa biểu.
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng đang ở trong màn hình chính (Home) của ứng dụng.
2. Người dùng chọn biểu tượng thời khóa biểu trên giao diện.
3. Hệ thống hiển thị màn hình Thời khóa biểu của người dùng.
4. Người dùng chọn thứ trong tuần muốn thêm thời khóa biểu và nhấn nút thêm Thời khóa biểu
5. Hệ thống hiển thị màn hình tạo thời khóa biểu bao gồm: Tiêu đề, nội dung, thời gian, thông báo.
6. Người dùng tạo xong thời khóa biểu thì nhấn nút Lưu trên màn hình.
7. Hệ thống lưu thời khóa biểu vừa tạo và hiển thị trên màn hình.
8. Người dùng có thể quay lại màn hình chính hoặc tiếp tục xem và quản lý thời khóa biểu.
9. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Nếu thời khóa biểu của người dùng không có sự kiện nào:
   * + Màn hình Thời khóa biểu hiển thị trống sự kiện
     + Người dùng có thể quay lại màn hình chính hoặc tiếp tục thêm sự kiện vào thời khóa biểu.
     + Use case kết thúc.

* Các yêu cầu đặc biệt: Kiểm tra xem thời khóa biểu của người dùng có sự kiện nào hay không để quyết định hiển thị thông báo.
* Tiền điều kiện: Người dùng đã vào hệ thống và đang ở trong màn hình chính của ứng dụng.
* Hậu điều kiện: Người dùng có thể xem thời khóa biểu của mình và đặt báo thức cho các sự kiện trong thời khóa biểu.
* Điểm mở rộng: Không có.
  + - 1. **Use case Sửa thời khóa biểu**
* Use case này mô tả quy trình khi người dùng muốn sửa một thời khóa biểu
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng đang ở trong màn hình Thời khóa biểu của ứng dụng.
2. Người dùng chọn Thời khóa biểu muốn sửa
3. Hệ thống hiển thị thông tin thời khóa biểu được chọn
4. Người dùng sửa xong thời khóa biểu thì nhấn nút Lưu trên màn hình.
5. Hệ thống cập nhật thời khóa biểu vừa sửa và hiển thị trên màn hình.
6. Người dùng có thể quay lại màn hình chính hoặc tiếp tục xem và quản lý thời khóa biểu.
7. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Nếu thời khóa biểu của người dùng không có sự kiện nào:
   * + Màn hình Thời khóa biểu hiển thị trống sự kiện
     + Người dùng có thể quay lại màn hình chính hoặc tiếp tục thêm sự kiện vào thời khóa biểu.
     + Use case kết thúc.
2. Nếu người dùng sửa nhưng không lưu thời khóa biểu mà nhấn tắt:
   * + Hệ thống hiển thị màn hình Thời khóa biểu
     + Thời khóa biểu được chọn vẫn giữ nguyên

* Các yêu cầu đặc biệt: Kiểm tra xem thời khóa biểu của người dùng có sự kiện nào hay không để quyết định hiển thị thông báo.
* Tiền điều kiện: Người dùng đã vào hệ thống và đang ở trong màn hình chính của ứng dụng.
* Hậu điều kiện: Người dùng có thể xem thời khóa biểu của mình và đặt báo thức cho các sự kiện trong thời khóa biểu.
* Điểm mở rộng: Không có.
  + - 1. **Use case Xóa thời khóa biểu**
* Use case này mô tả quy trình khi người dùng muốn xóa một thời khóa biểu
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng đang ở trong màn hình Thời khóa biểu của ứng dụng.
2. Người dùng chọn Thời khóa biểu muốn sửa
3. Hệ thống hiển thị thông tin thời khóa biểu được chọn
4. Người dùng sửa xong thời khóa biểu thì nhấn nút Lưu trên màn hình.
5. Hệ thống cập nhật thời khóa biểu vừa sửa và hiển thị trên màn hình.
6. Người dùng có thể quay lại màn hình chính hoặc tiếp tục xem và quản lý thời khóa biểu.
7. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Nếu thời khóa biểu của người dùng không có sự kiện nào:
   * + Màn hình Thời khóa biểu hiển thị trống sự kiện
     + Người dùng có thể quay lại màn hình chính hoặc tiếp tục thêm sự kiện vào thời khóa biểu.
     + Use case kết thúc.
2. Nếu người dùng sửa nhưng không lưu thời khóa biểu mà nhấn tắt:
   * + Hệ thống hiển thị màn hình Thời khóa biểu
     + Thời khóa biểu được chọn vẫn giữ nguyên

* Các yêu cầu đặc biệt: Kiểm tra xem thời khóa biểu của người dùng có sự kiện nào hay không để quyết định hiển thị thông báo.
* Tiền điều kiện: Người dùng đã vào hệ thống và đang ở trong màn hình chính của ứng dụng.
* Hậu điều kiện: Người dùng có thể xem thời khóa biểu của mình và đặt báo thức cho các sự kiện trong thời khóa biểu.
* Điểm mở rộng: Không có.
  + - 1. **Use case Đổi ngôn ngữ**
* Use case này cho phép người dùng thay đổi ngôn ngữ hiển thị của ứng dụng theo quốc gia mà họ chọn trong ứng dụng
* Luồng sự kiện

**Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng chọn chức năng "Thay đổi Ngôn ngữ" trong cài đặt của ứng dụng.
2. Hệ thống hiển thị danh sách các ngôn ngữ được hỗ trợ.
3. Người dùng chọn ngôn ngữ mới từ danh sách.
4. Hệ thống cập nhật ngôn ngữ hiển thị của ứng dụng thành ngôn ngữ mà người dùng đã chọn và lưu cài đặt.
5. Use case kết thúc.

**Luồng rẽ nhánh:**

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.

* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện: Người dùng đã vào hệ thống và đang ở trong màn hình chính của ứng dụng.
* Hậu điều kiện: Không có.
* Điểm mở rộng: Không có.
  + - 1. **Use case Đổi chủ đề**
* Use case này cho phép người dùng thay đổi chủ đề hiển thị của ứng dụng theo sở thích cá nhân.
* Luồng sự kiện

**Luồng cơ bản:**

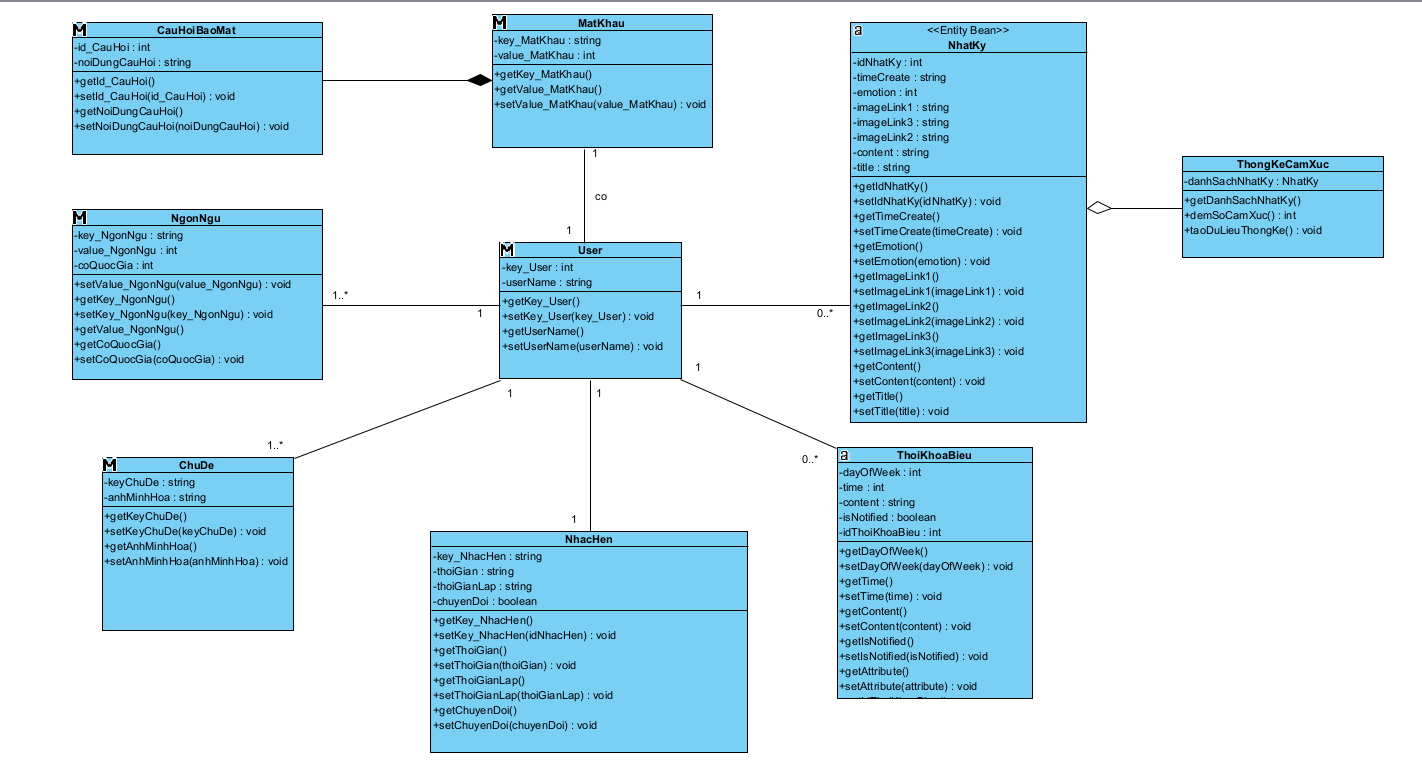
1. Use case này bắt đầu khi người dùng chọn chức năng "Thay đổi Chủ đề" trong cài đặt của ứng dụng.
2. Hệ thống hiển thị danh sách các chủ đề có sẵn hoặc các tùy chọn tùy chỉnh khác nhau cho người dùng chọn.
3. Người dùng chọn chủ đề mới từ danh sách hoặc tùy chỉnh chủ đề theo sở thích của mình.
4. Hệ thống cập nhật giao diện hiển thị của ứng dụng theo chủ đề mới và lưu cài đặt.
5. Use case kết thúc.

**Luồng rẽ nhánh:**

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.

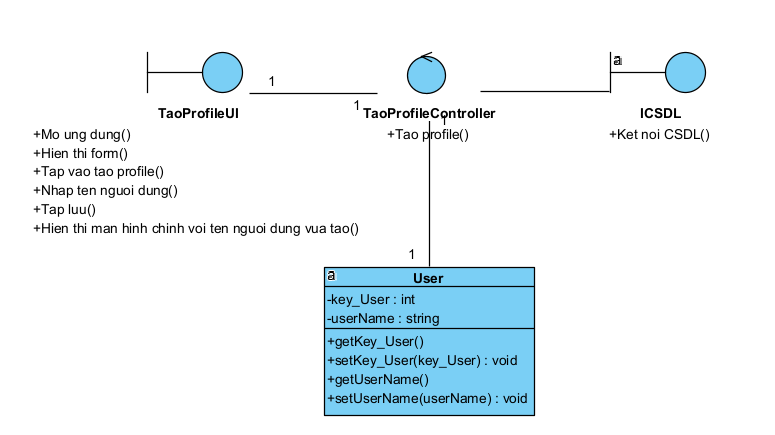
* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện: Người dùng đã vào hệ thống và đang ở trong màn hình chính của ứng dụng.
* Hậu điều kiện: Không có.
* Điểm mở rộng: Không có.

* 1. Biểu đồ lớp



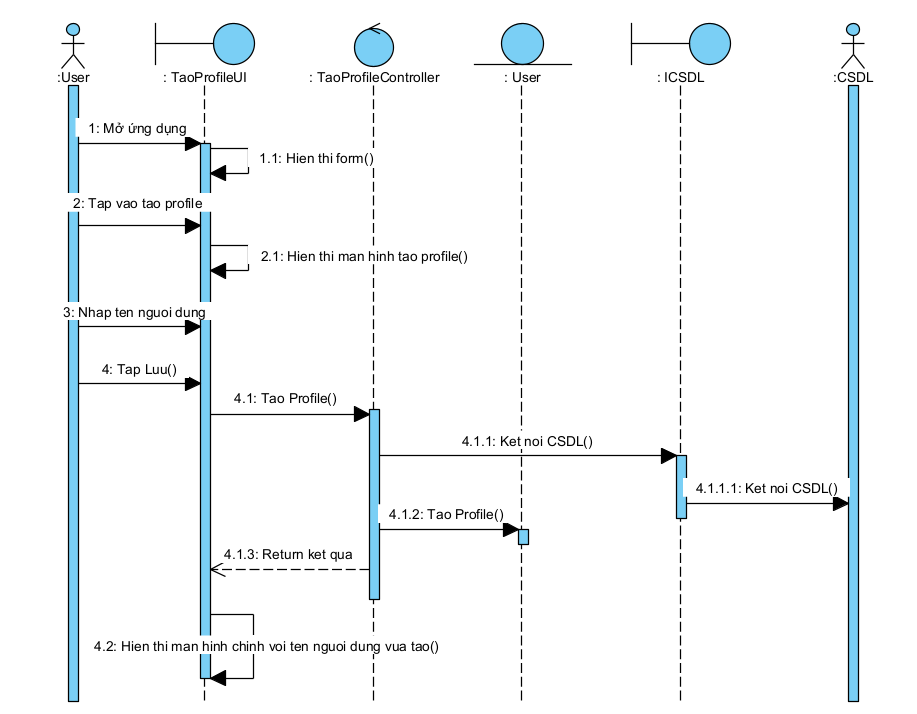
Hình 2. 2 Biểu đồ lớp

* 1. Phân tích use case
     1. Phân tích Use case Tạo Profile
        1. **Biểu đồ lớp phân tích**



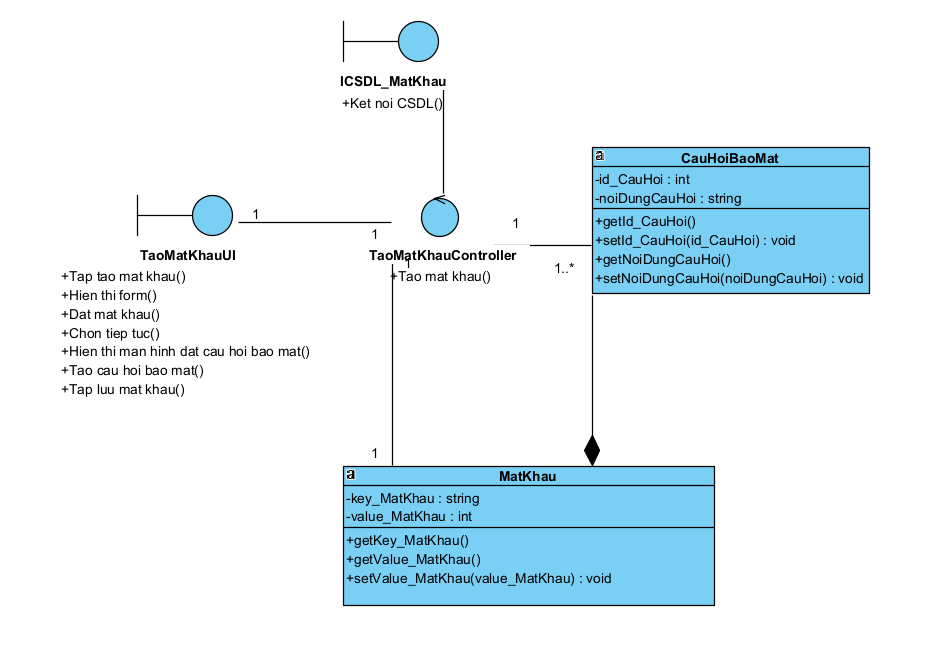
Hình 2. 3. Biểu đồ lớp phân tích Use case Tạo profile

* + - 1. **Biểu đồ trình tự**



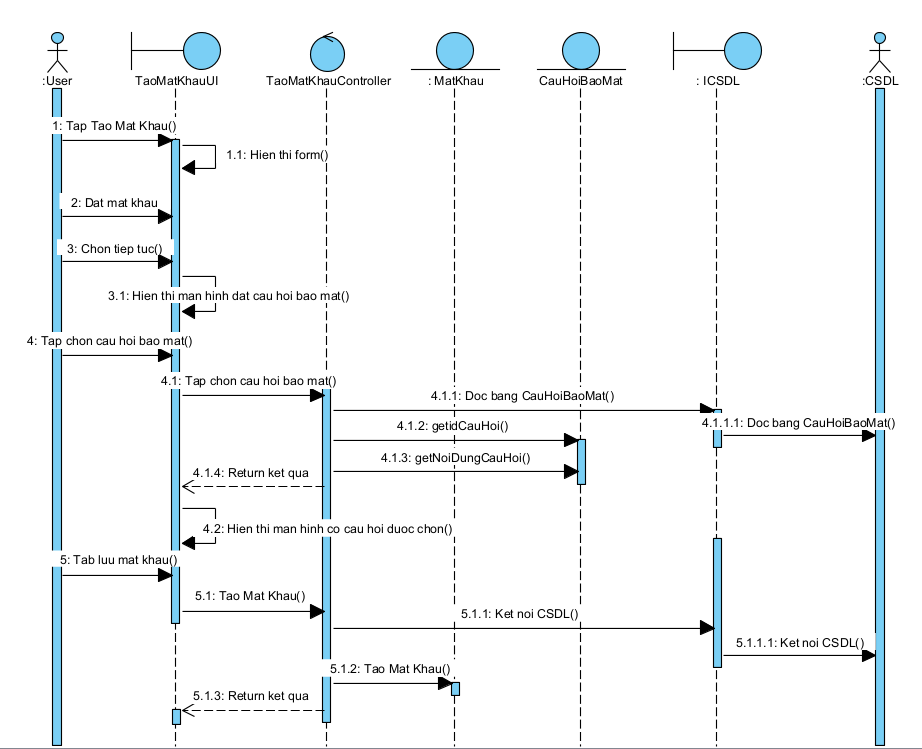
Hình 2. 4. Biểu đồ trình tự Use case Tạo profile

* + 1. Phân tích Use case Tạo mật khẩu
       1. **Biểu đồ lớp phân tích**



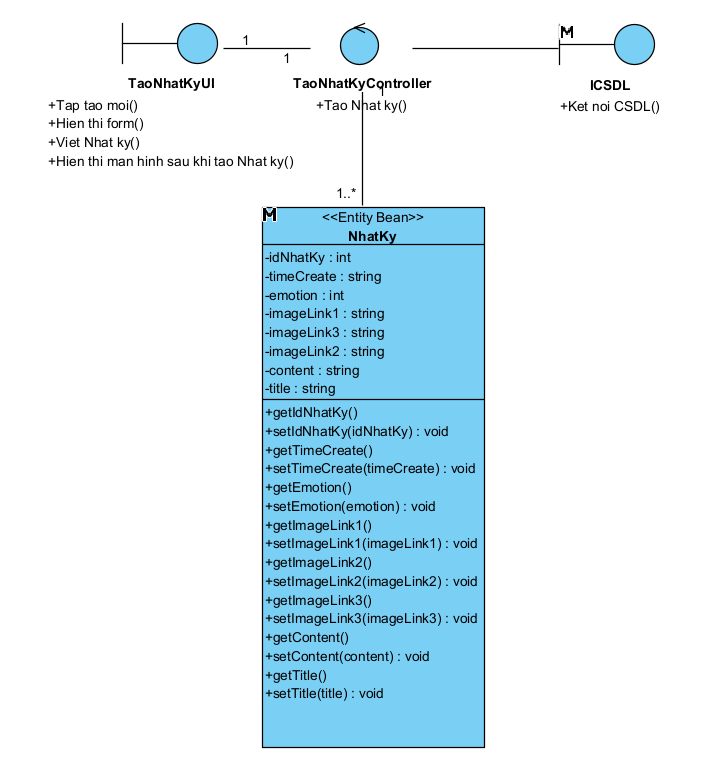
Hình 2. 5. Biểu đồ lớp phân tích Use case Tạo mật khẩu

* + - 1. **Biểu đồ trình tự**



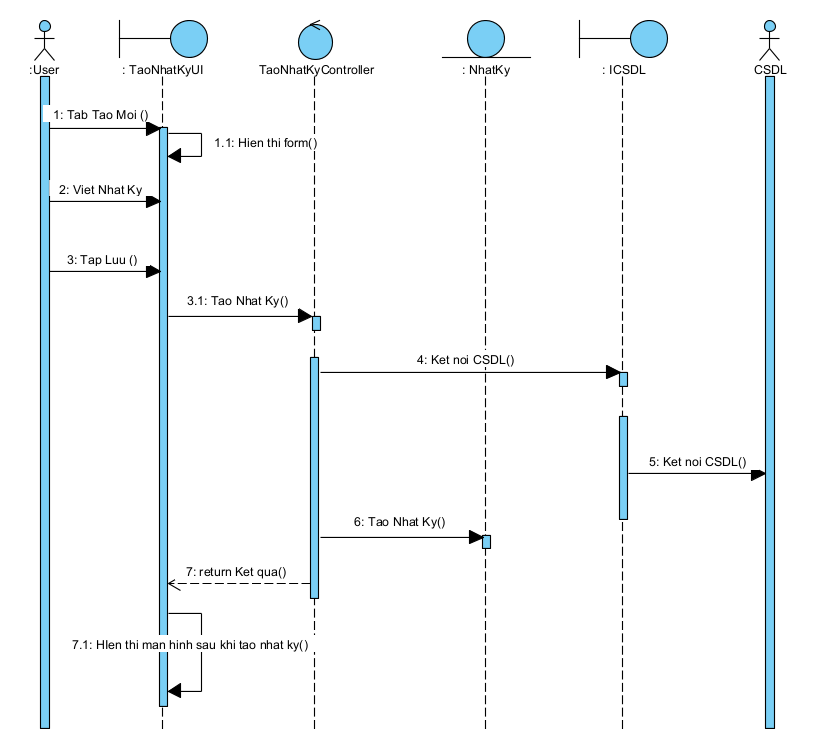
Hình 2. 6. Biểu đồ trình tự Use case Tạo mật khẩu

* + 1. Phân tích Use case Tạo nhật ký
       1. **Biểu đồ lớp phân tích**



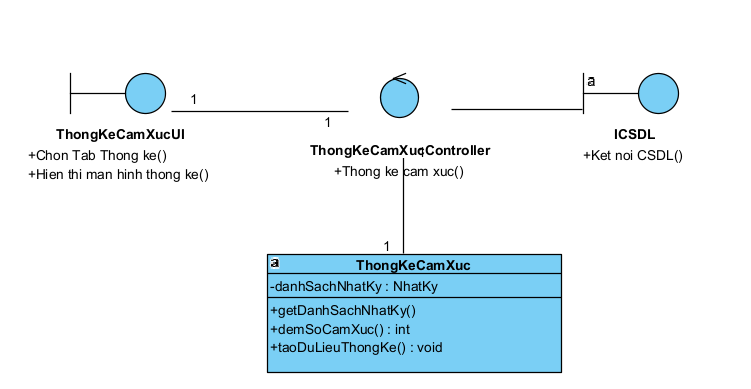
Hình 2. 7. Biểu đồ lớp phân tích Use case Tạo nhật ký

* + - 1. **Biểu đồ trình tự**



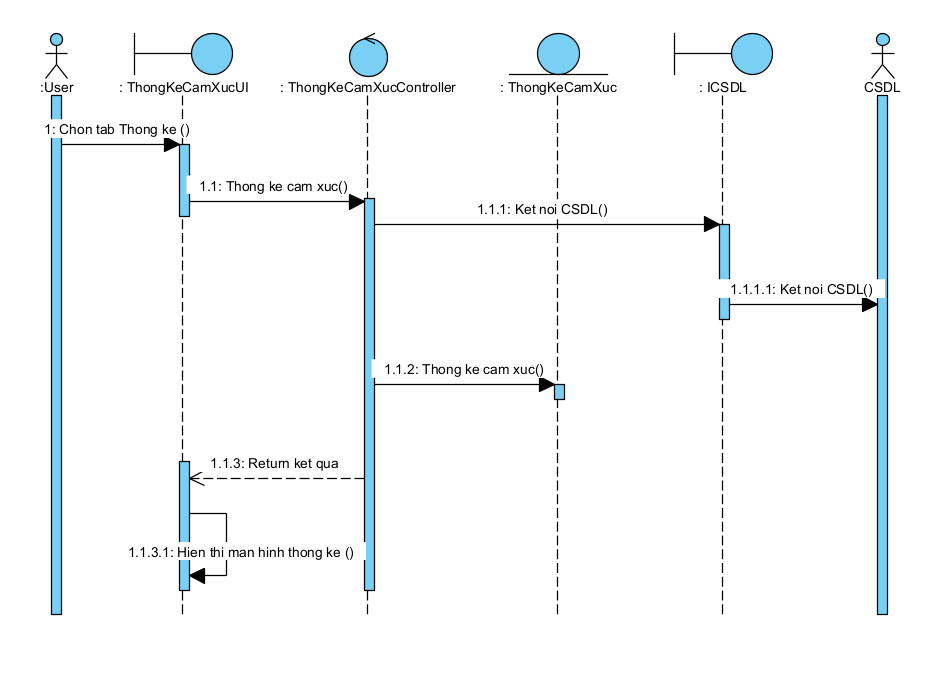
Hình 2. 8. Biểu đồ trình tự Use case Tạo nhật ký

* + 1. Phân tích Use case Thống kê cảm xúc
       1. **Biểu đồ lớp phân tích**

****

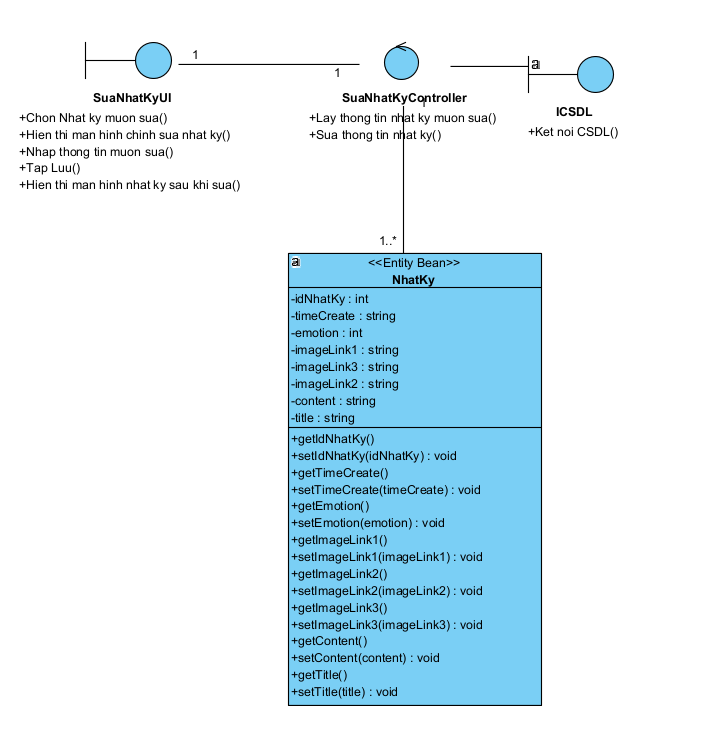
Hình 2. 9. Biểu đồ lớp Use case Thống kê cảm xúc

* + - 1. **Biểu đồ trình tự**

****

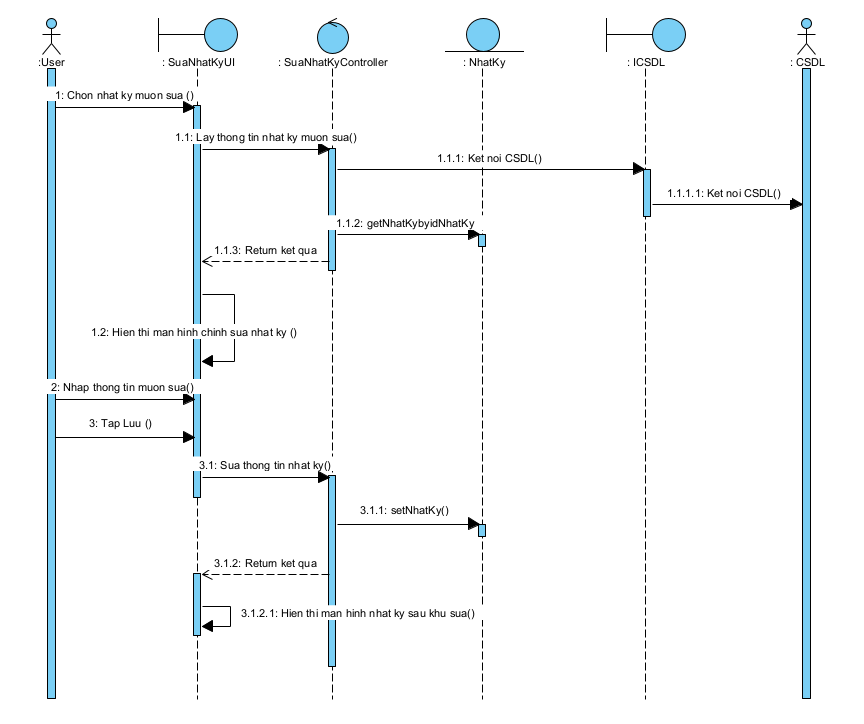
Hình 2. 10. Biểu đồ trình tự Use case Thống kê cảm xúc

* + 1. Phân tích Use case Sửa nhật ký
       1. **Biểu đồ lớp phân tích**



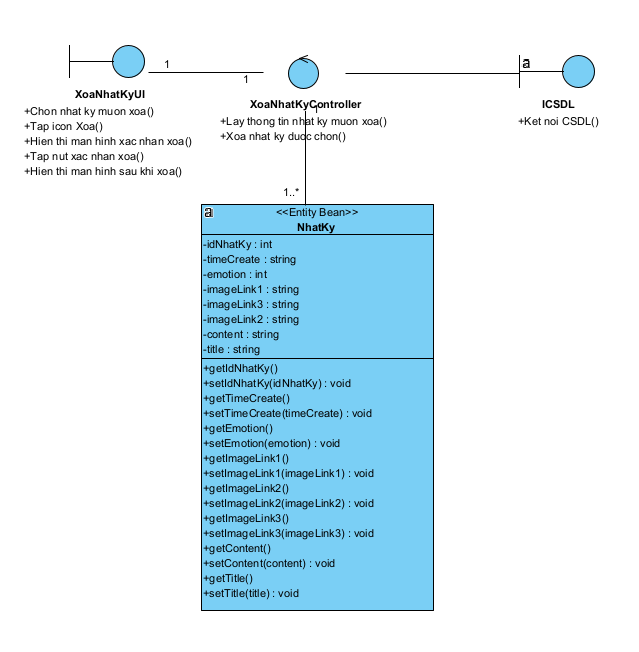
Hình 2. 11. Biểu đồ lớp phân tích Use case Sửa nhật ký

* + - 1. **Biểu đồ trình tự**



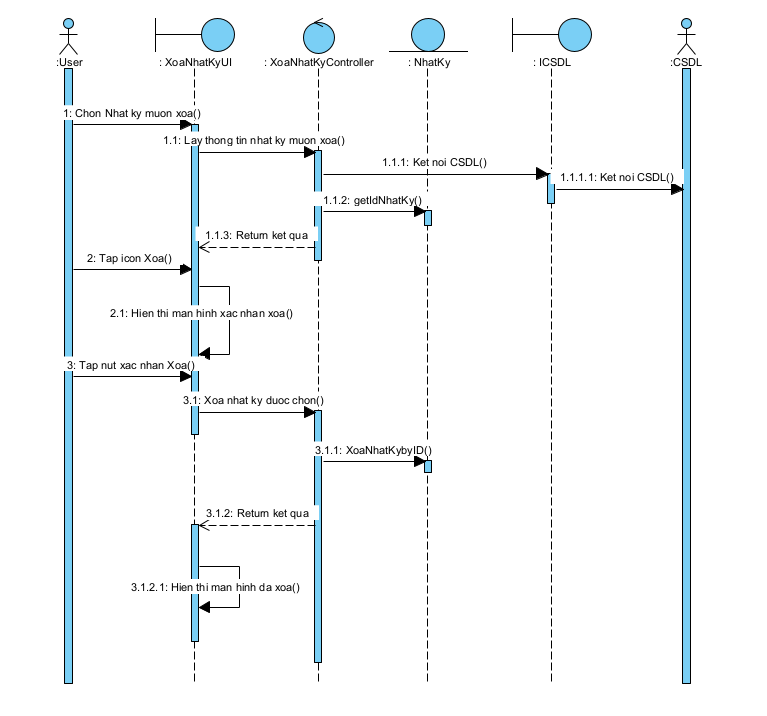
Hình 2. 12. Biểu đồ trình tự Use case Sửa nhật ký

* + 1. Phân tích Use case Xóa Nhật ký
       1. **Biểu đồ lớp phân tích**



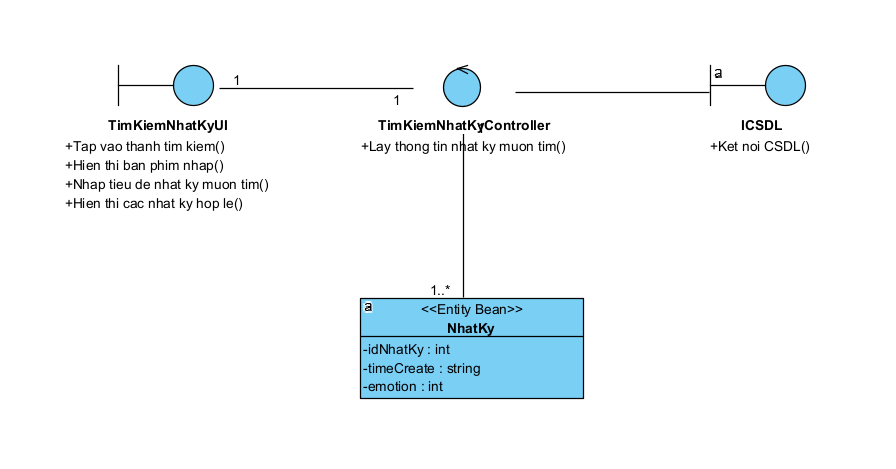
Hình 2. 13. Biểu đồ lớp phân tích Use case Xóa nhật ký

* + - 1. **Biểu đồ trình tự**



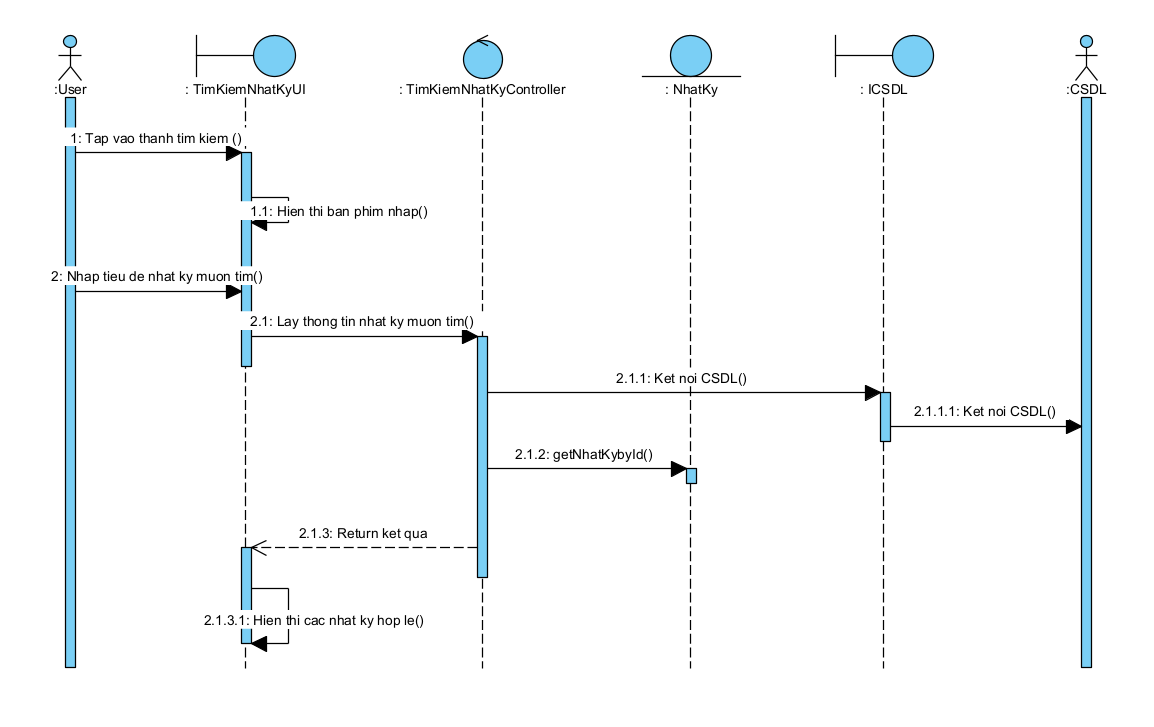
Hình 2. 14. Biểu đồ trình tự Use case Xóa nhật ký

* + 1. Phân tích Use case Tìm kiếm nhật ký
       1. **Biểu đồ lớp phân tích**

****

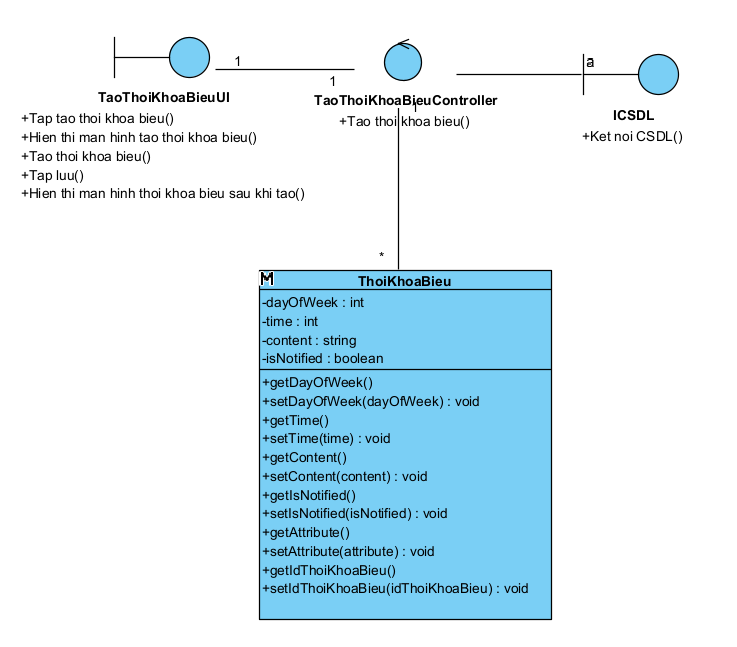
Hình 2. 15. Biểu đồ lớp phân tích Use case Tìm kiếm nhật ký

* + - 1. **Biểu đồ trình tự**



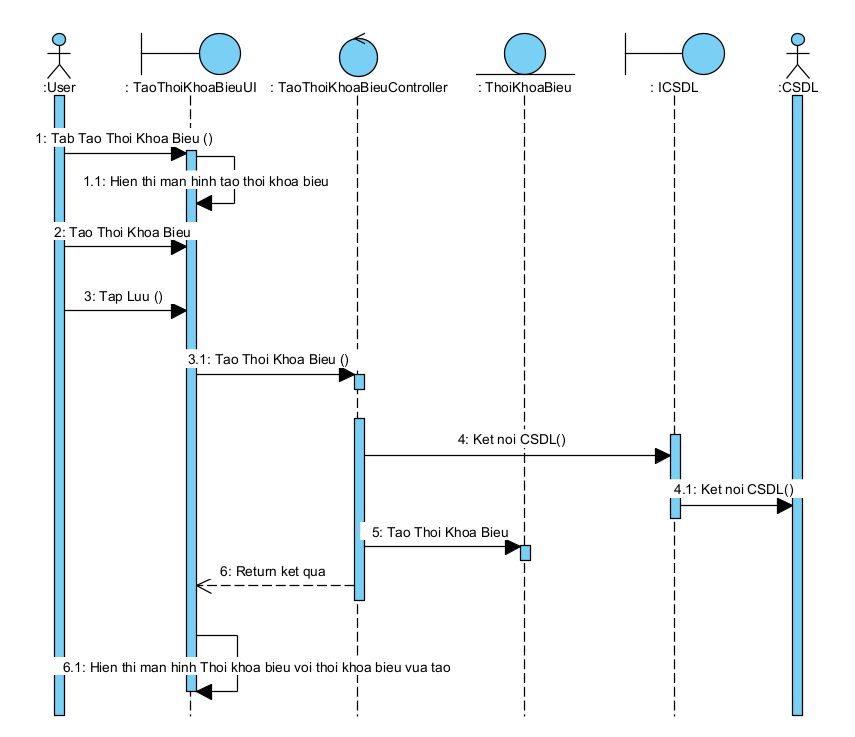
Hình 2. 16. Biểu đồ trình tự Use case Tìm kiếm nhật ký

* + 1. Phân tích Use case Tạo thời khóa biểu
       1. **Biểu đồ lớp phân tích**



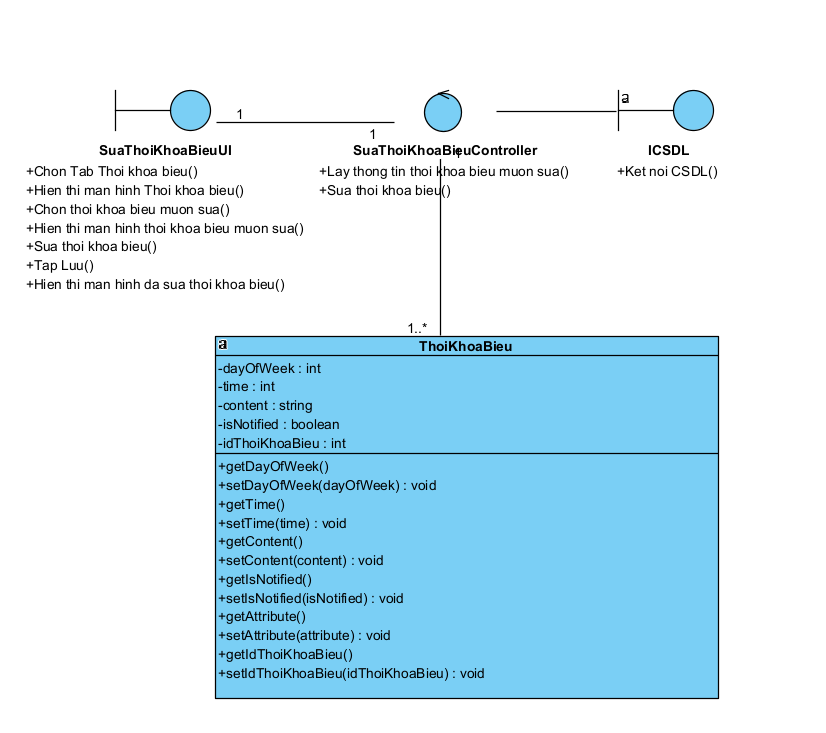
Hình 2. 17. Biểu đồ lớp phân tích Use case Tạo thời khóa biểu

* + - 1. **Biểu đồ trình tự**



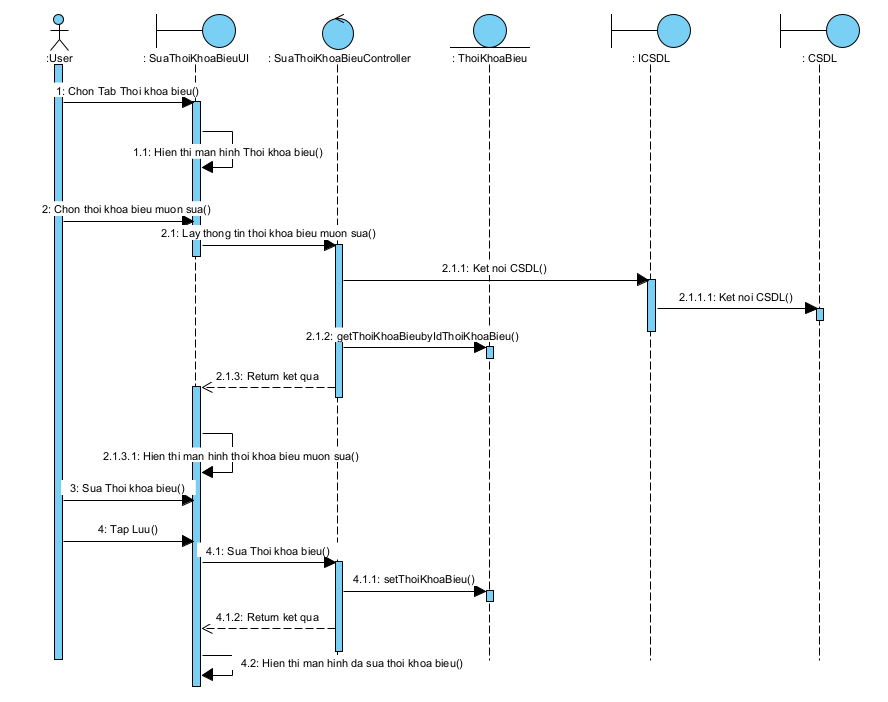
Hình 2. 18. Biểu đồ trình tự Use case Tạo thời khóa biểu

* + 1. Phân tích Use case Sửa thời khóa biểu
       1. **Biểu đồ lớp phân tích**

****

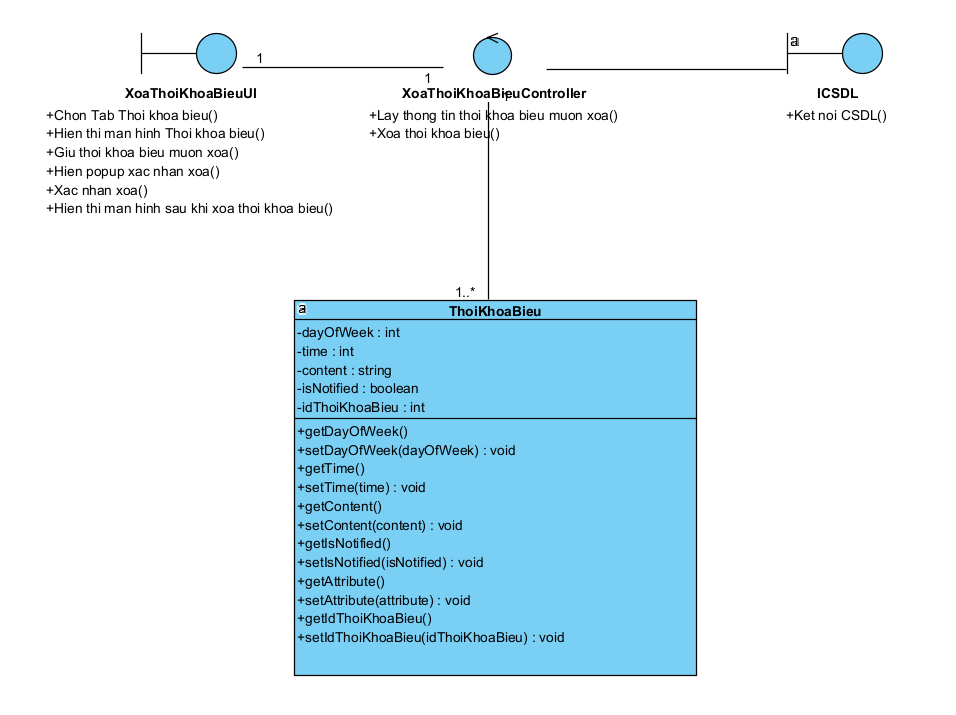
Hình 2. 19. Biểu đồ lớp Use case Sửa thời khóa biểu

* + - 1. **Biểu đồ trình tự**

****

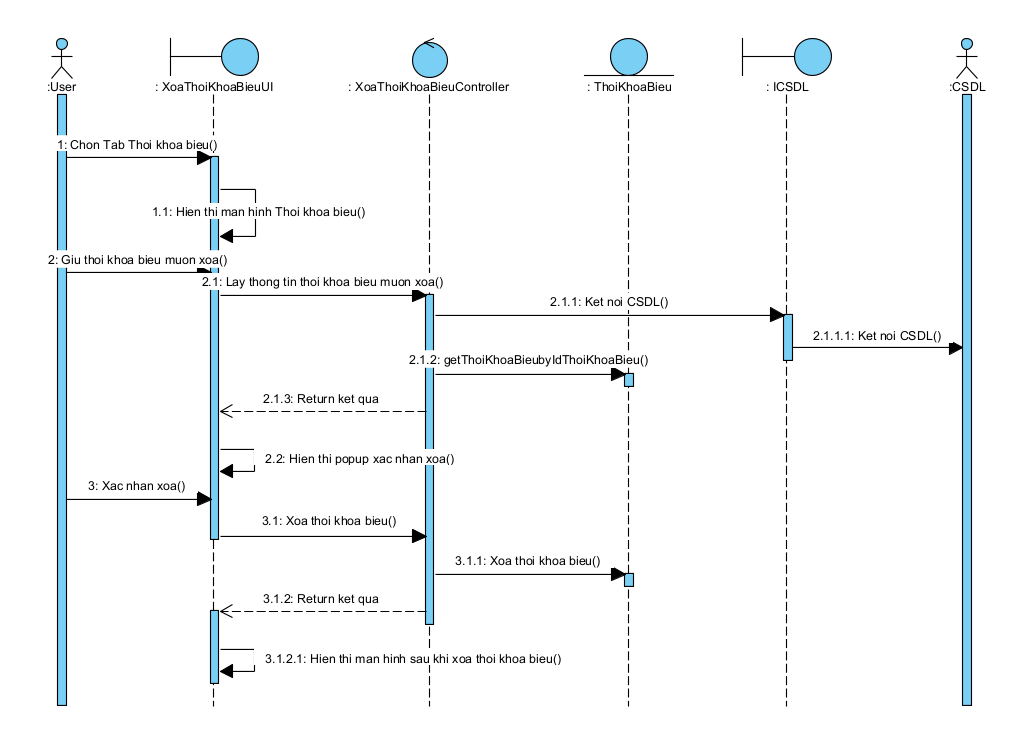
Hình 2. 20. Biểu đồ trình tự Use case Sửa thời khóa biểu

* + 1. Phân tích Use case Xóa thời khóa biểu
       1. **Biểu đồ lớp phân tích**

****

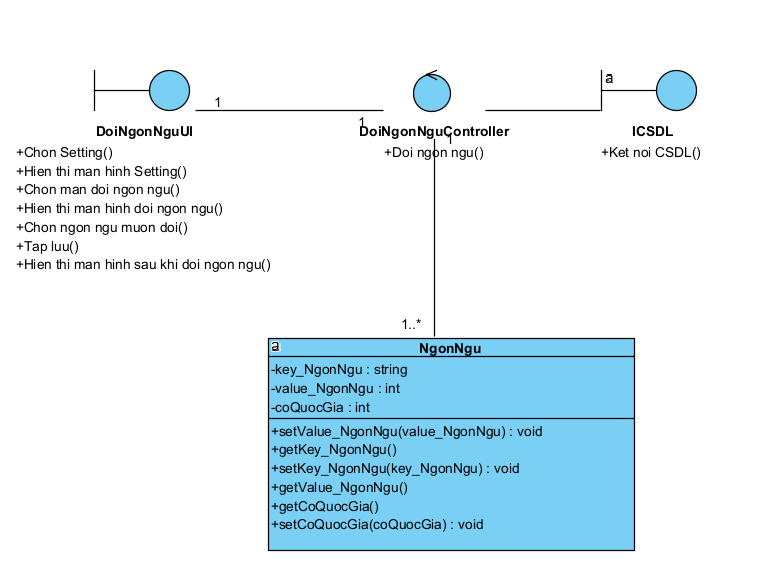
Hình 2. 21. Biểu đồ lớp Use case Xóa thời khóa biểu

* + - 1. **Biểu đồ trình tự**

****

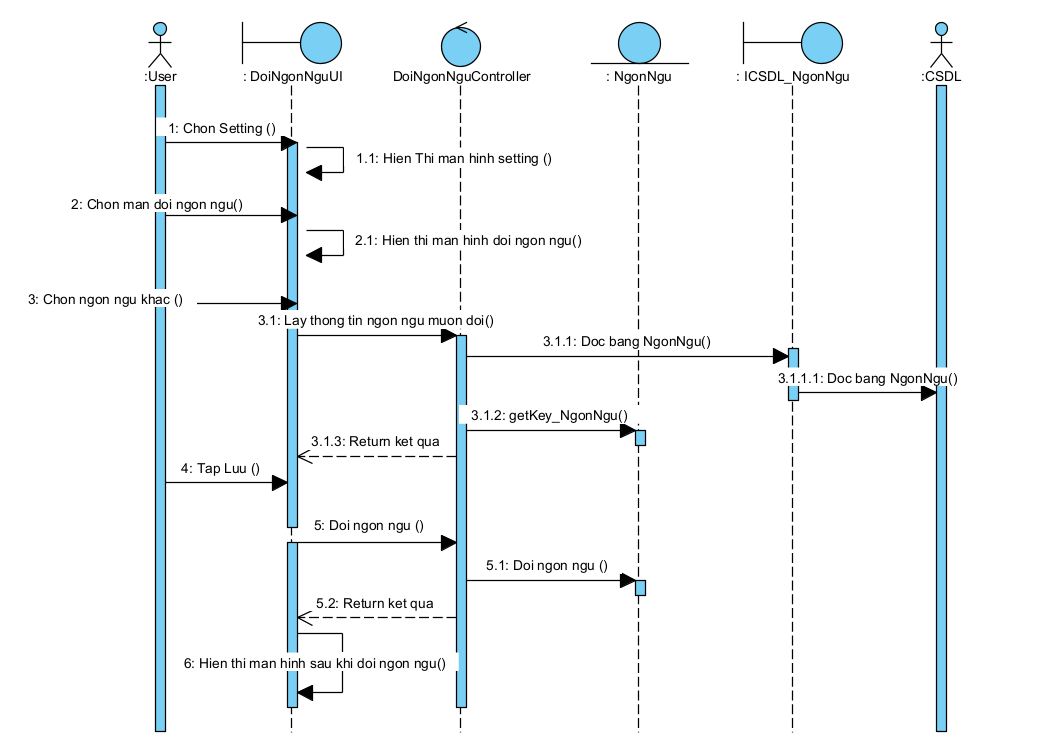
Hình 2. 22. Biểu đồ trình tự Use case Xóa thời khóa biểu

* + 1. Phân tích Use case Đổi ngôn ngữ
       1. **Biểu đồ lớp phân tích**



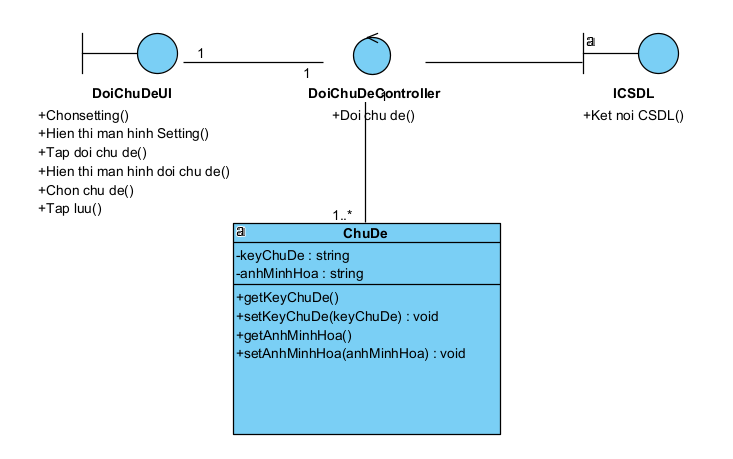
Hình 2. 23. Biểu đồ lớp phân tích Use case Đổi ngôn ngữ

* + - 1. **Biểu đồ trình tự**



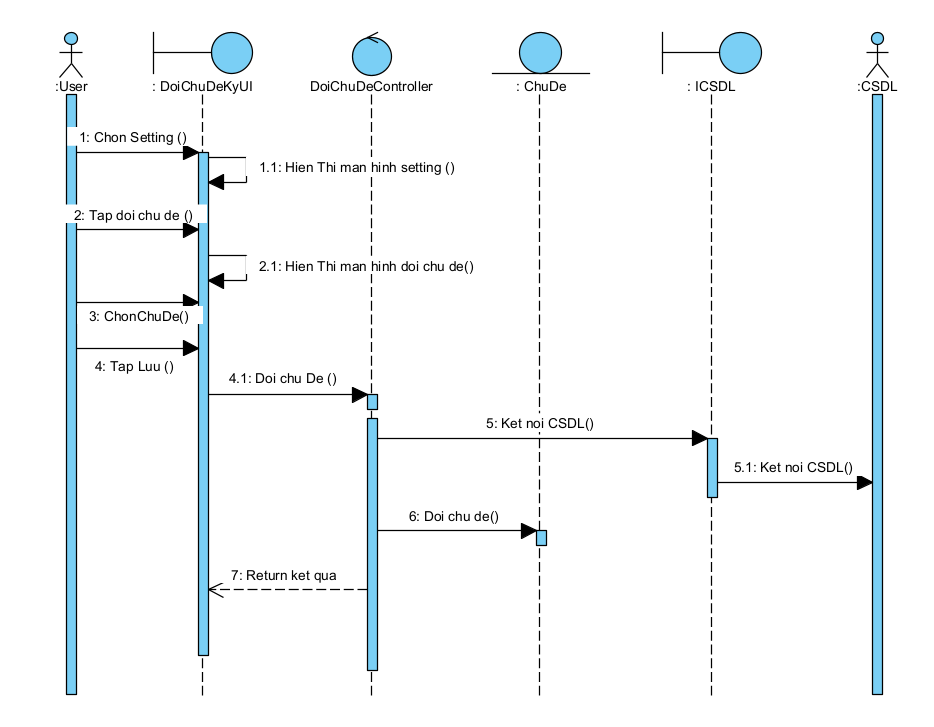
Hình 2. 24. Biểu đồ trình tự Use case Đổi ngôn ngữ

* + 1. Phân tích Use case Đổi chủ đề
       1. **Biểu đồ lớp phân tích**



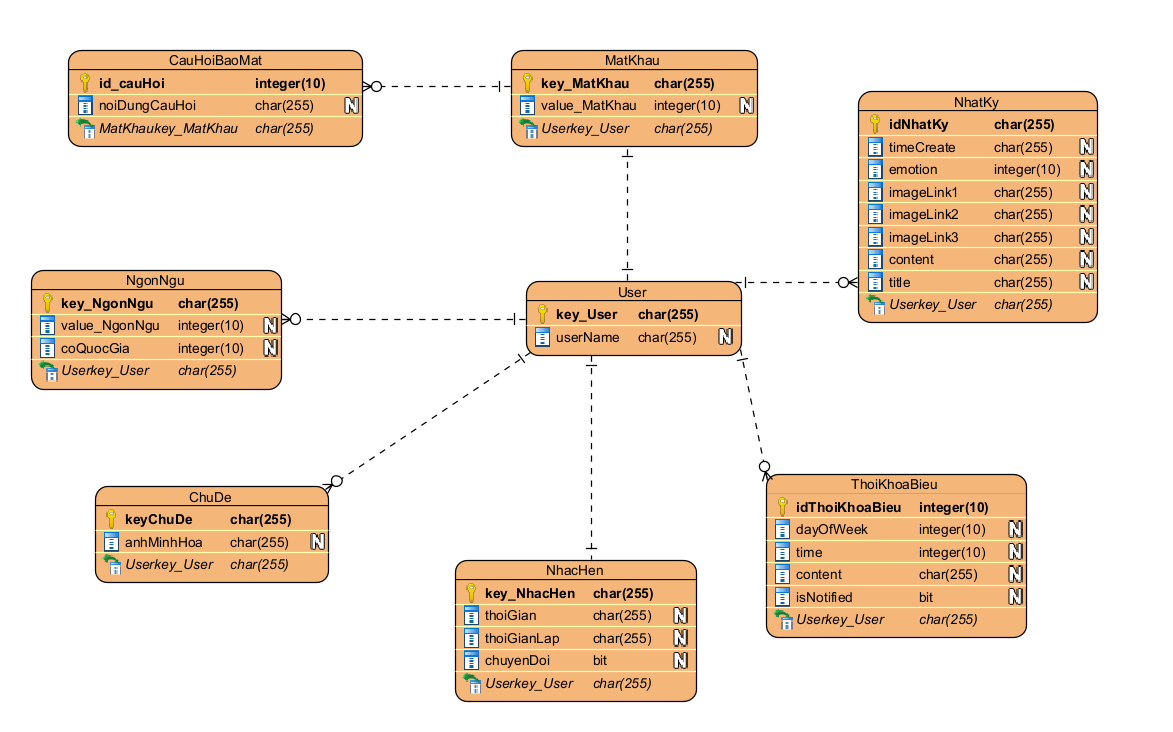
Hình 2. 25. Biểu đồ trình tự Use case Đổi chủ đề

* + - 1. **Biểu đồ trình tự**



Hình 2. 26. Biểu đồ trình tự Use case Đổi chủ đề

* 1. Thiết kế cơ sở dữ liệu
     1. Mô hình thực thể liên kết



Hình 2. 27. Mô hình thực thể liên kết

* + 1. Thiết kế bảng cơ sở dữ liệu

NhatKy để lưu thông tin nhật ký

Bảng 2. 2 Mô tả cơ sở dữ liệu bảng NhatKy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Bắt buộc | Khóa | Giải thích |
| id | Int | NN | PK | Id nhật ký |
| timeCreate | Stirng | NN |  | Thời gian tạo nhật ký |
| emotion | Int | NN |  | Cảm xúc |
| title | String |  |  | Tiêu đề nhật ký |
| image\_Link1 | String |  |  | Ảnh thêm vào nhật ký |
| image\_Link2 | String |  |  | Ảnh thêm vào nhật ký |
| image\_Link3 | String |  |  | Ảnh thêm vào nhật ký |
| content | String |  |  | Nội dung nhật ký |

ThoiKhoaBieu để lưu thông tin thời khóa biểu

Bảng 2. 3 Mô tả cơ sở dữ liệu bảng ThoiKhoaBieu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Bắt buộc | Khóa | Giải thích |
| idThoiKhoaBieu | Int | NN | PK | Id thời khóa biểu |
| dayOfWeek | Int | NN |  | Thứ trong tuần |
| Time | Int |  |  | Thời gian (giờ, phút) |
| Title | String |  |  | Tiêu đề |
| Content | String |  |  | Nội dung |
| isNotified | Boolean |  |  | Nhắc nhở |

NhacHen để lưu thông tin thời gian nhắc hẹn viết nhật ký

Bảng 2. 4 Mô tả cơ sở dữ liệu bảng NhacHen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Bắt buộc | Khóa | Giải thích |
| key\_NhacHen | String | NN | PK | Khóa chính của bảng nhắc hẹn |
| thoiGian | String |  |  | Thời gian nhắc hẹn |
| thoiGianLap | String |  |  | Thời gian cài đặt lặp lại nhắc hẹn |
| chuyenDoi | Boolean |  |  | Bật tắt chế độ nhắc hẹn |

ChuDe để lưu thông tin các chủ đề hiển thị trong ứng dụng

Bảng 2. 5 Mô tả cơ sở dữ liệu bảng ChuDe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Bắt buộc | Khóa | Giải thích |
| key\_ChuDe | String | NN | PK | Khóa chính của bảng ChuDe |
| anhMinhHoa | String |  |  | Hình ảnh minh họa của chủ đề để hiện thị trên ứng dụng |

NgonNgu để lưu thông tin ngôn ngữ các quốc gia người dùng muốn hiển thị trong ứng dụng

Bảng 2. 6 Mô tả cơ sở dữ liệu bảng NgonNgu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Bắt buộc | Khóa | Giải thích |
| key\_NgonNgu | String | NN | PK | Khóa chính của bảng NgonNgu |
| value\_NgonNgu | Int |  |  | Giá trị của ngôn ngữ trong list các ngôn ngữ có sẵn |
| coQuocGia | Int |  |  | Ảnh cờ quốc gia của các ngôn ngữ hiển thị trong ứng dụng |

MatKhau để lưu thông tin khóa bảo mật mà người dùng tạo

Bảng 2. 7 Mô tả cơ sở dữ liệu bảng MatKhau

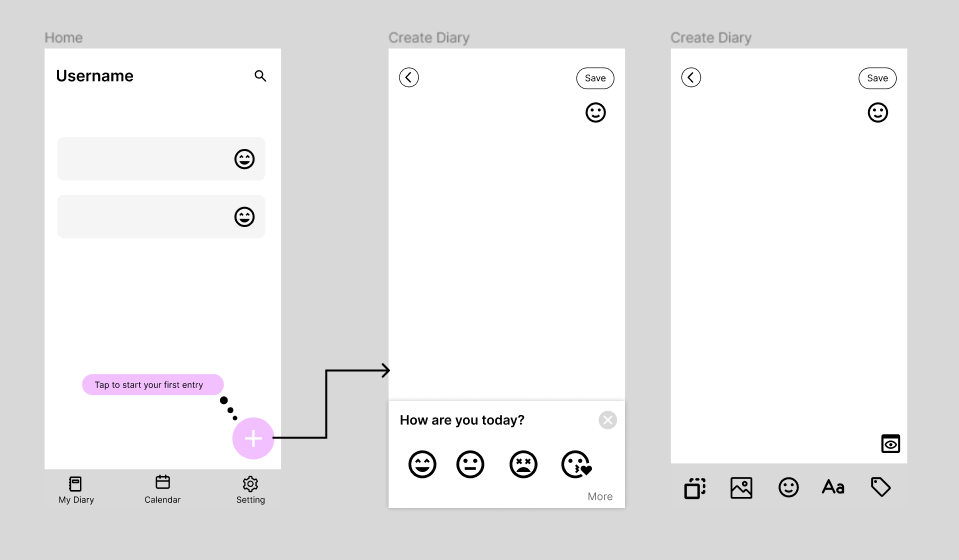
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Bắt buộc | Khóa | Giải thích |
| key\_MatKhau | String | NN | PK | Khóa chính của bảng mật khẩu |
| Value\_MatKhau | Int |  |  | Giá trị của mật |
| id\_CauHoi | Int | NN | FK | Khóa chính của bảng CauHoiBaoMat |

CauHoiBaoMat để lưu thông tin các câu hỏi trong phần đặt câu hỏi bảo mật

Bảng 2. 8 Mô tả cơ sở dữ liệu bảng CauHoiBaoMat

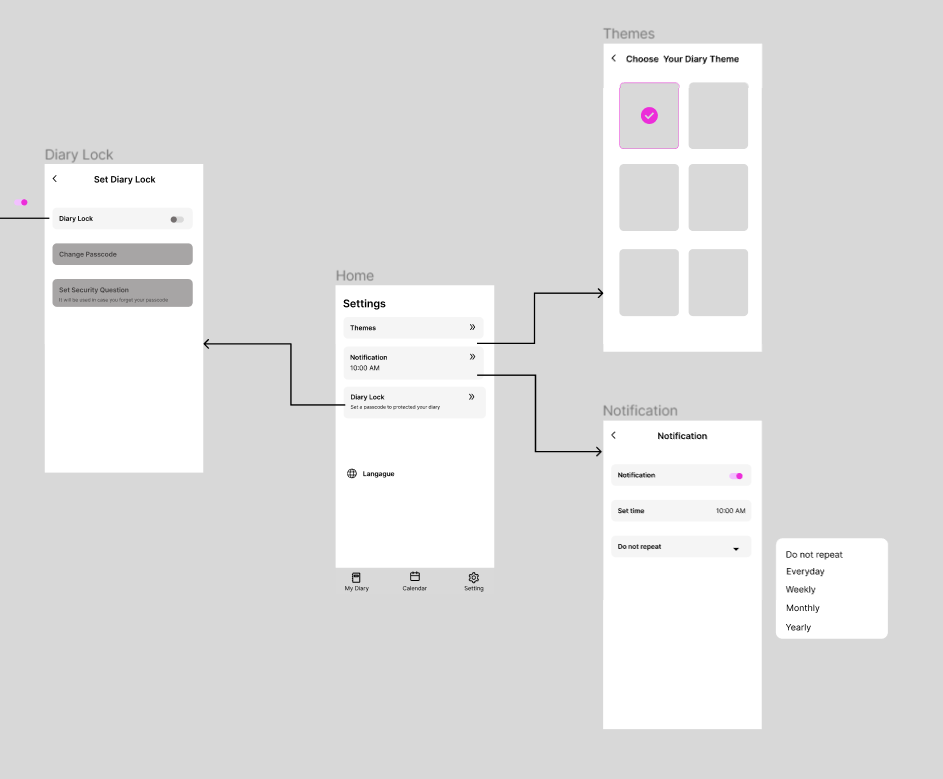
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Bắt buộc | Khóa | Giải thích |
| id\_CauHoi | Int | NN | PK | Id câu hỏi bảo mật |
| noiDungCauHoi | String |  |  | Nội dung của câu hỏi bảo mật |

* 1. Thiết kế giao diện
     1. Thiết kế giao diện giả lập luồng tạo nhật ký



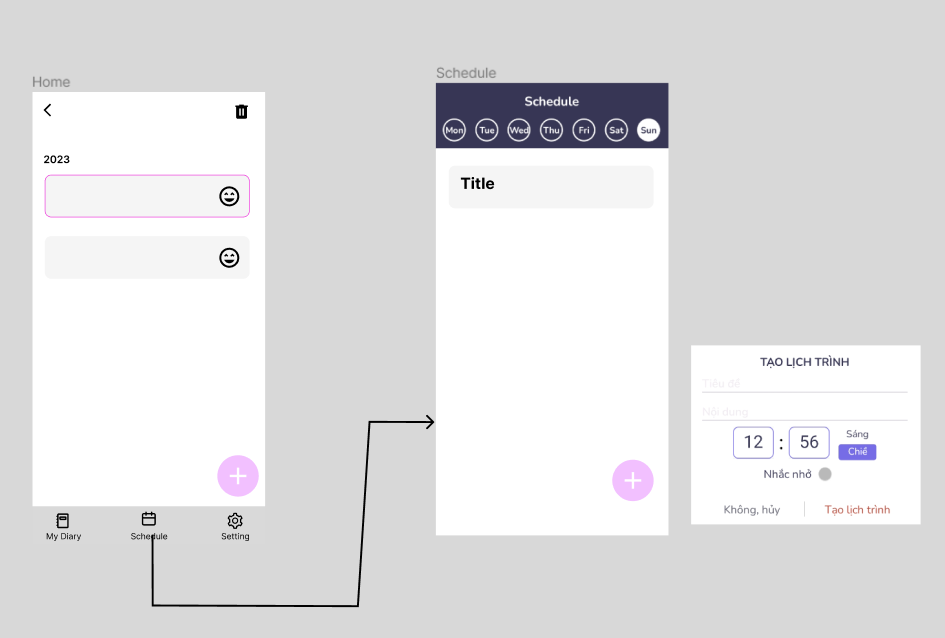
Hình 2. 28. Giao diện luồng tạo nhật ký

* + 1. Thiết kế giao diện giả lập luồng thao tác tại màn Cài đặt



Hình 2. 29. Giao diện luồng thao tác tại màn Cài đặt

* + 1. Thiết kế giao diện giả lập luồng tạo thời khóa biểu



Hình 2. 30. Giao diện luồng tạo thời khóa biểu

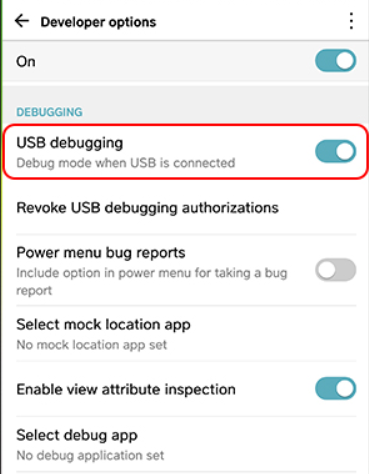
**Chương 3 KẾT QUẢ ĐẠT ĐƯỢC**

* 1. Cài đặt
* Yêu cầu thiết bị: Điện thoại có hệ điều hành Android
* Yêu cầu phần mềm: Android Studio

**Cài đặt ứng dụng bằng cách chạy trên thiết bị thực qua cáp USB:**

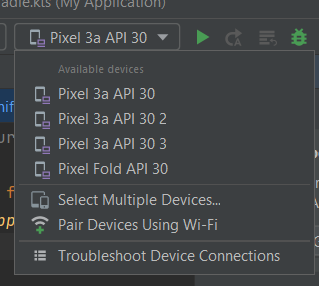
Bước 1: Kết nối thiết bị Android với máy tính qua cáp USB

Bước 2: Bật chế độ “Developer Options” và “USB Debugging” trên thiết bị Android



Hình 3. 1 Bật chế độ gỡ lỗi qua USB trên thiết bị Android

Bước 3: Chọn thiết bị thực từ danh sách thiết bị trong thanh công cụ của Android Studio



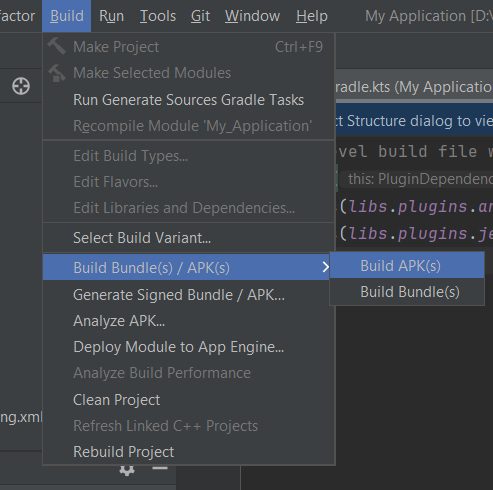
Hình 3. 2 Chọn thiết bị Android để chạy

Bước 4: Nhấn nút “Run” để chạy ứng dụng trên thiết bị thực

**Cài đặt ứng dụng bằng cách tải file APK từ Android Studio:**

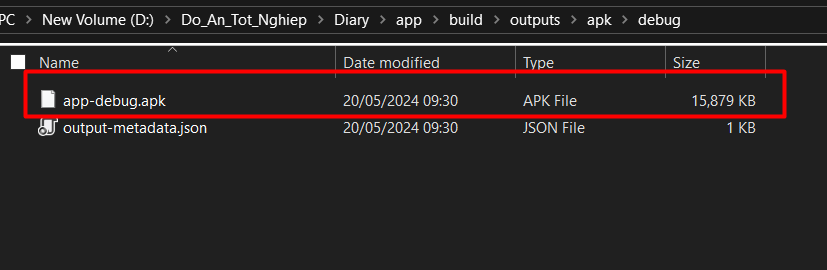
Bước 1: Chọn “Build” trên thanh menu của Andorid Studio

Bước 2: Chọn “Build Bundle(s)/APK(s)” rồi chọn “Build APK(s)”. Vậy là đã tạo xong file APK debug.



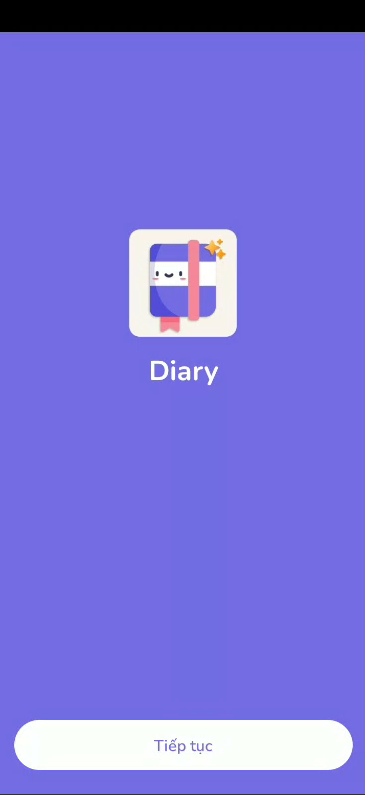
Hình 3. 3 Tạo file APK

Bước 2: File APK đã được lưu ở trong máy, lấy file này cài đặt trên thiết bị Android



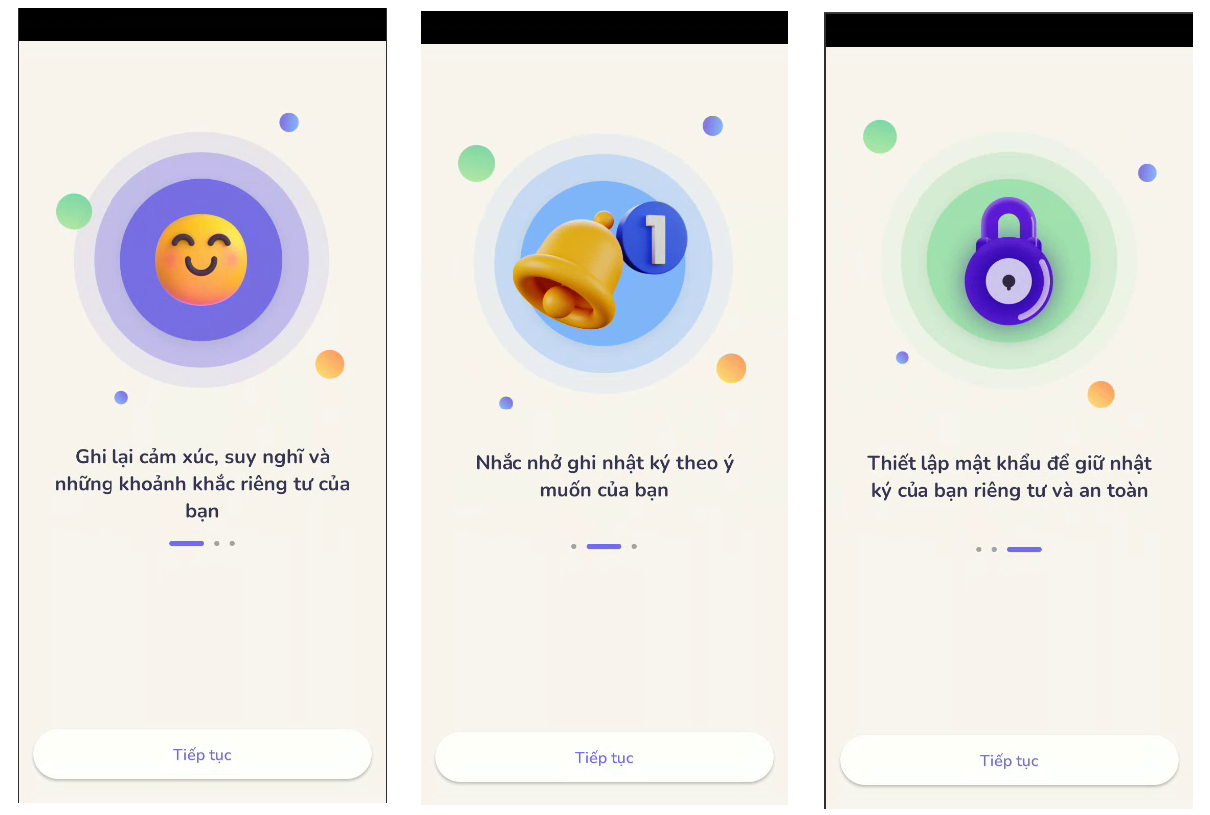
Hình 3. 4 File APK được lưu trong thư mục máy

* 1. Kết quả đạt được
     1. Giao diện các màn hình trên ứng dụng
        1. Giao diện màn Splash



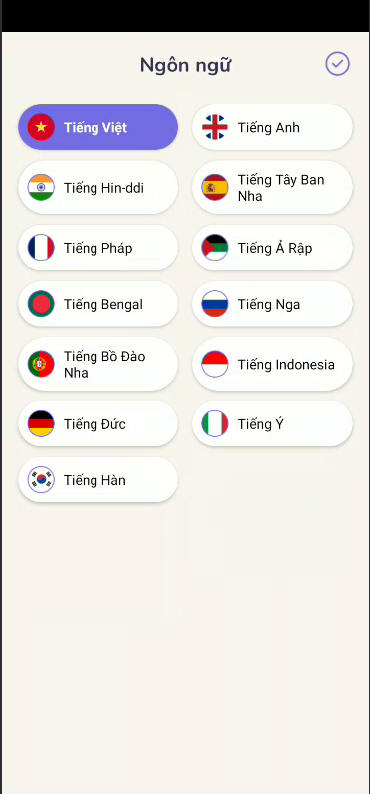
Hình 3. 5 Giao diện màn Splash

* + - 1. Giao diện màn Intro



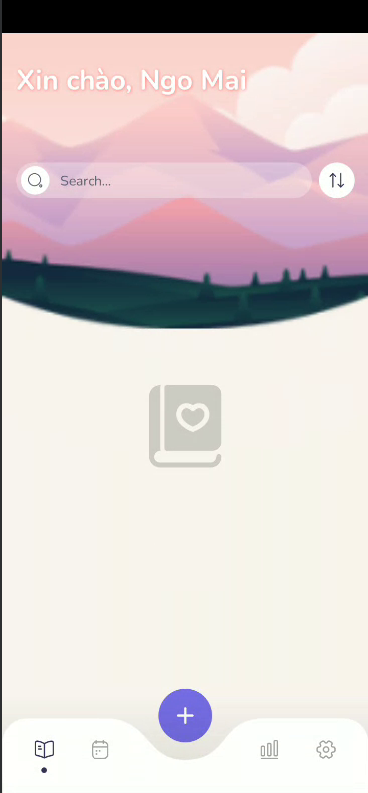
Hình 3. 6 Giao diện màn Intro

* + - 1. Giao diện màn Ngôn ngữ



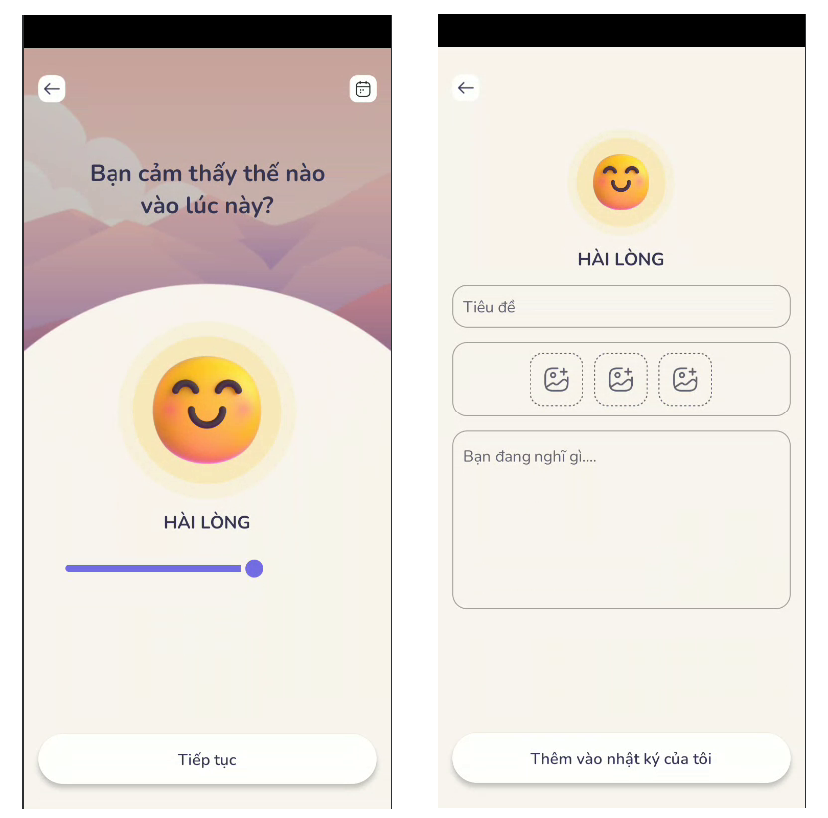
Hình 3. 7 Giao diện màn Ngôn ngữ

* + - 1. Giao diện màn hình chính



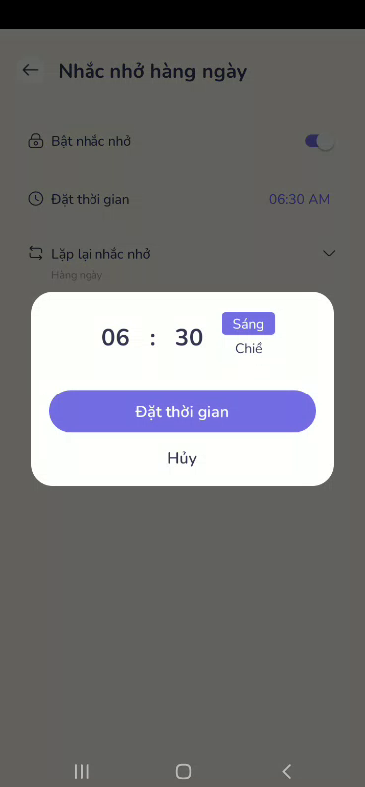
Hình 3. 8 Giao diện màn hình chính

* + - 1. Giao diện màn Tạo nhật ký



Hình 3. 9 Giao diện màn hình Tạo nhật ký

* + - 1. Giao diện màn Tạo nhắc hẹn



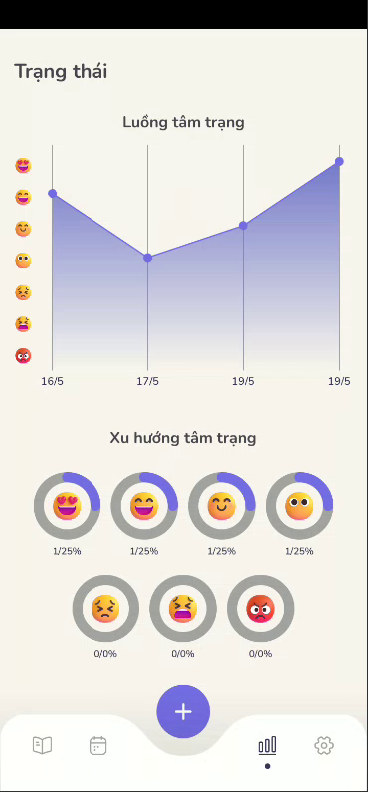
Hình 3. 10 Giao diện màn hình Tạo nhắc hẹn

* + - 1. Giao diện màn Thời khóa biểu



Hình 3. 11 Giao diện màn hình Tạo thời khóa biểu

* + - 1. Giao diện màn Thống kê cảm xúc



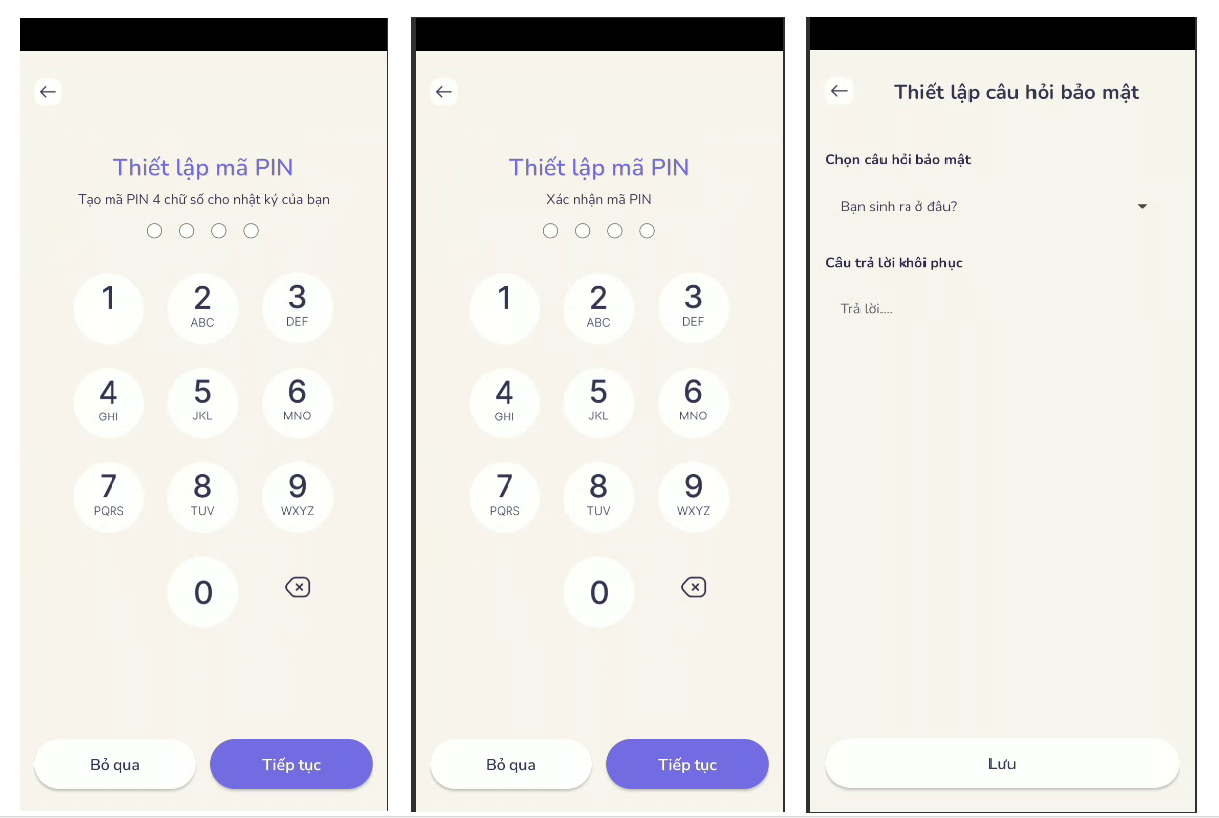
Hình 3. 12 Giao diện màn hình Thống kê cảm xúc

* + - 1. Giao diện màn Tạo Profile



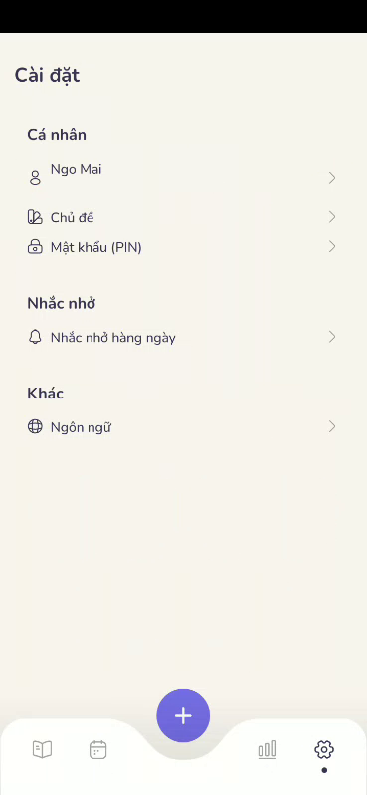
Hình 3. 13 Giao diện màn Tạo Profile

* + - 1. Giao diện màn Tạo mật khẩu



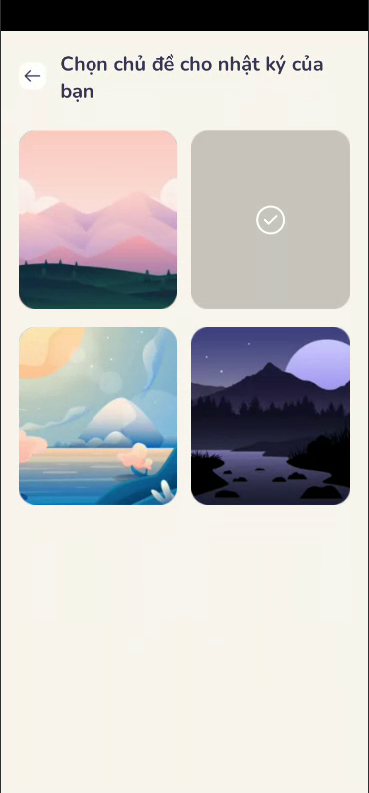
Hình 3. 14 Giao diện màn hình Tạo mật khẩu bảo mật

* + - 1. Giao diện màn Cài đặt



Hình 3. 15 Giao diện màn hình Cài đặt

* + - 1. Giao diện màn Chọn chủ đề



Hình 3. 16 Giao diện màn hình chọn chủ đề

* + 1. Kết quả kiểm thử

Bảng 3. 1. Bảng kết quả kiểm thử ứng dụng Viết nhật ký hàng ngày cho sinh viên

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| 1 | Tạo Profile | Kiểm tra chức năng “Tạo Profile” | Nhập tên người dùng | Vào trong hệ thống hiển thị đúng tên người dùng đã nhập | Pass |
| 2 | Tạo mật khẩu | Kiểm tra chức năng “Tạo mật khẩu” | Nhập dữ liệu | Tạo mật khẩu thành công | Pass |
| Không nhập dữ liệu | Thông báo yêu cầu nhập dữ liệu tạo mật khẩu | Pass |
| 3 | XeTạo Nhật ký | Kiểm tra chức năng “Tạo Nhật ký” | Nhập dữ liệu | Tạo Nhật ký thành công | Pass |
| Không nhập dữ liệu | Thông báo yêu cầu nhập dữ liệu để tạo Nhật ký | Pass |
| 4 | Sửa Nhật ký | Kiểm tra chức năng “Sửa Nhật ký” | Nhập dữ liệu | Sửa Nhật ký thành công | Pass |
| 5 | Xóa Nhật ký | Kiểm tra chức năng “Xóa Nhật ký” | Đồng ý xóa | Xóa thành công Nhật ký khỏi màn hình chính | Pass |
| 6 | Thống kê Cảm xúc | Kiểm tra chức năng “Thống kê cảm xúc” | Chưa tạo Nhật ký | Hiện thị tất cả các cảm xúc là 0% | Pass |
| Đã tạo Nhật ký | Hiển thị đúng phần trăm các cảm xúc đã thêm | Pass |
| 7 | XemTạo Thời khóa biểu | Kiểm tra chức năng “Tạo Thời khóa biểu” | Nhập dữ liệu | Tạo Thời khóa biểu thành công | Pass |
| Không nhập dữ liệu | Không tạo được Thời khóa biểu | Pass |
| 8 | Sửa Thời khóa biểu | Kiểm tra chức năng “Sửa Thời khóa biểu” | Nhập dữ liệu | Sửa thời khóa biểu thành công | Pass |
| Không nhập dữ liệu | Thời khóa biểu được chọn dữ nguyên | Pass |
| 9 | Xóa Thời khóa biểu | Kiểm tra chức năng “Xóa Thời khóa biểu” | Đồng ý xóa | Xóa thành công Thời khóa biểu khỏi màn hình | Pass |
| 10 | Tạo nhắc hẹn | Kiểm tra chức năng “Tạo nhắc hẹn” | Tạo nhắc nhở thành công | Thông báo lên màn hình đúng thông tin, đúng giờ | Pass |
| 11 | Đổi chủ đề | Kiểm tra chức năng “Đổi chủ đề” | Chọn chủ đề muốn đổi | Hiển thị đúng chủ đề đã chọn | Pass |
| 12 | Đổi ngôn ngữ | Kiểm tra chức năng “Đổi ngôn ngữ” | Chọn quốc gia muốn đổi | Ứng dụng đổi ngôn ngữ theo đúng quốc gia đã chọn | Pass |

* 1. Đánh giá kết quả

***Phần hoàn thành***

* Về mặt công nghệ, em đã tìm hiểu và nắm bắt quy trình để xây dựng một ứng dụng trên Android, tìm hiểu cách sử dụng trình soạn thảo Android Studio.
* Hoàn thiện bài khảo sát và phân tích thiết kế hệ thống “Ứng dụng viết nhật ký đời sống hàng ngày cho sinh viên”
* Lập trình thành thạo với ngôn ngữ Kotlin.

***Phần chưa hoàn thành***

* Cần hoàn thiện thêm về giao diện và tối ưu các chức năng giúp người dùng dễ sử dụng.
* Chưa thống kê cảm xúc được theo một khoảng thời gian cụ thể
  1. Hướng phát triển
* Thiết kế và lập trình mở rộng các chức năng.
* Phát triển thành một ứng dụng online, trở thành nền tảng giúp người dùng chia sẻ được nhật ký cho mọi người.
* Hỗ trợ thêm thay đổi giao diện, màu sắc ứng dụng.
* Cải thiện sự mượt mà, tối ưu giao diện của ứng dụng. Nâng cao trải nghiệm người dùng.

# **KẾT LUẬN**

Trong thời đại công nghệ hóa hiện nay, sinh viên thường phải đối mặt với nhiều thách thức trong việc quản lý thời gian và công việc học tập. Việc ghi lại những áp lực, cảm xúc hàng ngày rất có ích trong sự phát triển cảm xúc cá nhân, giúp chúng ta thư giãn hơn. Chính vì vậy, qua quá trình tìm hiểu và khảo sát, em đã chọn đề tài “Xây dựng ứng dụng Viết nhật ký đời sống hàng ngày cho sinh viên”.

Với kiến thức nền tảng cùng với sự hướng dẫn, giúp đỡ tận tình của thầy giáo hướng dẫn **Ts. Nguyễn Văn Tỉnh**, em đã hoàn thành đề tài đồ án tốt nghiệp và học hỏi thêm được nhiều kiến thức. Mặc dù ứng dụng đã hoàn thành nhưng nó vẫn còn nhiều mặt hạn chế và vấn đề thiếu sót như là các chức năng chưa đầy đủ, thiết kế giao diện sơ sài, chưa được chuẩn hóa và hoàn thiện, … .

Trong thời gian tới, em rất mong tiếp tục nhận được sự giúp đỡ và tạo điều kiện của thầy cô và nhà trường để em có cơ hội phát triển, hoàn thiện ứng dụng tốt hơn.

Hà Nội, ngày 20 tháng 05 năm 2024

Sinh viên thực hiện

Ngô Thị Mai

# **TÀI LIỆU THAM KHẢO**

[1] Đỗ Ngọc Sơn, Phan Văn Viên, Nguyễn Phương Nga (2015), *Giáo trình hệ quản trị cơ sở dữ liệu SQL,* NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Kim Phượng (2011), *Giáo trình phân tích thiết kế hệ thống,* NXB Giáo dục Việt Nam.

[3] Ths. Lê Văn Hùng (2017), *Cơ sở dữ liệu quan hệ và công nghệ phân tích – thiết kế,* Nhà xuất bản Thông tin và truyền thông.

[4] Barry Burd, PhD – Java Programming for Android Developers.

[5] Tìm hiểu về lập trình ứng dụng di động:

<https://tuhoclaptrinh.edu.vn/lap-trinh-ung-dung-di-dong-va-nhung-dieu-can-biet-687.html>, truy cập cuối cùng ngày 12/05/2024.

[6] Tìm hiểu về hệ điều hành Android:

<https://didongviet.vn/dchannel/he-dieu-hanh-android/#ftoc-5-uu-diem-cua-he-dieu-hanh-android>, truy cập cuối cùng ngày 11/05/2024.

[7] Tài liệu về ông cụ Android Studio:

<https://didongviet.vn/dchannel/android-studio/>, truy cập cuối cùng ngày 12/05/2024.

[8] Tài liệu về ngôn ngữ Kotlin:

<https://vi.wikipedia.org/wiki/Kotlin_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)>, truy cập cuối cùng ngày 12/05/2024.

<https://tuhoclaptrinh.edu.vn/gioi-thieu-ve-ngon-ngu-lap-trinh-kotlin-726.html>, truy cập cuối cùng ngày 12/05/2024.

[9] Tài liệu về mô hình MVVM:

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>, truy cập cuối cùng ngày 12/05/2024.

<https://viblo.asia/p/tim-hieu-ve-mo-hinh-mvvm-maGK7vW95j2>, truy cập cuối cùng ngày 12/05/2024.

[10] Tài liệu về java và các thư viện java hỗ trợ: <https://docs.oracle.com/en/java/>, truy cập cuối cùng ngày 12/05/2024.