# CHƯƠNG 1: TỔNG QUAN VỀ PHẦN MỀM KIDSAFE

## 1.1. Tổng quan về phần mềm kiểm soát của phụ huynh

Trong bối cảnh xã hội số hóa, Internet đã trở thành một phần không thể thiếu trong cuộc sống, đặc biệt là đối với thế hệ trẻ. Nó mở ra một thế giới tri thức khổng lồ, các phương tiện giải trí đa dạng và khả năng kết nối không giới hạn. Tuy nhiên, môi trường mạng cũng tiềm ẩn nhiều rủi ro khó lường đối với trẻ em như: tiếp xúc với nội dung độc hại, bạo lực, không phù hợp với lứa tuổi; nguy cơ nghiện game và mạng xã hội; lãng phí thời gian học tập; và các mối đe dọa về an ninh mạng, lừa đảo.

Trước những thách thức đó, việc giám sát và bảo vệ con cái trong không gian mạng đã trở thành một trong những mối quan tâm hàng đầu của các bậc phụ huynh. Tuy nhiên, việc giám sát thủ công 24/7 là điều bất khả thi và có thể gây ra những xung đột không đáng có trong gia đình.

Để giải quyết bài toán này, các phần mềm kiểm soát của phụ huynh (Parental Control Software) đã ra đời và ngày càng trở nên phổ biến. Đây là những công cụ công nghệ cho phép cha mẹ thiết lập các quy tắc, giới hạn và bộ lọc để tạo ra một môi trường số an toàn và lành mạnh cho con cái. Các phần mềm này không chỉ giúp ngăn chặn các mối nguy hiểm mà còn hỗ trợ phụ huynh trong việc giáo dục con về cách sử dụng Internet một cách có trách nhiệm.

Dự án "Phần mềm kiểm soát KidSafe" được phát triển nhằm cung cấp một giải pháp toàn diện, hiện đại và dễ sử dụng, giúp các bậc phụ huynh Việt Nam có thể bảo vệ con mình một cách hiệu quả. Hệ thống cho phép quản lý linh hoạt từ xa thông qua điện thoại thông minh, áp dụng các quy tắc gần như tức thời lên máy tính của con, từ đó xây dựng một hàng rào bảo vệ vững chắc trong thế giới số.

## 1.2. Giới thiệu hệ thống

### 1.2.1. Mục tiêu xây dựng

Dự án "Phần mềm kiểm soát KidSafe" được xây dựng với các mục tiêu cốt lõi, hướng tới việc tạo ra một công cụ hữu ích và đáng tin cậy cho các gia đình.

Cung cấp một công cụ quản lý từ xa, linh hoạt và tiện lợi: Mục tiêu quan trọng nhất là trao cho phụ huynh khả năng quản lý hoạt động sử dụng máy tính của con mọi lúc, mọi nơi thông qua một ứng dụng di động (Android). Phụ huynh không cần phải có mặt trực tiếp hay can thiệp thủ công vào máy tính của con, mọi thiết lập đều được đồng bộ hóa qua Internet.

Bảo vệ trẻ em khỏi các nội dung không phù hợp: Hệ thống phải có khả năng ngăn chặn truy cập vào các trang web có nội dung độc hại, bạo lực, người lớn hoặc các trang mạng xã hội, game online mà phụ huynh không mong muốn con mình tiếp xúc. Chức năng này giúp tạo ra một không gian mạng trong sạch và an toàn hơn.

Thiết lập kỷ luật về thời gian sử dụng thiết bị: Xây dựng chức năng quản lý thời gian sử dụng máy tính và Internet một cách hiệu quả. Phụ huynh có thể đặt ra giới hạn tổng thời gian sử dụng mỗi ngày, giúp con cân bằng giữa việc học, giải trí và các hoạt động thể chất khác, tránh tình trạng nghiện thiết bị điện tử.

Đảm bảo hệ thống hoạt động ổn định và không gây ảnh hưởng: Ứng dụng cài đặt trên máy tính của con phải được thiết kế để hoạt động như một dịch vụ nền, tiêu thụ ít tài nguyên hệ thống (CPU, RAM) và chạy một cách âm thầm, không làm gián đoạn hay ảnh hưởng đến hiệu suất học tập và sử dụng máy tính của trẻ.

### 1.2.2. Đối tượng khách hàng hướng tới

Việc xác định rõ đối tượng người dùng giúp hệ thống được thiết kế và phát triển sát với nhu cầu thực tế.

KidSafe hướng tới hai nhóm đối tượng chính:

Phụ huynh có con trong độ tuổi từ 6-16: Đây là nhóm đối tượng cốt lõi. Họ là những bậc cha mẹ quan tâm đến sự an toàn của con trên không gian mạng nhưng không có nhiều thời gian để giám sát trực tiếp. Họ cần một giải pháp công nghệ đơn giản, trực quan và hiệu quả để có thể quản lý từ xa. Nhu cầu của họ bao gồm:

Giao diện ứng dụng quản lý (trên điện thoại) phải dễ sử dụng, không yêu cầu kiến thức kỹ thuật phức tạp.

Các chức năng phải rõ ràng: chặn web, giới hạn thời gian.

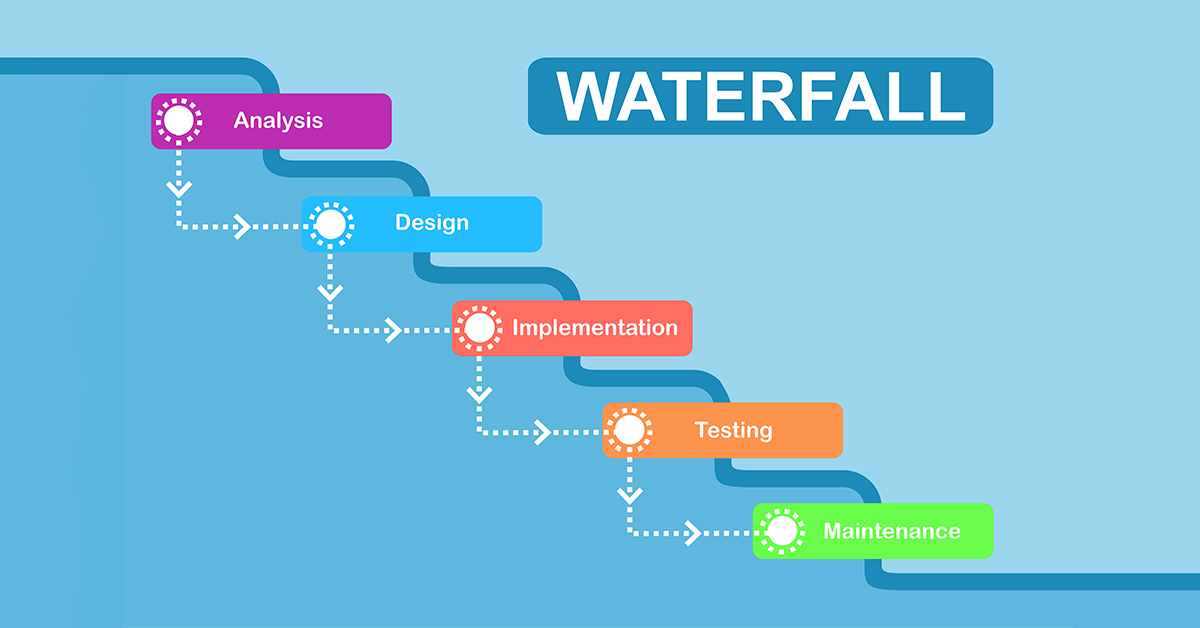
Hệ thống phải đáng tin cậy, các quy tắc phải được áp dụng ngay lập tức.

Trẻ em sử dụng máy tính Windows: Đây là người dùng cuối của ứng dụng thực thi. Hệ thống được thiết kế để hoạt động một cách minh bạch nhưng khó bị vô hiệu hóa. Các quy tắc được áp dụng một cách tự động, giúp trẻ dần hình thành thói quen sử dụng Internet có kỷ luật trong khuôn khổ mà cha mẹ đã đặt ra.

## 1.3. Công nghệ phát triển

### 1.3.1. Mô hình phát triển

Sau khi phân tích các yêu cầu và tính chất của dự án, mô hình Thác nước (Waterfall Model) được lựa chọn làm mô hình phát triển chính.



Hình 1.1: Mô hình phát triển Thác nước

Mô hình này chia quá trình phát triển thành các giai đoạn tuần tự và riêng biệt: *Phân tích yêu cầu -> Thiết kế hệ thống -> Lập trình -> Kiểm thử -> Triển khai và Bảo trì*.

Lý do lựa chọn mô hình Thác nước:

Yêu cầu rõ ràng: Các chức năng cốt lõi của hệ thống (chặn URL, quản lý thời gian, đồng bộ qua Firebase) đã được xác định rõ ràng và ít có khả năng thay đổi lớn trong quá trình phát triển. Điều này rất phù hợp với tính chất tuần tự của mô hình.

Dễ quản lý: Quy trình phát triển theo từng giai đoạn giúp dễ dàng theo dõi tiến độ, quản lý tài liệu và kiểm soát chất lượng ở mỗi bước.

Phù hợp với quy mô dự án: Đối với một dự án có phạm vi xác định, việc tuân thủ một kế hoạch nghiêm ngặt từ đầu đến cuối giúp đảm bảo dự án được hoàn thành đúng thời hạn và trong ngân sách dự kiến.

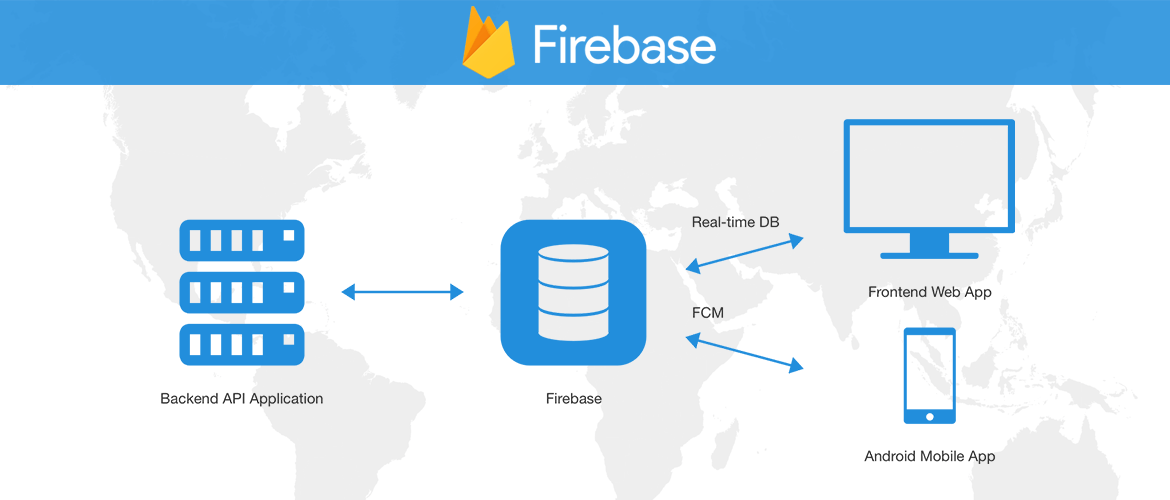
### 1.3.2. Công nghệ sử dụng

Để xây dựng một hệ thống đa nền tảng, đáp ứng tốt các yêu cầu về hiệu năng, đồng bộ hóa thời gian thực và trải nghiệm người dùng, dự án KidSafe đã lựa chọn và kết hợp một loạt các công nghệ hiện đại và mạnh mẽ. Các công nghệ này được phân chia thành ba nhóm chính: Nền tảng Backend, Ứng dụng Client cho Phụ huynh, và Ứng dụng Client trên máy tính của Con.

#### a. Nền tảng Backend và Cơ sở dữ liệu trung tâm



Trái tim của hệ thống là nền tảng Google Firebase, một giải pháp Backend-as-a-Service (BaaS) toàn diện. Việc lựa chọn Firebase giúp loại bỏ sự phức tạp trong việc quản lý hạ tầng máy chủ, đồng thời cung cấp các dịch vụ mạnh mẽ đã được tối ưu sẵn.



Firebase Realtime Database:

Giới thiệu: Là một cơ sở dữ liệu NoSQL được lưu trữ trên đám mây, cho phép lưu trữ và đồng bộ hóa dữ liệu giữa những người dùng theo thời gian thực. Dữ liệu được lưu dưới dạng một cây JSON lớn.

Vai trò: Đóng vai trò là cơ sở dữ liệu trung tâm, lưu trữ toàn bộ các quy tắc (URL bị chặn, quy tắc thời gian) và trạng thái của hệ thống.

Lý do chọn: Đây là lựa chọn then chốt của dự án. Realtime Database cung cấp khả năng đồng bộ hóa dữ liệu theo thời gian thực. Khi phụ huynh thay đổi một quy tắc trên ứng dụng Android, dữ liệu sẽ được đẩy (push) xuống ứng dụng PC gần như ngay lập tức mà không cần cơ chế làm mới (refresh) phức tạp.



Firebase Authentication:

Giới thiệu**:** Là một dịch vụ cung cấp các backend, SDK dễ sử dụng và thư viện giao diện người dùng sẵn có để xác thực người dùng cho ứng dụng.

Vai trò: Cung cấp dịch vụ xác thực người dùng an toàn, hỗ trợ đăng ký và đăng nhập qua email/mật khẩu.

Lý do chọn: Dễ dàng tích hợp, bảo mật cao, và quản lý người dùng một cách tập trung.

#### b. Ứng dụng di động cho Phụ huynh (Android Client)

Đây là "bộ điều khiển" của hệ thống, được phát triển dưới dạng một ứng dụng Android gốc để đảm bảo hiệu năng và trải nghiệm người dùng tốt nhất.



Ngôn ngữ lập trình: Java

Vai trò: Ngôn ngữ chính để xây dựng toàn bộ logic cho ứng dụng.

Lý do chọn: Java là ngôn ngữ lập trình truyền thống và rất mạnh mẽ cho nền tảng Android, với một hệ sinh thái thư viện và cộng đồng hỗ trợ khổng lồ.



Môi trường phát triển: Android Studio

Giới thiệu: Là Môi trường phát triển tích hợp (IDE) chính thức dành cho việc phát triển ứng dụng Android, được xây dựng bởi Google.

Vai trò: IDE (Integrated Development Environment) chính thức để viết mã, thiết kế giao diện, gỡ lỗi và đóng gói ứng dụng.

Lý do chọn: Cung cấp bộ công cụ đầy đủ và được tối ưu hóa nhất cho nền tảng Android, từ trình giả lập hiệu năng cao đến các công cụ phân tích và gỡ lỗi chuyên sâu, đảm bảo chất lượng cao nhất cho sản phẩm cuối cùng.

Thiết kế giao diện: XML Layouts

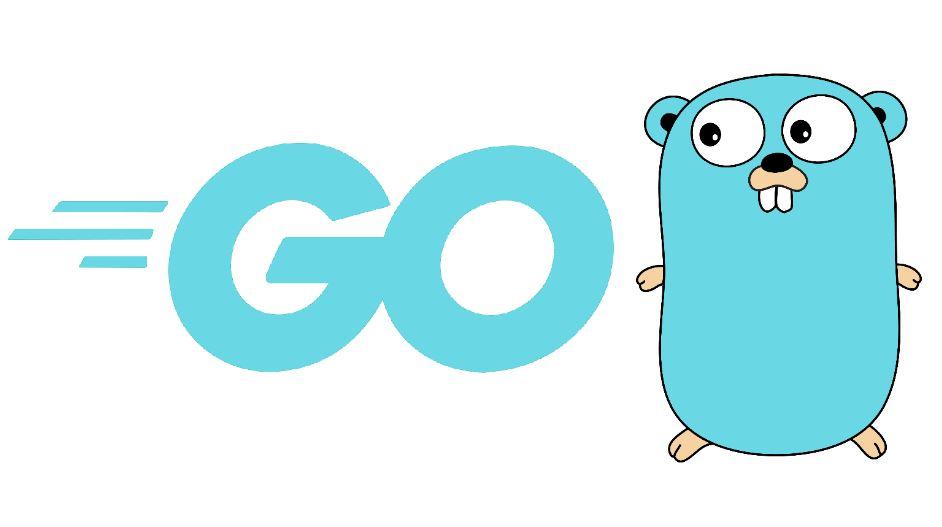
Vai trò: Xây dựng giao diện người dùng (UI) cho các màn hình như đăng nhập, danh sách URL, thiết lập thời gian.

Lý do chọn: Là phương pháp tiêu chuẩn để tạo giao diện gốc trên Android, đảm bảo tính tương thích và hiệu suất cao.

#### c. Ứng dụng trên máy tính của Con (PC Client)

Đây là "người thực thi" các quy tắc, được xây dựng với kiến trúc hai phần để tối ưu hóa hiệu năng và tài nguyên.

Dịch vụ lõi (Core Service – Backend tại Client):



Ngôn ngữ lập trình: Go (Golang)

Giới thiệu**:** Là một ngôn ngữ lập trình biên dịch, mã nguồn mở được phát triển bởi Google. Go được thiết kế để xây dựng các phần mềm đơn giản, hiệu quả và đáng tin cậy.

Vai trò**:** Xây dựng dịch vụ chạy nền (Windows Service) trên máy tính của con. Dịch vụ này chịu trách nhiệm lắng nghe các thay đổi từ Firebase và thực thi các hành động chặn (sửa file hosts, cập nhật Windows Firewall).

Lý do chọn: Go nổi bật với hiệu năng vượt trội, khả năng xử lý đồng thời (concurrency) hiệu quả và biên dịch ra file thực thi .exe nhỏ gọn, không cần dependency. Điều này cực kỳ quan trọng để dịch vụ có thể chạy âm thầm mà không làm ảnh hưởng đến máy tính của người dùng.

Cơ sở dữ liệu cục bộ: SQLite

Vai trò: Lưu trữ các log hoạt động và một số cấu hình cục bộ.

Lý do chọn: Là một thư viện CSDL gọn nhẹ, không cần cài đặt máy chủ, phù hợp để lưu trữ dữ liệu đơn giản ngay trên máy client.

Giao diện quản trị (UI Admin – Frontend tại Client):



Nền tảng: Electron

Giới thiệu: Là một framework mã nguồn mở cho phép xây dựng các ứng dụng máy tính đa nền tảng bằng các công nghệ web (HTML, CSS và JavaScript).

Vai trò: Đóng gói phần giao diện người dùng trên PC (được viết bằng web) thành một ứng dụng .exe hoàn chỉnh có thể chạy trên Windows.

Lý do chọn: Electron cho phép sử dụng các công nghệ web quen thuộc (HTML, CSS, JavaScript) để tạo ứng dụng cho máy tính, giúp đẩy nhanh quá trình phát triển giao diện.

### 1.3.3. Cơ sở dữ liệu

Hệ thống sử dụng Firebase Realtime Database, một cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng một cây JSON lớn. Cấu trúc này rất linh hoạt và phù hợp với yêu cầu đồng bộ hóa thời gian thực của dự án.

Lược đồ cơ sở dữ liệu chính bao gồm các thực thể sau:

families: Lưu thông tin tài khoản của phụ huynh. Khóa chính là familyId (UID từ Firebase Authentication), dùng để liên kết đến tất cả các dữ liệu khác của gia đình đó.

blockedUrls: Một tập hợp các đối tượng, mỗi đối tượng đại diện cho một trang web bị chặn. Thực thể này liên kết với families qua familyId.

timeRules: Một tập hợp các đối tượng, mỗi đối tượng là một quy tắc về thời gian (ví dụ: giới hạn sử dụng 120 phút/ngày). Thực thể này cũng liên kết với families qua familyId.

Mô hình này đảm bảo dữ liệu của mỗi gia đình được phân tách rõ ràng và bảo mật.

## 1.4. Kế hoạch triển khai và kiểm thử

### 1.4.1. Kế hoạch triển khai dự án

Dự án được triển khai theo các giai đoạn cụ thể để đảm bảo chất lượng và tiến độ.

Giai đoạn 1: Phân tích và Thiết kế (2 tuần):

Hoàn thiện tài liệu phân tích yêu cầu chức năng và phi chức năng.

Thiết kế kiến trúc hệ thống tổng thể, thiết kế chi tiết cơ sở dữ liệu trên Firebase.

Thiết kế giao diện người dùng (UI/UX) cho ứng dụng Android.

Giai đoạn 2: Phát triển Backend và PC Service (4 tuần):

Thiết lập dự án Firebase, cấu hình Authentication và Realtime Database.

Xây dựng các module chính của Core Service bằng Go: Lắng nghe Firebase, quản lý file hosts, quản lý Windows Firewall, theo dõi thời gian.

Giai đoạn 3: Phát triển ứng dụng Android (3 tuần):

Xây dựng giao diện và lập trình logic cho các chức năng: Đăng nhập, quản lý URL, quản lý thời gian.

Tích hợp Firebase SDK để ghi dữ liệu quy tắc.

Giai đoạn 4: Tích hợp và Kiểm thử (2 tuần):

Tích hợp giao diện Electron với Core Service trên PC.

Thực hiện kiểm thử tích hợp toàn bộ luồng từ Android -> Firebase -> PC.

Đóng gói ứng dụng PC thành bộ cài đặt .exe và ứng dụng Android thành file .apk.

Giai đoạn 5: Triển khai và Bàn giao (1 tuần):

Triển khai phiên bản cuối cùng.

Viết tài liệu hướng dẫn sử dụng và bàn giao sản phẩm.

### 1.4.2. Kế hoạch kiểm thử

Kiểm thử đơn vị (Unit Test):

Mục tiêu: Đảm bảo từng hàm, từng module nhỏ hoạt động chính xác.

Thực hiện: Viết các bài test cho các hàm quan trọng trong mã nguồn Go (ví dụ: hàm kiểm tra thời gian, hàm thêm quy tắc vào Firewall) và Java.

Kiểm thử tích hợp (Integration Test):

Mục tiêu: Kiểm tra sự phối hợp giữa các thành phần trong hệ thống.

Kịch bản chính: Thêm một quy tắc trên ứng dụng Android và xác minh rằng quy tắc đó được core-service trên PC nhận và thực thi chính xác trong thời gian ngắn nhất.

Kiểm thử hệ thống (System Test):

Mục tiêu: Đánh giá toàn bộ hệ thống trên một môi trường gần giống thực tế.

Thực hiện: Cài đặt ứng dụng trên các môi trường giả lập và thực tế (máy tính Windows 10/11, điện thoại Android phiên bản khác nhau) để kiểm tra hiệu năng, độ ổn định và tài nguyên tiêu thụ.

Kiểm thử chấp nhận của người dùng (UAT):

Mục tiêu: Đảm bảo sản phẩm đáp ứng đúng nhu cầu của người dùng cuối.

Thực hiện: Đưa phiên bản beta cho một nhóm nhỏ phụ huynh để sử dụng và thu thập phản hồi về giao diện, tính dễ sử dụng và các lỗi phát sinh.

## 1.5. Kết luận

Chương 1 đã trình bày một cái nhìn tổng quan và toàn diện về dự án "Phần mềm kiểm soát của phụ huynh KidSafe". Bằng cách phân tích bối cảnh thực tế, xác định mục tiêu rõ ràng và lựa chọn các công nghệ phù hợp, dự án đã đặt nền móng vững chắc cho việc xây dựng một giải pháp hiệu quả, đáp ứng nhu cầu cấp thiết của xã hội hiện đại.

Kiến trúc hệ thống với hai thành phần riêng biệt (Android và PC) được kết nối thông qua Firebase không chỉ giải quyết được bài toán quản lý từ xa mà còn đảm bảo khả năng đồng bộ hóa tức thời, một yếu tố quan trọng tạo nên sự tin cậy của sản phẩm. Kế hoạch triển khai và thử nghiệm chi tiết theo mô hình Thác nước sẽ là kim chỉ nam để đảm bảo sản phẩm cuối cùng đạt chất lượng cao, hoạt động ổn định và đáp ứng đúng kỳ vọng của đối tượng người dùng là các bậc phụ huynh.

Các chương tiếp theo sẽ đi sâu vào việc phân tích, thiết kế chi tiết và triển khai các chức năng đã được đề ra trong chương tổng quan này.

# CHƯƠNG 2: PHẦN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1. Phân tích hệ thống

Phân tích hệ thống là giai đoạn quan trọng nhằm xác định rõ các yêu cầu, chức năng và các yếu tố phi chức năng của hệ thống "KidSafe". Mục tiêu là xây dựng một hệ thống toàn diện, hiệu quả và đáng tin cậy để giúp phụ huynh bảo vệ con em mình khỏi các mối nguy hại trong không gian mạng và quản lý việc sử dụng thiết bị điện tử một cách hợp lý. Hệ thống được thiết kế theo mô hình client-server, bao gồm ba thành phần chính:

Ứng dụng trên thiết bị của trẻ (Kid's Agent): Được cài đặt trên máy tính (Windows) và điện thoại/máy tính bảng (Android) của trẻ. Ứng dụng này chạy ngầm để ghi nhận hoạt động và thực thi các quy tắc do cha mẹ thiết lập.

Ứng dụng dành cho cha mẹ (Parent's Dashboard): Giao diện quản lý dành cho cha mẹ, có thể truy cập qua ứng dụng di động hoặc trình duyệt web. Đây là nơi cha mẹ theo dõi, cấu hình và nhận các cảnh báo.

Hệ thống máy chủ (Backend Server): Trung tâm xử lý, lưu trữ dữ liệu, đồng bộ hóa thông tin và quản lý logic nghiệp vụ giữa thiết bị của cha mẹ và của trẻ.

### 2.1.1. Phân tích các chức năng của hệ thống

Các chức năng của hệ thống được chia thành các nhóm dựa trên vai trò của người dùng và các thành phần của hệ thống.

A. Chức năng chung và Quản lý tài khoản

Đăng ký/Đăng nhập: Phụ huynh có thể tạo tài khoản bằng email hoặc các tài khoản mạng xã hội. Hệ thống cần có cơ chế xác thực an toàn (mật khẩu mạnh, xác thực hai yếu tố).

Quản lý hồ sơ phụ huynh: Cho phép thay đổi thông tin cá nhân, mật khẩu, và các cài đặt tài khoản.

Quản lý hồ sơ trẻ em:

Thêm mới hồ sơ cho mỗi người con (tên, tuổi, ảnh đại diện).

Liên kết mỗi hồ sơ với một hoặc nhiều thiết bị cụ thể.

Chỉnh sửa hoặc xóa hồ sơ khi cần thiết.

Đồng bộ hóa dữ liệu: Tất cả các thiết lập, quy tắc và dữ liệu hoạt động được đồng bộ hóa gần như tức thời trên tất cả các thiết bị đã liên kết.

B. Chức năng dành cho Cha mẹ (Parent's Dashboard)

1. Bảng điều khiển và Giám sát (Dashboard & Monitoring)

- Tổng quan hoạt động: Hiển thị tóm tắt các thông tin quan trọng như tổng thời gian sử dụng thiết bị, các ứng dụng/trang web được truy cập nhiều nhất, các cảnh báo gần đây.

- Báo cáo chi tiết: Cung cấp các báo cáo theo ngày/tuần/tháng về:

Lịch sử duyệt web (các trang đã truy cập, các trang bị chặn).

Thời gian sử dụng từng ứng dụng.

Lịch sử tìm kiếm trên các công cụ như Google, YouTube.

- Quản lý yêu cầu từ trẻ: Nhận và phê duyệt/từ chối các yêu cầu xin thêm thời gian sử dụng hoặc truy cập vào nội dung bị chặn.

2. Thiết lập quy tắc và Giới hạn (Rules & Restrictions)

- Lọc nội dung web (Web Filtering):

Chặn các trang web dựa trên danh mục được định sẵn (ví dụ: nội dung người lớn, bạo lực, cờ bạc).

Tạo danh sách đen (Blacklist) để chặn các trang web cụ thể.

Tạo danh sách trắng (Whitelist) để chỉ cho phép truy cập vào một số trang web nhất định.

- Quản lý ứng dụng (Application Management):

Chặn hoàn toàn các ứng dụng không phù hợp.

Đặt giới hạn thời gian sử dụng hàng ngày cho từng ứng dụng (ví dụ: Facebook 30 phút/ngày).

- Quản lý thời gian sử dụng (Screen Time):

Đặt tổng thời gian tối đa được phép sử dụng thiết bị mỗi ngày.

Tạo lịch biểu sử dụng thiết bị (ví dụ: chặn hoàn toàn trong giờ học, giờ ngủ).

- Hàng rào địa lý (Geofencing):

Tạo các "vùng an toàn" (nhà, trường học) và nhận cảnh báo khi trẻ đi vào hoặc rời khỏi các khu vực này.

3. Cảnh báo và Thông báo (Alerts & Notifications)

Nhận thông báo tức thì khi trẻ cố gắng truy cập trang web/ứng dụng bị chặn.

Nhận cảnh báo khi trẻ ra/vào khu vực hàng rào địa lý.

Nhận cảnh báo SOS khẩn cấp kèm theo vị trí khi trẻ kích hoạt.

Thông báo về các hoạt động đáng ngờ (ví dụ: cài đặt ứng dụng mới, thay đổi cài đặt hệ thống).

C. Chức năng trên thiết bị của Trẻ em (Kid's Agent)

- Thực thi quy tắc: Ứng dụng chạy ngầm, tự động áp dụng các quy tắc do cha mẹ thiết lập mà không cần can thiệp thủ công.

- Giao diện tối thiểu: Cung cấp giao diện đơn giản để trẻ có thể:

Xem thời gian sử dụng thiết bị/ứng dụng còn lại trong ngày.

Gửi yêu cầu (request) đến cha mẹ để xin thêm thời gian hoặc gỡ bỏ chặn.

- Nút SOS: Cho phép trẻ gửi tín hiệu khẩn cấp đến cho cha mẹ chỉ bằng một nút bấm, kèm theo vị trí hiện tại.

- Chế độ ẩn (Stealth Mode): Ứng dụng hoạt động một cách kín đáo để không làm ảnh hưởng đến trải nghiệm sử dụng của trẻ.

- Ghi nhận hoạt động: Tự động ghi lại các hoạt động (duyệt web, sử dụng ứng dụng, vị trí) và gửi về máy chủ để cha mẹ xem.

### 2.1.2. Xác định các yêu cầu phi chức năng

Đây là các yêu cầu về chất lượng, ràng buộc kỹ thuật và trải nghiệm người dùng, đảm bảo hệ thống hoạt động hiệu quả và ổn định.

1. Hiệu năng (Performance)

Thời gian phản hồi:

Các thay đổi về quy tắc từ Parent's Dashboard phải được áp dụng trên thiết bị của trẻ trong vòng dưới 60 giây khi có kết nối mạng.

Bảng điều khiển của cha mẹ phải tải dữ liệu và hiển thị trong vòng dưới 3 giây.

Tài nguyên hệ thống: Ứng dụng trên máy của trẻ phải được tối ưu để:

Sử dụng dưới 5% CPU trung bình.

Tiêu thụ pin ở mức tối thiểu, không gây ảnh hưởng đáng kể đến thời lượng sử dụng thiết bị.

Khả năng chịu tải: Hệ thống máy chủ phải có khả năng xử lý đồng từ các thiết bị để đảm bảo hoạt động thông suốt khi có nhiều người dùng.

2. Độ tin cậy (Reliability)

Độ sẵn sàng: Hệ thống phải đạt thời gian hoạt động (uptime) 99.9%, đảm bảo cha mẹ có thể truy cập và giám sát gần như mọi lúc.

Khả năng chịu lỗi:

Trong trường hợp mất kết nối Internet, ứng dụng trên máy trẻ vẫn phải tiếp tục thực thi các quy tắc đã được đồng bộ hóa gần nhất.

Dữ liệu hoạt động được lưu tạm thời trên thiết bị và tự động gửi lên máy chủ ngay khi có kết nối trở lại.

3. Bảo mật (Security)

Mã hóa dữ liệu: Toàn bộ dữ liệu trao đổi giữa các thành phần của hệ thống (thiết bị trẻ em, máy chủ, thiết bị cha mẹ) phải được mã hóa bằng giao thức TLS 1.2 trở lên.

Bảo vệ dữ liệu cá nhân: Dữ liệu nhạy cảm của người dùng (tin nhắn, lịch sử web) phải được mã hóa khi lưu trữ và chỉ có phụ huynh được cấp quyền mới có thể truy cập.

Chống can thiệp (Tamper-Proofing): Ứng dụng trên thiết bị của trẻ phải có cơ chế tự bảo vệ, ngăn chặn việc gỡ cài đặt hoặc vô hiệu hóa trái phép nếu không có mật khẩu của phụ huynh.

Xác thực mạnh: Hệ thống phải hỗ trợ các chính sách mật khẩu mạnh và khuyến khích sử dụng xác thực hai yếu tố (2FA) để bảo vệ tài khoản phụ huynh.

4. Tính khả dụng (Usability)

Giao diện trực quan: Giao diện cho cha mẹ phải được thiết kế đơn giản, sạch sẽ, dễ hiểu và dễ thao tác, kể cả với người dùng không am hiểu về công nghệ.

Hướng dẫn rõ ràng: Quá trình cài đặt ứng dụng trên thiết bị của trẻ phải có hướng dẫn chi tiết từng bước, bao gồm cả việc cấp các quyền cần thiết cho ứng dụng.

5. Khả năng tương thích (Compatibility)

Hệ điều hành:

Ứng dụng Agent phải tương thích với Windows 10, 11 và Android từ phiên bản 8.0 trở lên.

Trình duyệt: Parent's Dashboard phiên bản web phải hoạt động tốt trên các trình duyệt phổ biến như Google Chrome, Firefox, Safari và Microsoft Edge.

6. Khả năng bảo trì và mở rộng (Maintainability & Scalability)

Kiến trúc module: Hệ thống cần được thiết kế theo kiến trúc microservices hoặc module hóa để dễ dàng bảo trì, sửa lỗi và phát triển các tính năng mới mà không ảnh hưởng đến toàn bộ hệ thống.

Khả năng mở rộng: Cả cơ sở dữ liệu và máy chủ ứng dụng phải có khả năng mở rộng theo chiều ngang để đáp ứng số lượng người dùng ngày càng tăng.

## 2.2. Thiết kế hệ thống

### 2.2.1. Biểu đồ Usecase tổng quát

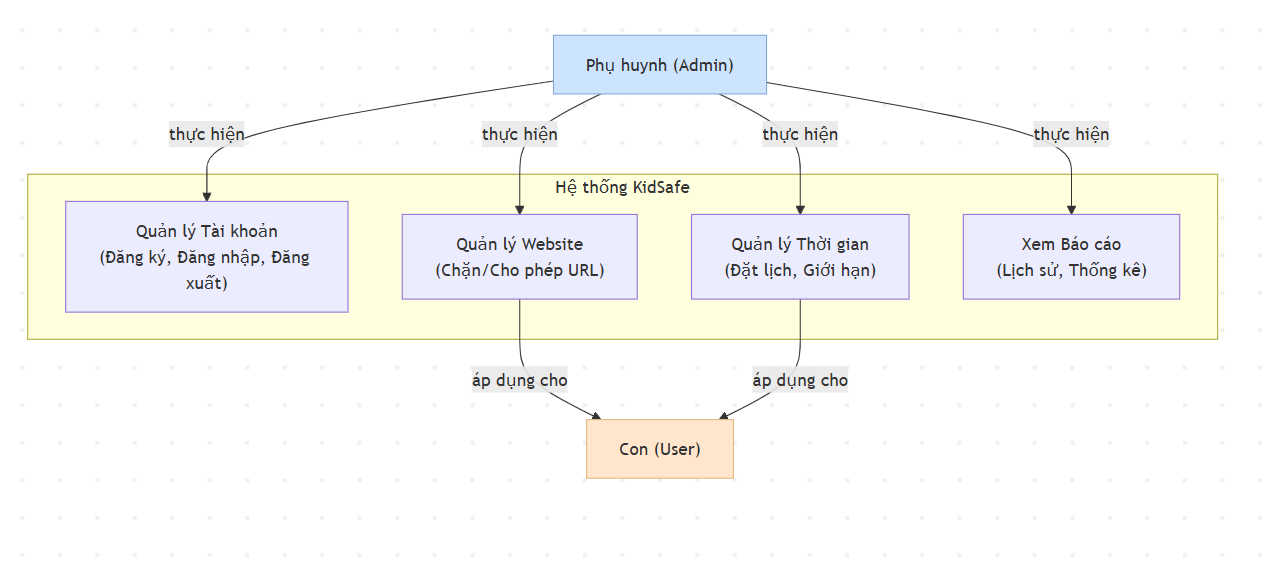
Biểu đồ Use Case tổng quát là biểu đồ ở mức khái quát cao, mô tả toàn bộ các chức năng chính của hệ thống và mối quan hệ giữa chúng với các tác nhân (actor) bên ngoài mà không đi sâu vào chi tiết các bước thực hiện. Mục đích của Biểu đồnày là cung cấp một cái nhìn tổng thể về hệ thống, giúp xác định phạm vi dự án và các nhóm chức năng lớn trước khi phân rã thành các Biểu đồchi tiết. Trong Biểu đồUse Case tổng quát, các thành phần chủ yếu bao gồm:

Actor (Tác nhân): Đại diện cho người dùng hoặc hệ thống bên ngoài tương tác với hệ thống.

Use Case (Trường hợp sử dụng): Các chức năng hoặc dịch vụ chính mà hệ thống cung cấp cho actor.

Quan hệ (Relationships): Các kết nối giữa actor và use case, hoặc giữa các use case với nhau (bao gồm include, extend, generalization).

Biểu đồ Use Case tổng quát thường được sử dụng ở giai đoạn đầu của phân tích yêu cầu để đảm bảo tất cả các bên liên quan hiểu rõ phạm vi và chức năng chính của hệ thống.



Hình 2.1. Biểu đồ Usecase tổng quát

Biểu đồ mô tả hai tác nhân chính là Phụ huynh (Admin) và Con (User).

Phụ huynh là người thực hiện các chức năng quản lý chính của Hệ thống KidSafe, bao gồm:

Quản lý Tài khoản: Các thao tác như Đăng ký, Đăng nhập, Đăng xuất.

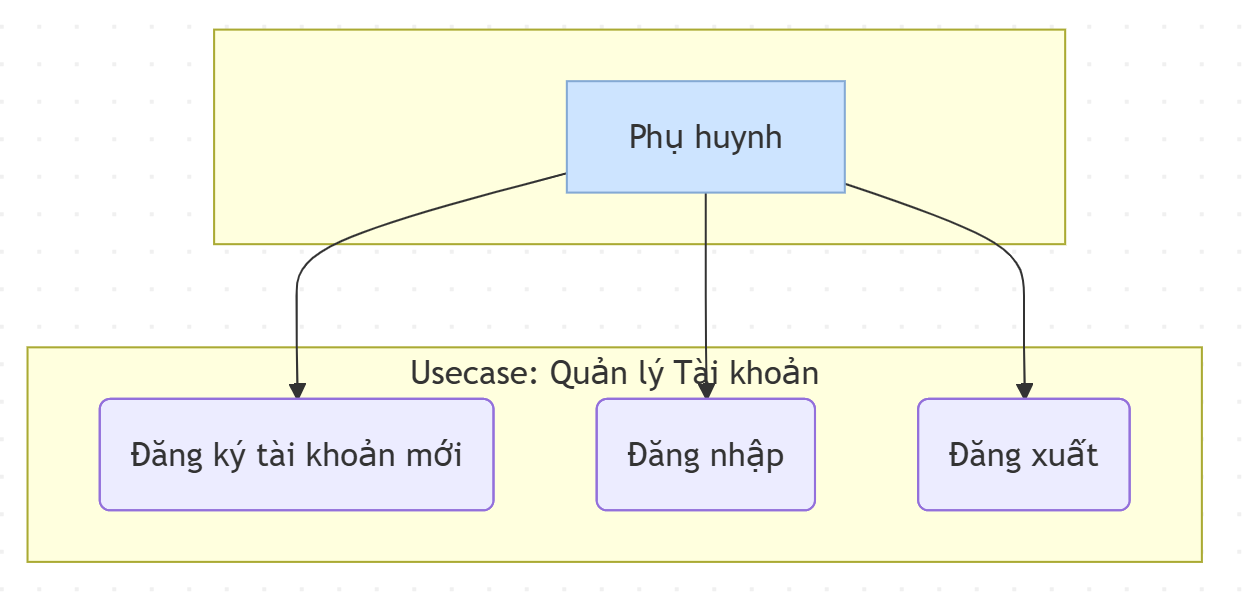
Quản lý Website: Chặn hoặc cho phép các URL cụ thể.

Quản lý Thời gian: Đặt lịch và giới hạn thời gian sử dụng thiết bị.

Xem Báo cáo: Theo dõi lịch sử và thống kê hoạt động của con.

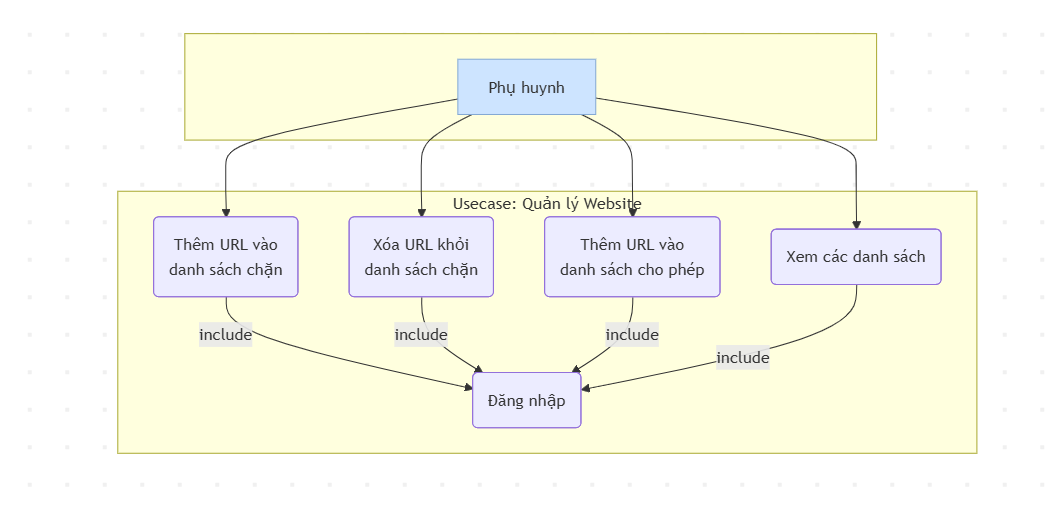
Con là người dùng chịu sự tác động của các quy tắc được thiết lập. Các chức năng Quản lý Website và Quản lý Thời gian sẽ được hệ thống "áp dụng cho" tài khoản của Con.

### 2.2.2. Biểu đồ Usecase Phân rã



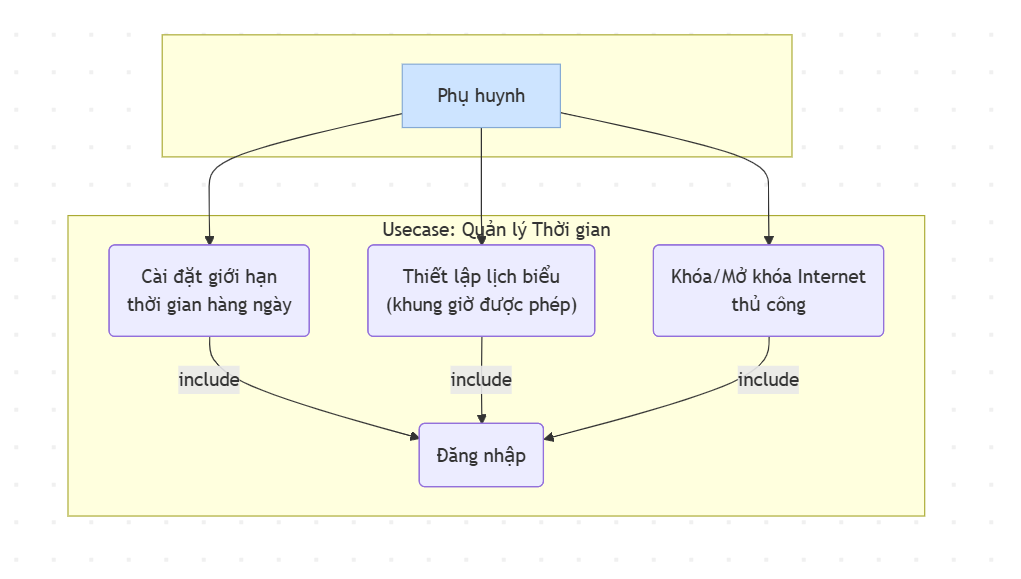
Hình 2.2. Phân rã Usecase quản lý tài khoản

Biểu đồ chi tiết hóa chức năng "Quản lý Tài khoản". Cho thấy Phụ huynh có thể thực hiện ba hành động cơ bản: Đăng ký tài khoản mới để tạo tài khoản, Đăng nhập để truy cập hệ thống và Đăng xuất để kết thúc phiên làm việc.



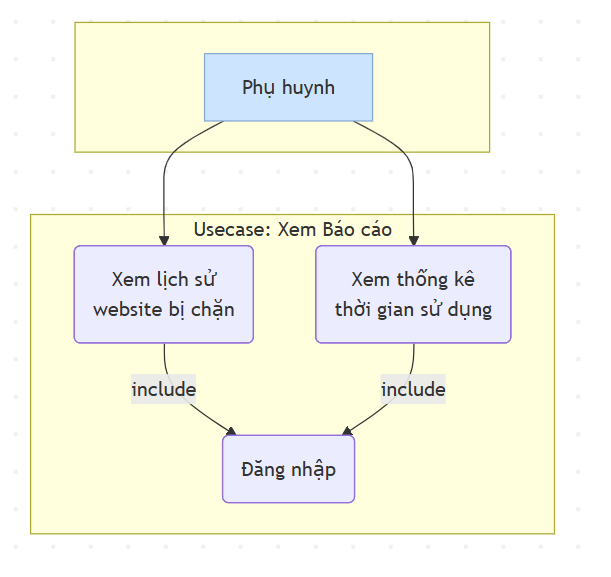
Hình 2.3. Phân rã Usecase quản lý Website

Biểu đồ này làm rõ các thao tác trong chức năng quản lý Website. Phụ huynh có thể Thêm URL vào danh sách chặn, Xóa URL khỏi danh sách chặn, Thêm URL vào danh sách cho phép, và Xem các danh sách đã thiết lập. Tất cả các hành động này đều yêu cầu Phụ huynh phải Đăng nhập vào hệ thống trước



Hình 2.4. Phân rã Usecase quản lý thời gian

Biểu đồ mô tả chi tiết các hành động thuộc quản lý Thời gian. Phụ huynh có thể Cài đặt giới hạn thời gian hàng ngày, Thiết lập lịch biểu (khung giờ được phép sử dụng), và Khóa/Mở Internet thủ công. Tương tự như quản lý website, các chức năng này cũng yêu cầu phải Đăng nhập trước.



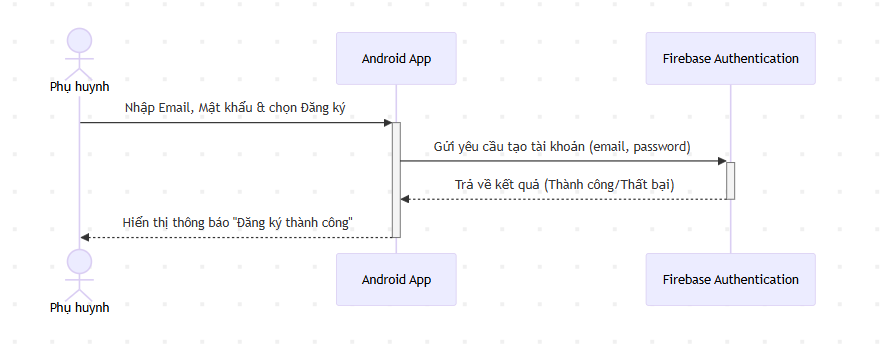
Hình 2.5. Phân rã Usecase xem báo cáo

Biểu đồ này phân rã chức năng xem báo cáo thành hai hành động chính: Xem lịch sử website bị chặn và Xem thống kê thời gian sử dụng. Cả hai chức năng này đều bắt buộc Phụ huynh phải Đăng nhập để đảm bảo tính riêng tư và bảo mật.

### 2.2.3. Biểu đồ tuần tự

Tổng quan về Biểu đồ tuần tự:

Biểu đồ tuần tự (Sequence Diagram) là một loại Biểu đồ trong UML dùng để mô tả cách các đối tượng trong hệ thống tương tác với nhau theo trình tự thời gian. Biểu đồnày tập trung thể hiện thông điệp (message) được trao đổi giữa các đối tượng, thứ tự thực hiện các hành động và khoảng thời gian diễn ra chúng. Các thành phần chính của Biểu đồ tuần tự bao gồm: các đối tượng (lifeline) được biểu diễn dưới dạng hình chữ nhật ở đầu cột dọc, đường sống (lifeline) thể hiện sự tồn tại của đối tượng theo thời gian, các mũi tên biểu thị thông điệp hoặc lời gọi phương thức giữa các đối tượng, và thanh kích hoạt (activation bar) thể hiện khoảng thời gian một đối tượng thực hiện một hành động. Biểu đồtuần tự thường được sử dụng để mô tả chi tiết các kịch bản (scenario) trong hệ thống, giúp lập trình viên và nhà phân tích hệ thống hiểu rõ luồng xử lý, quan hệ và thứ tự tương tác giữa các thành phần.



Hình 2.6. Biểu đồ tuần tự chức năng đăng ký tài khoản mới

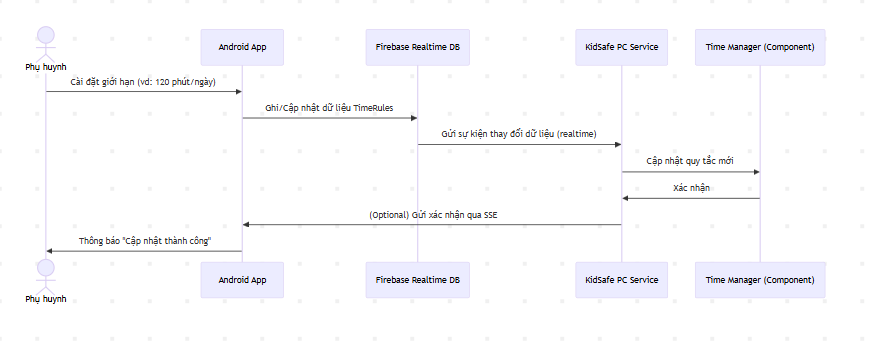
Biểu đồ mô tả luồng đăng ký tài khoản của Phụ huynh.

Phụ huynh nhập thông tin (Email, Mật khẩu) và chọn "Đăng ký" trên Android App.

Android App gửi yêu cầu tạo tài khoản đến Firebase Authentication.

Firebase Authentication xử lý và trả về kết quả (Thành công/Thất bại).

Android App nhận kết quả và hiển thị thông báo "Đăng ký thành công" cho Phụ huynh.



Hình 2.7. Biểu đồ tuần tự đặt giới hạn thời gian sử dụng

Biểu đồ này thể hiện quá trình đồng bộ hóa quy tắc thời gian từ cha mẹ đến máy tính của con.

Phụ huynh cài đặt giới hạn thời gian trên Android App.

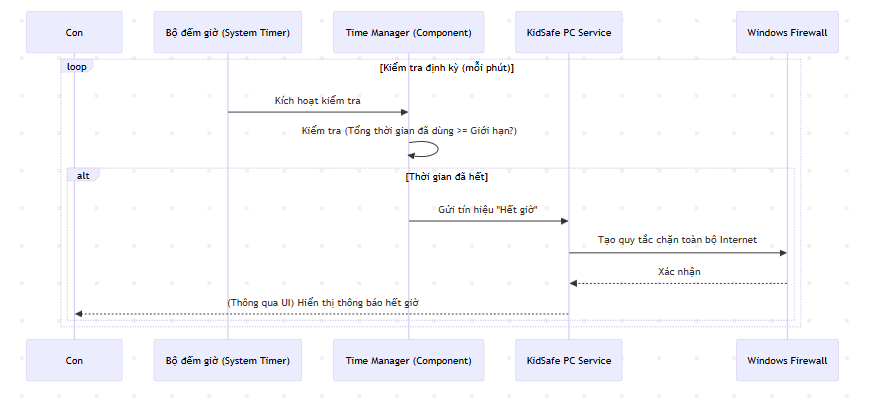
Ứng dụng ghi nhận và cập nhật dữ liệu (TimeRules) lên Firebase Realtime DB.

Firebase tự động gửi sự kiện thay đổi dữ liệu (realtime) đến KidSafe PC Service đang lắng nghe.

KidSafe PC Service nhận dữ liệu mới và yêu cầu Time Manager (Component) cập nhật quy tắc.

Time Manager xác nhận việc cập nhật.

Một thông báo "Cập nhật thành công" được gửi lại cho Phụ huynh.



Hình 2.8. Biểu đồ tuần tự hết thời gian sử dụng trên PC

Biểu đồ mô tả cơ chế hệ thống tự động chặn Internet khi hết thời gian.

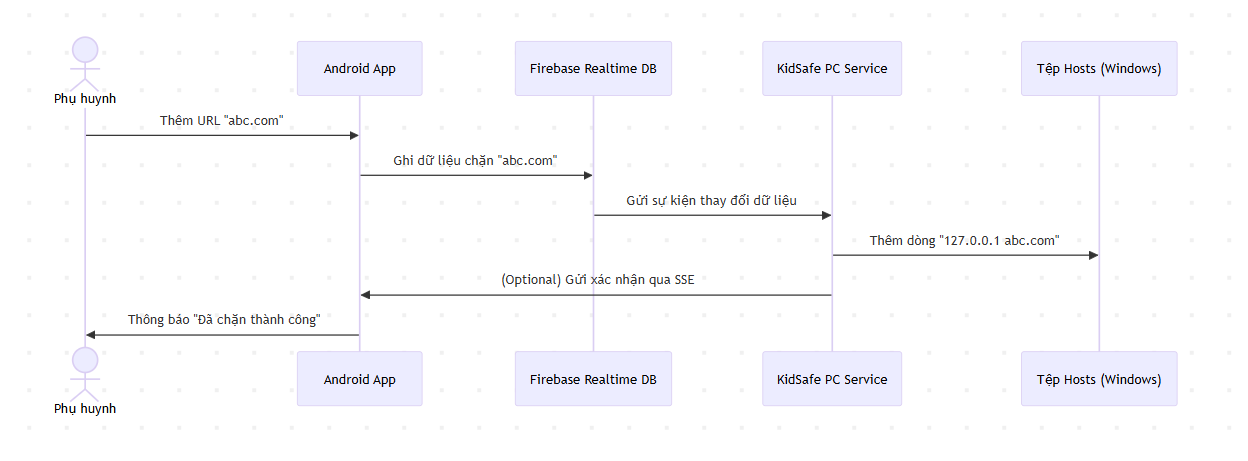
Bộ đếm giờ (System Timer) của hệ thống kích hoạt kiểm tra định kỳ (mỗi phút) tới Time Manager.

Time Manager kiểm tra xem tổng thời gian đã sử dụng có lớn hơn giới hạn không.

Khi điều kiện "Thời gian đã hết" được thỏa mãn (khối alt), Time Manager gửi tín hiệu "Hết giờ" đến KidSafe PC Service.

KidSafe PC Service ra lệnh cho Windows Firewall tạo quy tắc chặn toàn bộ kết nối Internet.

Đồng thời, một thông báo hết giờ được hiển thị trên giao diện người dùng (UI) cho Con biết.



Hình 2.9. Biểu đồ tuần tự phụ huynh chặn một trang web từ ứng dụng Android

Biểu đồ mô tả luồng hoạt động khi phụ huynh thêm một trang web vào danh sách chặn.

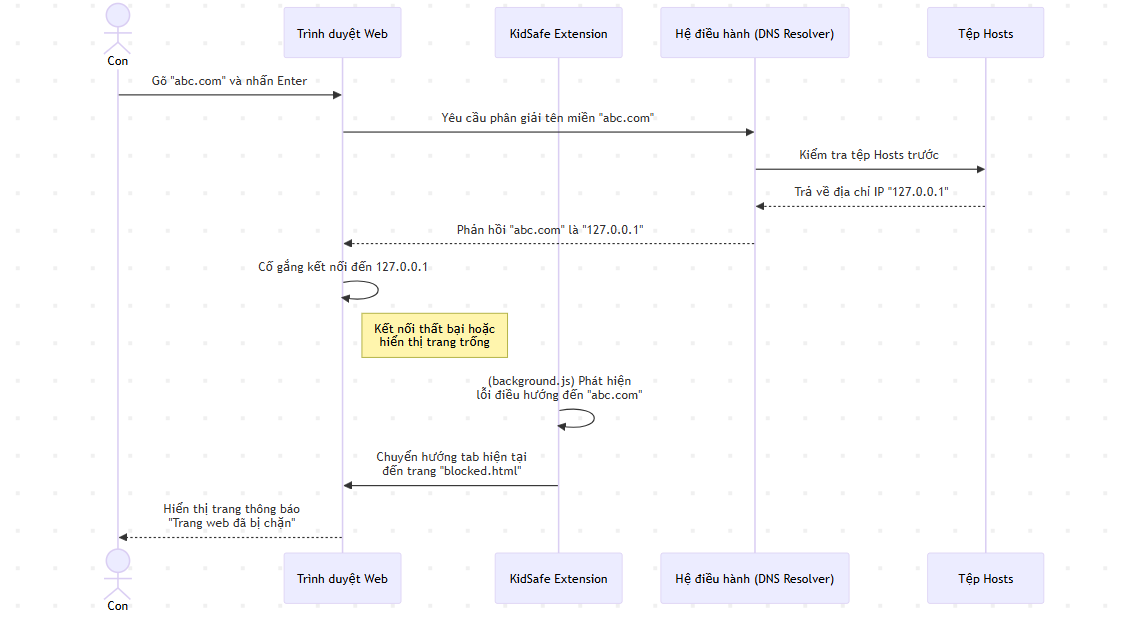
Phụ huynh nhập URL (ví dụ: "abc.com") trên Android App.

Ứng dụng ghi dữ liệu này lên Firebase Realtime DB.

Firebase gửi sự kiện thay đổi đến KidSafe PC Service.

KidSafe PC Service nhận được URL mới và thêm một dòng tương ứng (ví dụ: "127.0.0.1 abc.com") vào Tệp Hosts của Windows để chặn trang web này ở cấp hệ điều hành.

Thông báo thành công được gửi về cho Phụ huynh.



Hình 2.10. Biểu đô tuần tự con truy cập một trang web đã bị chặn

Biểu đồ này giải thích cách hệ thống ngăn chặn truy cập và hiển thị trang thông báo cho người dùng.

Con gõ "abc.com" vào Trình duyệt Web.

Trình duyệt gửi yêu cầu phân giải tên miền đến Hệ điều hành (DNS Resolver).

Hệ điều hành kiểm tra Tệp Hosts trước tiên và thấy rằng "abc.com" được trỏ về IP "127.0.0.1".

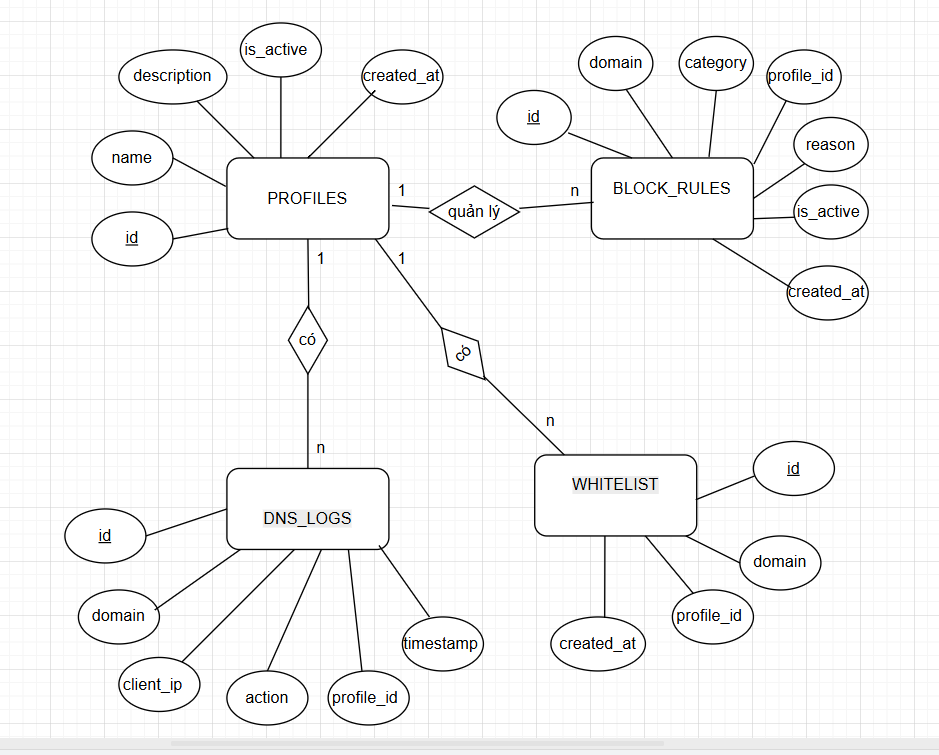
Trình duyệt cố gắng kết nối đến "127.0.0.1" nhưng thất bại.

KidSafe Extension (tiện ích mở rộng trên trình duyệt) phát hiện lỗi truy cập này, nhận diện đó là trang bị chặn và chuyển hướng tab hiện tại đến trang "blocked.html".

Con sẽ thấy trang thông báo "Trang web đã bị chặn".

### 2.2.4. Biểu đồ thực thể quan hệ

Biểu đồ thực thể – quan hệ (ERD) là công cụ trực quan quan trọng trong quá trình phân tích và thiết kế cơ sở dữ liệu cho các hệ thống thông tin. ERD giúp mô hình hóa toàn bộ các thành phần dữ liệu và mối quan hệ giữa chúng, từ đó đảm bảo tính nhất quán, dễ quản lý và thuận tiện cho việc truy xuất thông tin. Việc xây dựng ERD là bước nền tảng để các nhà phát triển và quản trị viên cơ sở dữ liệu hiểu rõ cấu trúc dữ liệu trước khi triển khai hệ thống.



Hinh 2.11. Biểu đồ thực thể quan hệ

Biểu đồ mô tả cấu trúc dữ liệu của hệ thống:

PROFILES: Thực thể trung tâm, lưu thông tin hồ sơ của mỗi đứa trẻ (tên, mô tả...).

BLOCK\_RULES: Lưu các quy tắc chặn web (tên miền, lý do chặn).

WHITELIST: Lưu các trang web được phép truy cập.

DNS\_LOGS: Ghi lại lịch sử các lần truy vấn DNS để phục vụ cho việc báo cáo.

Mối quan hệ: Một PROFILES quản lý nhiều (n) BLOCK\_RULES và nhiều (n) WHITELIST. Mỗi PROFILES cũng có (n) bản ghi DNS\_LOGS tương ứng. Đây là các mối quan hệ một-nhiều (1-n).

### 2.2.5. Biểu đồ lớp

Biểu đồ lớp mô tả cấu trúc tĩnh của hệ thống, bao gồm các lớp (trong trường hợp này là các struct chính trong Go), các thuộc tính và mối quan hệ giữa chúng. Biểu đồ này tập trung vào thành phần core-service là trái tim của ứng dụng PC.

CoreService: Lớp trung tâm, điều phối hoạt động của các thành phần khác.

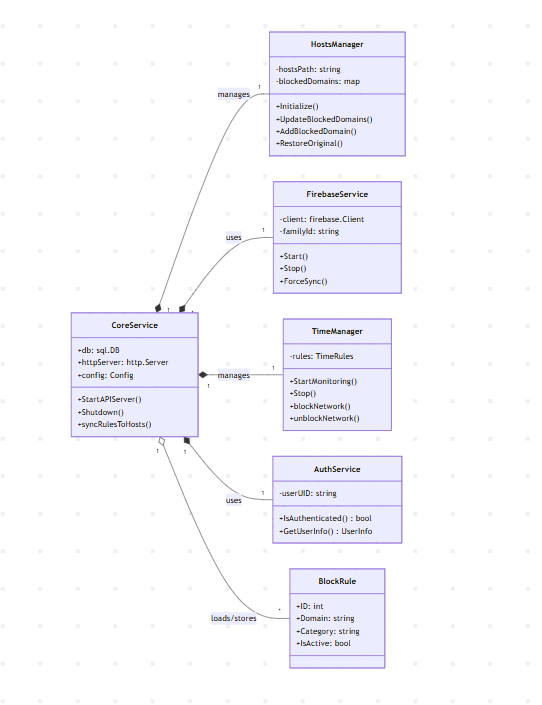
HostsManager: Chịu trách nhiệm đọc và ghi vào tệp hosts của Windows.

FirebaseService: Xử lý kết nối và đồng bộ dữ liệu với Firebase.

TimeManager: Quản lý các quy tắc về thời gian sử dụng.

AuthService: Xử lý xác thực người dùng.

Các lớp dữ liệu như BlockRule, Profile đại diện cho các thực thể trong cơ sở dữ liệu.



Hình 2.12. Biểu đồ lớp

Biểu đồ thể hiện kiến trúc phần mềm của ứng dụng trên PC:

CoreService: Là lớp điều phối chính, quản lý hoạt động của các thành phần khác.

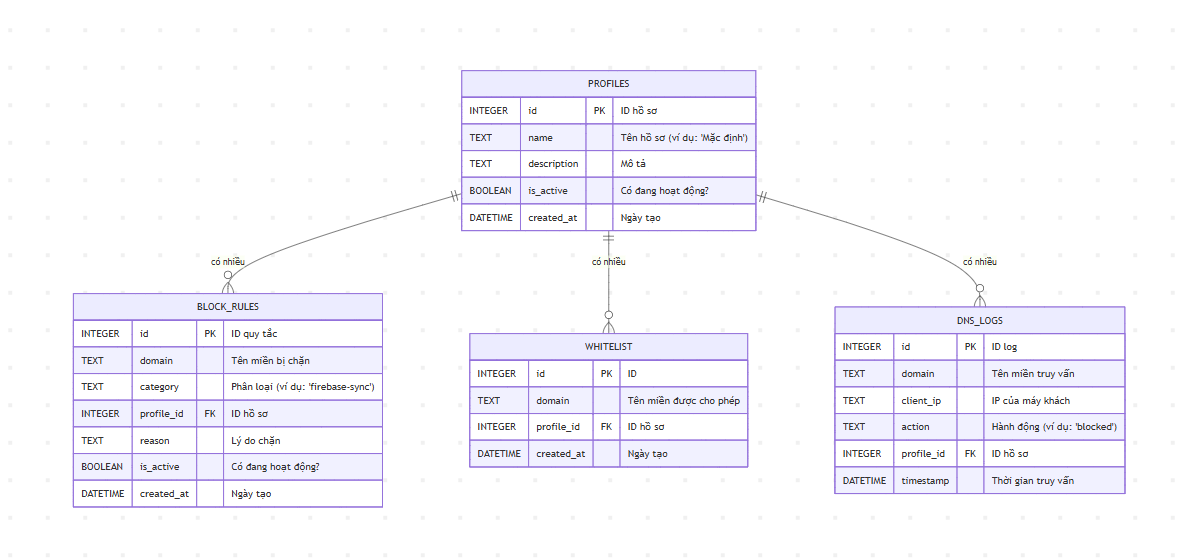
CoreService quản lý (manages) HostsManager (để sửa đổi tệp hosts) và TimeManager (để theo dõi thời gian sử dụng).

CoreService sử dụng (uses) FirebaseService (để đồng bộ dữ liệu với máy chủ) và AuthService (để xử lý xác thực).

CoreService tải và lưu trữ (loads/stores) dữ liệu dưới dạng các đối tượng BlockRule. Mối quan hệ này cho thấy sự phụ thuộc và cách các thành phần tương tác với nhau để thực thi các quy tắc.

### 2.2.6. Xây dựng CSDL (Database Schema)

Cấu trúc cơ sở dữ liệu của hệ thống được định nghĩa trong mã nguồn core-service (hàm initDB). Hệ thống sử dụng SQLite, một CSDL gọn nhẹ dựa trên tệp. Dưới đây là biểu đồ thực thể-kết hợp (ERD) và mô tả chi tiết các bảng.



Sơ đồ này là bản thiết kế chi tiết cho cơ sở dữ liệu SQLite cục bộ trên máy của trẻ.

PROFILES: Bảng chính chứa thông tin hồ sơ (ID, tên, trạng thái).

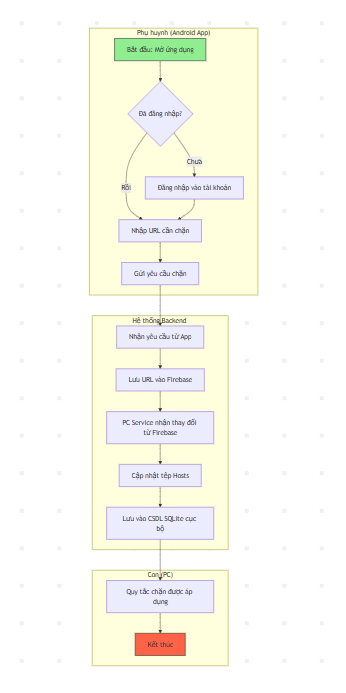
BLOCK\_RULES: Chứa các quy tắc chặn, mỗi quy tắc có profile\_id là khóa ngoại (FK) để liên kết với một hồ sơ cụ thể.

WHITELIST: Chứa các tên miền được cho phép, cũng liên kết với bảng PROFILES qua profile\_id.

DNS\_LOGS: Ghi lại nhật ký truy cập, bao gồm tên miền, IP của máy khách và profile\_id để biết hồ sơ nào đã thực hiện truy vấn. Các đường nối thể hiện mối quan hệ một-nhiều giữa PROFILES và các bảng còn lại.

### 2.2.7. Biểu đồ hoạt động

Biểu đồ hoạt động mô tả các luồng công việc hoặc quy trình. Nó cho thấy từng bước trong một quy trình và logic quyết định (rẽ nhánh).



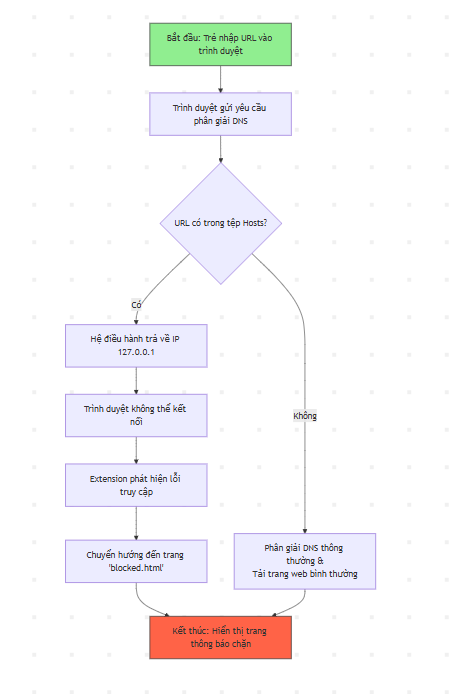
Biểu đồ này mô tả quy trình từ lúc phụ huynh thêm một URL mới cho đến khi quy tắc được áp dụng trên máy tính của con.

Phụ huynh (Android App): Mở ứng dụng, đăng nhập, nhập URL cần chặn và gửi yêu cầu.

Hệ thống Backend: Nhận yêu cầu, lưu URL vào Firebase.

Con (PC): PC Service nhận thấy sự thay đổi từ Firebase, tự động cập nhật tệp Hosts và lưu vào CSDL SQLite cục bộ.

Kết quả: Quy tắc chặn được áp dụng thành công trên máy tính.



Biểu đồ cho thấy các bước hệ thống thực hiện khi phát hiện một nỗ lực truy cập vào trang web bị chặn.

Bắt đầu: Trẻ nhập URL vào trình duyệt.

Trình duyệt gửi yêu cầu phân giải DNS.

Logic quyết định: Hệ thống kiểm tra "URL có trong tệp Hosts không?".

Nếu có: Hệ điều hành trả về IP 127.0.0.1, trình duyệt không thể kết nối, và Extension sẽ chuyển hướng đến trang thông báo chặn.

Nếu không: Yêu cầu được phân giải DNS bình thường và trang web được tải.

Kết thúc: Hiển thị trang báo chặn hoặc tải trang web thành công.