



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẢNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0041058

(51)⁷ H04M 1/725

(13) B

(21) 1-2018-01049

(22) 13/08/2016

(86) PCT/IB2016/054880 13/08/2016

(87) WO 2017/025942 16/02/2017

(30) 62/204,960 13/08/2015 US; 62/252,357 06/11/2015 US; 62/274,152 31/12/2015 US

(45) 25/09/2024 438

(43) 27/08/2018 365A

(73) SAMSUNG ELECTRONICS CO., LTD. (KR)

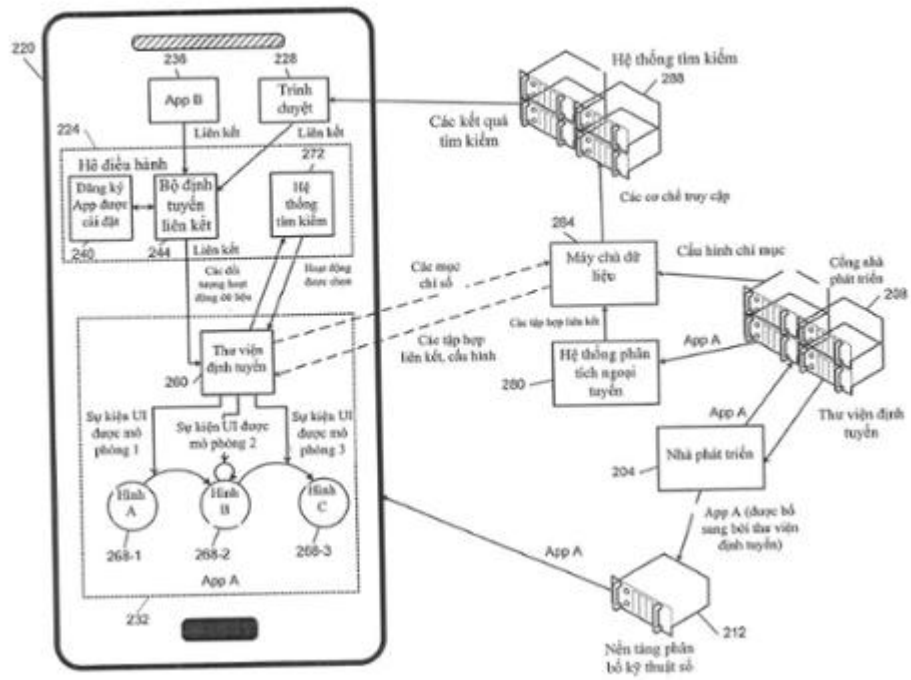
129, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, 16677, Republic of Korea

(72) DESINENI, Kalyan (US); LEVI, Matan (US); KRUSH, Moshe (US);
TSECHANSKY, Danny (US).

(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)

(54) HỆ THỐNG PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG VÀ PHƯƠNG PHÁP ĐIỀU
HÀNH HỆ THỐNG PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG

(57) Sáng chế đề cập đến hệ thống phát triển ứng dụng di động. Hệ thống này bao gồm công nhà phát triển nhận ứng dụng từ nhà phát triển và đề xuất thư viện định tuyến cho nhà phát triển để tăng cường ứng dụng này. Hệ thống phân tích ngoại tuyến phân tích ứng dụng để (i) xác định tập hợp của các hoạt động mà bộ xử lý trong ứng dụng này được lập trình để khôi phục nhằm đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ và (ii) xác định các tham số đối với mỗi trong số các hoạt động. Hệ thống phân tích ngoại tuyến tạo ra tập hợp của các liên kết mà mỗi trong số đó tương ứng với một trong số các hoạt động tương ứng. Thư viện định tuyến, được cài đặt là một phần của ứng dụng bổ sung trên thiết bị người dùng, nhận liên kết, từ hệ điều hành của thiết bị người dùng, nhận dạng hoạt động thứ nhất. Thư viện định tuyến bao gồm các lệnh để tạo yêu cầu khôi phục thứ nhất dựa trên các tham số tương ứng với hoạt động thứ nhất và truyền yêu cầu khôi phục thứ nhất đến bộ xử lý của ứng dụng bổ sung. Sáng chế cũng đề cập đến phương pháp điều hành hệ thống phát triển ứng dụng di động.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập chung đến sự phát triển ứng dụng di động và cụ thể hơn là liên kết ngầm đến các trạng thái riêng của các ứng dụng di động.

Tình trạng kỹ thuật của sáng chế

Phần này đề xuất thông tin nền tảng liên quan đến sáng chế và không nhất thiết là tình trạng kỹ thuật.

Ứng dụng (được gọi thay thế nhau theo sáng chế là “app”), như app di động, có thể chứa nhiều trạng thái ngầm. Ví dụ, trong app đánh giá chất lượng của các nhà hàng dựa trên dữ liệu ý kiến phương tiện xã hội, trang chi tiết cho mỗi nhà hàng sẽ được xem là trạng thái ngầm. Các trạng thái ngầm có thể đạt được từ trong app thông qua chuỗi của các thao tác người dùng có thể liên quan đến việc điều hướng thông qua nhiều màn hình trình đơn (hoặc, các hình ảnh) cũng như các tương tác với các thành phần giao diện người dùng. Mỗi trong số các màn hình trình đơn và các thành phần giao diện người dùng có thể được làm trung gian bởi bộ điều khiển hình ảnh duy nhất được liên kết với màn hình được hiển thị.

Thông thường, các trạng thái ngầm này truy cập được chỉ từ trong chính app đó. Các công cụ tìm kiếm web hoạt động bên ngoài app, ví dụ, không thể đạt được các trạng thái ngầm trong app. Điều này có nghĩa là khi người dùng thực hiện tìm kiếm web thông thường đối với các nhà hàng và muốn khảo sát một trong số các lựa chọn được trở lại trong app đánh giá nhà hàng chuyên dụng, người dùng sẽ phải sao chép và dán tên của kết quả tìm kiếm được chọn một cách thủ công vào trường tìm kiếm của app đánh giá nhà hàng và chỉ dẫn app xếp hạng nhà hàng để truy cập trạng thái ngầm bên trong của nó tương ứng với nhà hàng được chọn. Điều này biểu diễn tương tác bổ sung không mong muốn được yêu cầu của người dùng.

Nếu các trạng thái ngầm của các app có thể bị lộ ra với các app bên ngoài và các xử lý máy tính, người dùng có thể tận hưởng thêm tính năng và sự tiện lợi. Ví dụ, người dùng có thể bắt đầu tìm kiếm nhà hàng thích hợp sử dụng máy chủ tìm kiếm dựa trên internet và sau đó, bằng cách chọn một trong số các kết quả tìm kiếm

đó, được dẫn tự động đến trang được liên kết ngầm thích hợp của app xếp hạng nhà hàng chuyên dụng.

Tuy nhiên, việc thực hiện chức năng này yêu cầu sự nỗ lực của nhà phát triển và yêu cầu về chuyên môn về liên kết ngầm mà nhà phát triển này có thể không có. Khi sự phát triển app bị giới hạn bởi thời gian, ngân sách, hoặc chuyên môn, chức năng liên kết ngầm cho một vài hoặc thậm chí tất cả các trạng thái của app có thể không có sự ưu tiên đủ cao để được thực hiện.

Các hệ điều hành di động, như hệ điều hành iOS từ Apple Inc. và hệ điều hành Android từ Google Inc., có thể cho phép nhà phát triển cung cấp dữ liệu cho hệ điều hành để lập chỉ mục. Trong hệ điều hành iOS, nó có thể có dạng đối tượng `CSSearchableItem` (trong đó CS là viết tắt của Core Spotlight). Nhà phát triển có thể triển khai mã để đánh dấu các trạng thái nhất định của app bằng cách cung cấp thông tin về các trạng thái đó cho hệ điều hành. Trong hệ điều hành iOS, nó có thể có dạng đối tượng `NSUserActivity` (trong đó NS là viết tắt của NeXTSTEP).

Sau đó, người dùng có thể thực hiện tìm kiếm thông qua hệ điều hành, trong đó dữ liệu hoặc các trạng thái liên quan từ app của nhà phát triển được nhận dạng. Nếu người dùng chọn một trong số các hoạt động đó từ giao diện hệ điều hành, như hội thoại tìm kiếm hoặc trình đơn tác vụ gần đây, mã của nhà phát triển có thể khôi phục app về trạng thái tương ứng với hoạt động đánh dấu. Hoạt động đánh dấu trên thiết bị thứ nhất thậm chí có thể được tiếp tục trên thiết bị khác (giả sử app này cũng được cài đặt trên các thiết bị khác). Thông qua truyền thông liên thiết bị, hệ điều hành của thiết bị thứ nhất thông báo hệ điều hành của các thiết bị khác về trạng thái nào mà người dùng đã tương tác gần nhất.

Ngoài ra, nhà phát triển có thể xác định các kiểu dữ liệu nhất định, như các số điện thoại và tên từ cơ sở dữ liệu liên lạc được duy trì bởi app này, được lập chỉ mục bởi hệ điều hành. Khi người dùng chỉ báo cho hệ điều hành một trong số các đối tượng dữ liệu của chúng (ví dụ, liên lạc nhất định) là liên quan, mã của nhà phát triển có thể chuyển tiếp đến trạng thái của app trong đó thể hiện đối tượng dữ liệu để xem hoặc hiệu chỉnh.

Để tận dụng các khả năng hệ điều hành này, nhà phát triển có thể khai triển mã mà có thể khôi phục app với trạng thái dựa trên hoạt động hoặc đối tượng dữ

liệu được chỉ báo. Tuy nhiên, các hoạt động và các đối tượng dữ liệu này có thể được biết chỉ với hệ điều hành hoặc các hệ thống tìm kiếm được duy trì bởi nhà phát triển của hệ điều hành. Theo cách khác, các app bên thứ ba và các dịch vụ tìm kiếm có thể không được phép truy cập vào dữ liệu này và điều hướng trực tiếp vào các trạng thái hoặc các đối tượng dữ liệu trong app.

Ngoài ra, việc phát triển mã để hiển thị các đối tượng dữ liệu, lập chỉ mục các đối tượng dữ liệu, đánh dấu các hoạt động, và tiếp tục các hoạt động yêu cầu nỗ lực của nhà phát triển và không thể được hoàn thiện đối với trạng thái bất kỳ trong số các trạng thái của app ít hơn nhiều so với tất cả các trạng thái của app. Do đó, các khả năng hệ điều hành cải tiến này không thể khả dụng đối với các trạng thái khác nhau của app và cũng không thể truy cập được vào các app bên thứ ba và các hệ thống tìm kiếm.

Trên Fig.1, chức năng hệ điều hành cải tiến được minh họa bằng biểu đồ. Trên thiết bị người dùng, hệ điều hành 100 chạy app thứ nhất (được gọi là “App A”) 104. App A 104 bao gồm tập hợp biểu diễn của các hình, Hình A 108-1, Hình B 108-2, Hình C 108-3, Hình D 108-4, và Hình E 108-5 (gọi chung là, các hình 108). Các hình 108 có thể được quản lý bởi một hoặc nhiều bộ điều khiển hình, có thể được phát triển theo mẫu kết cấu phần mềm bộ điều khiển hình mẫu (model-view-controller - MVC).

Chỉ là ví dụ, App A 104 là app thông tin nhà hàng, Hình A 108-1 là trạng thái chủ (hoặc, mặc định) của App A 104 mà từ đó các tìm kiếm nhà hàng với món ăn, địa lý, v.v. có thể được thực hiện. Tiếp tục ví dụ này, Hình B 108-2 là giao diện app hai chiều thể hiện các địa điểm của nhà hàng, Hình C 108-3 là hình chi tiết nhà hàng, Hình D 108-4 là danh sách các nhà hàng với món ăn, và Hình E 108-5 là danh sách các nhà hàng được đánh giá gần đây. Theo các phương án thực hiện khác nhau, bao gồm ví dụ này, một hình có thể là mẫu được phổ biến với dữ liệu dành riêng cho thực thể. Ví dụ, Hình C 108-3 có thể có sơ đồ trực quan đối với nhà hàng xác định trong đó ảnh của nhà hàng sẽ được bố trí, vị trí, kích thước, và mặt trước của tên nhà hàng, cách mà các đánh giá được tóm tắt, v.v. Hình ảnh bản đồ trực quan sẽ được tạo đối tượng với dữ liệu tương ứng với nhà hàng cụ thể từ kho dữ liệu.

App A 104 chọn và cung cấp các đối tượng dữ liệu cho hệ điều hành 100 để lập chỉ mục. Ví dụ, App A 104 có thể cung cấp tên gọi của mỗi trong số các món ăn đặc trưng được bao gồm bởi App A 104. Sau đó hệ điều hành 100 có thể đưa ra các kết quả cho người dùng đang tìm kiếm tên món ăn đó.

App A 104 bao gồm bộ xử lý hoạt động 112 nhận tín hiệu hoạt động người dùng tiếp tục từ hệ điều hành 100. Ví dụ, tín hiệu hoạt động người dùng tiếp tục có thể xác định loại đồ ăn nhất định mà người dùng quan tâm. Sau đó bộ xử lý hoạt động 112 dẫn ra Hình D 108-4 và gắn Hình D 108-4 với các kết quả tương ứng với món ăn đặc trưng. App A 104 có thể chọn lọc liên quan đến các món ăn nào để đề xuất cho hệ điều hành 100 để lập chỉ mục vì đưa ra quá nhiều đối tượng dữ liệu có thể dẫn đến sự giảm độ thích hợp trung bình của chúng. Điều này có thể khiến hệ điều hành 100 xuống hạng hoặc thậm chí loại bỏ các đối tượng được lập chỉ mục khỏi các kết quả tìm kiếm không xuất hiện để có độ liên quan cao.

Do đó, App A 104 có thể bao gồm việc lập trình để lập chỉ mục danh sách các món ăn phổ biến nhất hoặc các món ăn mà tên gọi của chúng có khả năng là duy nhất với các món ăn. Ví dụ, trong khi “American” có thể là tên món ăn, có tỷ lệ nhầm lẫn cao vì thuật ngữ này áp dụng cho nhiều tìm kiếm khác không chỉ đơn giản là món ăn. Tuy nhiên, nếu người dùng chỉ ra sự quan tâm đến món ăn, như bằng cách xem xét lại danh sách nhà hàng với đồ ăn nhất định một hoặc nhiều lần, App A 104 có thể thêm tên món ăn đó vào danh sách các đối tượng để lập chỉ mục bởi hệ điều hành 100.

Ngoài ra, App A 104 có thể chỉ báo cho hệ điều hành 100, sử dụng đối tượng hoạt động, mà hiện tại người dùng xem xét các kết quả của nhà hàng đối với món ăn đặc trưng. Điều này cho phép hệ điều hành 100 duy trì về cơ bản lịch sử của các hoạt động được thực hiện bởi người dùng trong App A 104. Ngoài ra, hoạt động gần đây nhất được tham gia bởi người dùng có thể được chia sẻ với thiết bị khác, cho phép sự chuyển vùng của tương tác người dùng với hoạt động đó từ một thiết bị đến thiết bị khác. Đưa đối tượng hoạt động vào hệ điều hành, cho phép sự khôi phục hoạt động sau đó trên cùng thiết bị hoặc các thiết bị khác, được đề cập theo sáng chế khi đánh dấu hoạt động này.

Các đối tượng hoạt động có thể được gửi đến hệ điều hành 100 khi người dùng nhập Hình B 108-2 để xem các kết quả nhà hàng trong việc hiển thị bản đồ. Ví dụ, đối tượng hoạt động được gửi đến hệ điều hành 100 có thể bao gồm vĩ độ và kinh độ ở trung tâm của bản đồ. Nếu người dùng điều chỉnh trung tâm của bản đồ, App A 104 có thể đề xuất đối tượng hoạt động được cập nhật vào hệ điều hành 100 chỉ báo điểm trung tâm mới. App A 104 còn có thể bao gồm trong dữ liệu bổ sung của đối tượng hoạt động như các bộ lọc. Ví dụ, người dùng có thể bị giới hạn các kết quả được hiển thị chỉ đối với các nhà hàng đó mà hiện tại mở cửa. Bộ lọc này có thể được nhận dạng trong đối tượng hoạt động được đề xuất cho hệ điều hành 100.

Bộ xử lý hoạt động 112 có thể nhận tín hiệu hoạt động người dùng tiếp tục từ hệ điều hành 100 chỉ báo là người dùng quan tâm đến việc tiếp diễn hình ảnh bản đồ của các nhà hàng ở điểm trung tâm nhất định với tập hợp các bộ lọc nhất định. Chỉ với ví dụ, trong hệ điều hành iOS, nó có thể có dạng cuộc gọi `continueuserActivity` cho ủy nhiệm app của app. Sau đó bộ xử lý hoạt động 112 dẫn ra Hình B 108-2 với các tham số được đề xuất và thể hiện người dùng với hình ảnh bản đồ mong muốn. Tín hiệu hoạt động người dùng tiếp tục có thể được nhận từ thiết bị khác mà người dùng xem trước để hiển thị bản đồ. Như được thể hiện trên Fig.1, bộ xử lý hoạt động 112 được lập trình để dẫn ra Hình B 108-2, Hình D 108-4, và Hình E 108-5, nhưng không phải Hình A 108-1 hoặc Hình C 108-3.

Trong hệ điều hành 100, chỉ mục tìm kiếm 120 nhận các đối tượng dữ liệu từ App A 104. Chỉ mục tìm kiếm 120 cũng nhận các đối tượng dữ liệu từ các app khác và, theo một số phương án thực hiện, từ chính hệ điều hành 100, như tên gọi của các cài đặt thiết bị được truy cập thường xuyên. Chỉ làm ví dụ, trong hệ điều hành iOS, chỉ mục tìm kiếm 120 có thể được gọi là Spotlight và các đối tượng dữ liệu được đề xuất cho chỉ mục tìm kiếm 120 có thể có dạng các đối tượng `CSSearchableItem`.

Chỉ mục tìm kiếm 120 cũng có thể nhận siêu dữ liệu hoạt động từ bộ theo dõi hoạt động 124. Bộ theo dõi hoạt động 124 nhận các đối tượng hoạt động, bao gồm từ App A 104. Đối tượng hoạt động có thể có dạng đối tượng `NSUserActivity`. Đối tượng hoạt động được nhận với bộ theo dõi hoạt động 124 có thể bao gồm siêu dữ liệu tương tự với đối tượng dữ liệu. Siêu dữ liệu có thể được lập chỉ mục với chỉ mục tìm kiếm 120. Siêu dữ liệu có thể có dạng đối tượng

CSSearchableItemAttributeSet, có thể có cùng dạng được sử dụng trong các đối tượng CSSearchableItem.

Các hoạt động được chỉ ra là công khai có thể được chia sẻ với chỉ mục đám mây được duy trì bởi nhà phát triển của hệ điều hành 100. Giao diện chỉ mục đám mây 128 đề xuất các hoạt động công khai cho chỉ mục đám mây này. Các hoạt động công khai này được lập chỉ mục chọn lọc bởi chỉ mục đám mây, cho phép người dùng khác tìm kiếm và tìm các hoạt động được tạo ra bởi App A 104 ngay cả khi App A 104 không được cài đặt trên các thiết bị của họ.

Bộ điều khiển chuyển giao 132 chia sẻ hoạt động gần nhất với các thiết bị khác. Trong các phương án thực hiện khác nhau, các thiết bị khác được xác thực cho bộ điều khiển chuyển giao 132 để chỉ báo là chúng cho phép để nhận các hoạt động gần nhất từ thiết bị này. Các thiết bị khác này, do đó, có thể cho phép người dùng bắt đầu hoạt động trên thiết bị trong đó hệ điều hành 100 được thực thi và bắt đầu lại hoạt động trên các thiết bị khác.

Giao diện tìm kiếm 136 cho phép người dùng thực hiện tìm kiếm, như bằng cách nhập câu hỏi văn bản. Chỉ mục tìm kiếm 120 đưa ra các kết quả tìm kiếm liên quan cho giao diện tìm kiếm 136 và, nhờ lựa chọn người dùng, kết quả được chọn bởi người dùng được cung cấp cho bộ theo dõi hoạt động 124. Bộ theo dõi hoạt động 124 nhận dạng app nào tương ứng với kết quả được chọn bởi người dùng và gửi tín hiệu hoạt động người dùng tiếp tục đến app liên quan. Tín hiệu hoạt động người dùng tiếp tục này có thể bao gồm bộ chỉ báo liệu tín hiệu hoạt động người dùng tiếp tục có liên quan đến đối tượng dữ liệu hoặc đối tượng hoạt động hay không.

Bản chất kỹ thuật của sáng chế

Hệ thống phát triển ứng dụng di động bao gồm công nhà phát triển và hệ thống phân tích ngoại tuyến. Công nhà phát triển này được tạo cấu hình để nhận sao chép ứng dụng thứ nhất từ nhà phát triển thứ nhất và cung cấp thư viện định tuyến cho nhà phát triển thứ nhất để sát nhập vào ứng dụng thứ nhất trước khi nhà phát triển thứ nhất phân phối ứng dụng bổ sung thông qua nền tảng phân phối số. Thư viện định tuyến được sát nhập vào ứng dụng thứ nhất để tạo ứng dụng bổ sung. Hệ thống phân tích ngoại tuyến được tạo cấu hình để phân tích ứng dụng thứ nhất để

(i) xác định tập hợp các hoạt động mà bộ xử lý trong ứng dụng thứ nhất được lập trình để khôi phục việc đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ và (ii) xác định các tham số cho mỗi hoạt động trong số tập hợp các hoạt động. Hệ thống phân tích ngoại tuyến được tạo cấu hình để tạo tập hợp liên kết. Mỗi liên kết trong số tập hợp liên kết tương ứng với hoạt động tương ứng trong số tập hợp các hoạt động. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, theo sau việc cài đặt ứng dụng bổ sung từ nền tảng phân phối số trên các thiết bị người dùng, nhận liên kết từ hệ điều hành của các thiết bị người dùng. Liên kết nhận dạng hoạt động thứ nhất. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để tạo yêu cầu khôi phục thứ nhất dựa trên các tham số tương ứng với hoạt động thứ nhất và truyền yêu cầu khôi phục thứ nhất đến bộ xử lý của ứng dụng bổ sung.

Theo các dấu hiệu khác, các tham số tương ứng với hoạt động thứ nhất bao gồm kiểu hoạt động và tên hoạt động. Theo các dấu hiệu khác, yêu cầu khôi phục thứ nhất được truyền bởi cuộc gọi `continueuserActivity` cho ủy nhiệm app của ứng dụng bổ sung. Theo các dấu hiệu khác, hoạt động thứ nhất bao gồm đối tượng `NSNuserActivity`.

Theo các dấu hiệu khác, hệ thống phân tích ngoại tuyến được tạo cấu hình để (i) phân tích ứng dụng thứ nhất để xác định tập hợp các đối tượng dữ liệu mà bộ xử lý trong ứng dụng thứ nhất được lập trình để truy cập đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ, (ii) xác định các tham số cho mỗi đối tượng dữ liệu trong số tập hợp các đối tượng dữ liệu, và (iii) tạo tập hợp liên kết thứ hai. Mỗi liên kết trong số tập hợp liên kết thứ hai tương ứng với đối tượng dữ liệu tương ứng của tập hợp các đối tượng dữ liệu. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, theo sau việc cài đặt của ứng dụng bổ sung trong thiết bị người dùng, nhận liên kết thứ hai từ hệ điều hành của thiết bị người dùng. Liên kết thứ hai nhận dạng đối tượng dữ liệu thứ nhất. Các lệnh còn được tạo cấu hình để tạo yêu cầu khôi phục thứ hai dựa trên các tham số tương ứng với đối tượng dữ liệu thứ nhất và truyền yêu cầu khôi phục thứ hai đến bộ xử lý của ứng dụng bổ sung.

Theo các dấu hiệu khác, đối tượng dữ liệu thứ nhất là đối tượng `CSSearchableItem`. Theo các dấu hiệu khác, bằng cách sạt nhập thư viện định tuyến, ứng dụng bổ sung được tạo cấu hình để đăng ký mẫu mã nhận dạng tài nguyên đồng

nhất (uniform resource identifier - URI) thứ nhất với hệ điều hành nhờ việc cài đặt. Các liên kết thích hợp với mẫu URI thứ nhất. Theo các dấu hiệu khác, mẫu URI thứ nhất xác định sơ đồ bao gồm chuỗi văn bản kết thúc với dấu hai chấm và hai dấu gạch chéo lên. Sơ đồ này là duy nhất đối với ứng dụng bổ sung. Chuỗi văn bản là sự móc nối của mã nhận dạng văn bản được liên kết với công nhà phát triển và tên của ứng dụng thứ nhất. Liên kết này bắt đầu với sơ đồ này.

Theo các dấu hiệu khác, hệ thống phân tích ngoại tuyến được tạo cấu hình để chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi các sự kiện giao diện người dùng (UI) được khởi tạo bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất. Hệ thống phân tích ngoại tuyến được tạo cấu hình để lưu trữ chuỗi các sự kiện UI làm cấu trúc dữ liệu thứ nhất tương ứng với trạng thái thứ nhất. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, đáp lại người dùng của thiết bị người dùng bắt gặp trạng thái thứ nhất trong lúc sử dụng ứng dụng bổ sung, truyền đánh dấu đến hệ điều hành. Các lệnh được tạo cấu hình để, đáp lại nhận yêu cầu khôi phục chỉ báo trạng thái thứ nhất từ hệ điều hành, phát lại chuỗi của các sự kiện UI từ cấu trúc dữ liệu thứ nhất.

Theo các dấu hiệu khác, hệ thống phân tích ngoại tuyến được tạo cấu hình để chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi của các sự kiện giao diện người dùng (UI) được khởi động bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất tương ứng với đối tượng dữ liệu thứ nhất. Hệ thống phân tích ngoại tuyến được tạo cấu hình để lưu trữ chuỗi các sự kiện UI là cấu trúc dữ liệu thứ nhất tương ứng với đối tượng dữ liệu thứ nhất. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, đáp lại việc chạy ứng dụng bổ sung, truyền chọn lọc đối tượng dữ liệu thứ nhất đến hệ điều hành. Các lệnh được tạo cấu hình để, đáp lại nhận yêu cầu khôi phục chỉ báo đối tượng dữ liệu thứ nhất từ hệ điều hành, phát lại chuỗi các sự kiện UI từ cấu trúc dữ liệu thứ nhất.

Theo các dấu hiệu khác, liên kết bao gồm chọn lọc dữ liệu nối tiếp. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để giải mã các tham số tương ứng với hoạt động thứ nhất từ dữ liệu nối tiếp trong liên kết này. Theo các dấu hiệu khác, liên kết này bao gồm chọn lọc mã nhận dạng duy nhất. Thư viện định tuyến

bao gồm các lệnh được tạo cấu hình để truy hồi các tham số tương ứng với hoạt động thứ nhất dựa trên mã nhận dạng duy nhất.

Theo các dấu hiệu khác, hệ thống phát triển ứng dụng di động bao gồm máy chủ dữ liệu được tạo cấu hình để lưu trữ các tham số tương ứng với các hoạt động được xác định bởi hệ thống phân tích ngoại tuyến. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, nhờ lần chạy ứng dụng bổ sung thứ nhất, tải xuống các tham số tương ứng với ít nhất hoạt động thứ nhất từ máy chủ dữ liệu để lưu trữ trong khu vực kho dữ liệu đến thư viện định tuyến. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để truy hồi các tham số tương ứng với hoạt động thứ nhất từ kho dữ liệu đáp lại mã nhận dạng duy nhất.

Hệ thống bao gồm hệ thống phát triển ứng dụng di động và hệ thống tìm kiếm nêu trên được tạo cấu hình để trả lại các kết quả về các thiết bị người dùng đáp lại tìm kiếm được ủy quyền bởi người dùng của các thiết bị người dùng. Kết quả thứ nhất trong số các kết quả được trả lại bao gồm liên kết này. Đáp lại việc chọn kết quả thứ nhất bởi người dùng, liên kết này được truyền đến thư viện định tuyến bởi hệ điều hành.

Phương pháp điều hành hệ thống phát triển ứng dụng di động bao gồm các bước nhận sao chép ứng dụng thứ nhất từ nhà phát triển thứ nhất. Phương pháp này bao gồm bước cung cấp thư viện định tuyến cho nhà phát triển thứ nhất để sát nhập vào ứng dụng thứ nhất trước khi nhà phát triển thứ nhất phân phối ứng dụng bổ sung thông qua nền tảng phân phối số. Thư viện định tuyến được sát nhập vào ứng dụng thứ nhất để tạo ứng dụng bổ sung. Phương pháp này bao gồm phân tích ứng dụng thứ nhất để (i) xác định tập hợp các hoạt động mà bộ quản lý trong ứng dụng thứ nhất được lập trình để khôi phục việc đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ và (ii) xác định các tham số cho mỗi hoạt động trong số tập hợp các hoạt động. Phương pháp này bao gồm bước tạo ra tập hợp liên kết. Mỗi liên kết trong số tập hợp liên kết tương ứng với hoạt động tương ứng trong số tập hợp các hoạt động. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, tiếp theo việc cài đặt ứng dụng bổ sung từ nền tảng phân phối số trên các thiết bị người dùng. Các lệnh còn được tạo cấu hình để nhận liên kết từ hệ điều hành của thiết bị người dùng. Liên kết này nhận dạng hoạt động thứ nhất. Các lệnh còn được tạo cấu hình

để tạo yêu cầu khôi phục thứ nhất dựa trên các tham số tương ứng với hoạt động thứ nhất. Các lệnh còn được tạo cấu hình để truyền yêu cầu khôi phục thứ nhất đến bộ xử lý của ứng dụng bổ sung.

Theo các dấu hiệu khác, các tham số tương ứng với hoạt động thứ nhất bao gồm kiểu hoạt động và tên hoạt động. Theo các dấu hiệu khác, yêu cầu khôi phục thứ nhất được truyền với cuộc gọi `continueuserActivity` đến ủy nhiệm app của ứng dụng bổ sung. Theo các dấu hiệu khác, phương pháp bao gồm bước phân tích ứng dụng thứ nhất để xác định tập hợp các đối tượng dữ liệu mà bộ xử lý trong ứng dụng thứ nhất được lập trình để truy cập đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ. Phương pháp này bao gồm xác định các tham số cho mỗi đối tượng dữ liệu trong số tập hợp các đối tượng dữ liệu. Phương pháp bao gồm bước tạo ra tập hợp liên kết thứ hai. Mỗi liên kết trong số tập hợp liên kết thứ hai tương ứng với đối tượng dữ liệu tương ứng của tập hợp các đối tượng dữ liệu. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, tiếp theo việc cài đặt ứng dụng bổ sung trong các thiết bị người dùng. Các lệnh còn được tạo cấu hình để nhận liên kết thứ hai từ hệ điều hành của các thiết bị người dùng. Liên kết thứ hai nhận dạng đối tượng dữ liệu thứ nhất. Các lệnh còn được tạo cấu hình để tạo yêu cầu khôi phục thứ hai dựa trên các tham số tương ứng với đối tượng dữ liệu thứ nhất. Các lệnh còn được tạo cấu hình để truyền yêu cầu khôi phục thứ hai đến bộ xử lý của ứng dụng bổ sung.

Theo các dấu hiệu khác, đối tượng dữ liệu thứ nhất là đối tượng `CSSearchableItem`. Theo các dấu hiệu khác, bằng cách sát nhập thư viện định tuyến, ứng dụng bổ sung được tạo cấu hình để đăng ký mẫu mã nhận dạng tài nguyên đồng nhất thứ nhất (uniform resource identifier - URI) với hệ điều hành nhờ việc cài đặt. Liên kết thích hợp với mẫu URI thứ nhất. Theo các dấu hiệu khác, phương pháp này bao gồm chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi các sự kiện giao diện người dùng (UI) được khởi động bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất. Phương pháp bao gồm lưu trữ chuỗi các sự kiện UI là, cấu trúc dữ liệu thứ nhất tương ứng với trạng thái thứ nhất. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, đáp lại người dùng của các thiết bị người dùng bắt gặp trạng thái thứ nhất trong lúc sử dụng ứng dụng bổ sung,

truyền đánh dấu đến hệ điều hành. Các lệnh còn được tạo cấu hình để, đáp lại nhận yêu cầu khôi phục chỉ báo trạng thái thứ nhất từ hệ điều hành, phát lại chuỗi của các sự kiện UI từ cấu trúc dữ liệu thứ nhất.

Theo các dấu hiệu khác, phương pháp bao gồm bước chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi các sự kiện giao diện người dùng (UI) được khởi động bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất tương ứng với đối tượng dữ liệu thứ nhất. Phương pháp bao gồm lưu trữ chuỗi các sự kiện UI làm cấu trúc dữ liệu thứ nhất tương ứng với đối tượng dữ liệu thứ nhất. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, đáp lại việc chạy ứng dụng bổ sung, truyền chọn lọc đối tượng dữ liệu thứ nhất đến hệ điều hành.

Các lệnh còn được tạo cấu hình để, đáp lại nhận yêu cầu khôi phục chỉ báo đối tượng dữ liệu thứ nhất từ hệ điều hành, phát lại chuỗi các sự kiện UI từ cấu trúc dữ liệu thứ nhất. Theo các dấu hiệu khác, liên kết bao gồm chọn lọc dữ liệu nối tiếp. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để giải mã các tham số tương ứng với hoạt động thứ nhất từ dữ liệu nối tiếp trong liên kết. Theo các dấu hiệu khác, liên kết này bao gồm chọn lọc mã nhận dạng duy nhất. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để truy hồi các tham số tương ứng với hoạt động thứ nhất dựa trên mã nhận dạng duy nhất.

Theo các dấu hiệu khác, phương pháp bao gồm bước lưu trữ, trong kho dữ liệu, các tham số tương ứng với tập hợp các hoạt động được xác định. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, nhờ lần chạy ứng dụng bổ sung thứ nhất, tải xuống các tham số tương ứng với ít nhất hoạt động thứ nhất từ kho dữ liệu để lưu trữ trong khu vực kho dữ liệu đến thư viện định tuyến. Thư viện định tuyến bao gồm các lệnh được tạo cấu hình để truy hồi các tham số tương ứng với hoạt động thứ nhất từ kho dữ liệu đáp lại mã nhận dạng duy nhất.

Các lĩnh vực khác áp dụng sáng chế sẽ trở nên rõ ràng khác từ mô tả chi tiết, các yêu cầu bảo hộ, và các hình vẽ. Mô tả chi tiết và các ví dụ cụ thể chỉ nhằm mục đích minh họa và không nhằm làm giới hạn phạm vi bảo hộ của sáng chế.

Mô tả vắn tắt các hình vẽ

Các hình vẽ được mô tả ở đây chỉ nhằm để minh họa của các phương án được chọn và không phải tất cả các phương án khả thi và không nhằm làm giới hạn phạm vi bảo hộ của sáng chế.

Fig.1 là minh họa bằng biểu đồ và sơ đồ khối chức năng của app làm ví dụ tương tác với chức năng lập chỉ mục app của hệ điều hành.

Fig.2 là sơ đồ khối chức năng cấp độ cao của môi trường ứng dụng di động theo các nguyên lý của sáng chế.

Fig.3A là sơ đồ khối chức năng theo việc thực thi làm ví dụ của sự tích hợp của thư viện định tuyến với app làm ví dụ.

Fig.3B là sơ đồ khối chức năng của sự tích hợp làm ví dụ của thư viện định tuyến với app làm ví dụ khác.

Fig.4 là sơ đồ khối chức năng theo việc thực thi làm ví dụ của thư viện định tuyến.

Fig.5 là lưu đồ của hoạt động chung làm ví dụ của hệ sinh thái ứng dụng di động theo các nguyên lý của sáng chế.

Fig.6 là lưu đồ về hoạt động làm ví dụ của thư viện định tuyến.

Fig.7 là lưu đồ về thao tác làm ví dụ của hoạt động và việc xử lý liên kết trong thư viện định tuyến.

Fig.8 là lưu đồ của chỗ giao nhau làm ví dụ của tập hợp liên kết (breadcrumb) để đạt trạng thái được xác định.

Fig.9 là sơ đồ khối chức năng theo việc thực thi làm ví dụ của hệ thống tìm kiếm trên Fig.2.

Fig.10 là lưu đồ về hoạt động làm ví dụ về việc tạo và cung cấp liên kết ngầm cho các thiết bị người dùng theo các nguyên lý của sáng chế.

Fig.11 là sơ đồ khối chức năng theo việc thực thi làm ví dụ của hệ thống phân tích ngoại tuyến.

Fig.12 là lưu đồ về hoạt động làm ví dụ của việc tạo tập hợp liên kết (breadcrumb) đối với app là một phần của phân tích ngoại tuyến.

Fig.13 là lưu đồ về phép ngoại suy là một phần của phân tích ngoại tuyến để nhận dạng các trạng thái bổ sung liên quan.

Các số tham chiếu tương ứng chỉ báo các phần tương ứng thông qua nhiều hình chiếu trong các hình vẽ.

Mô tả chi tiết sáng chế

Giới thiệu

Nhà phát triển của app có thể viết mã để tận dụng chức năng của hệ điều hành (OS) cho phép, theo một ví dụ, hoạt động trong app được đánh dấu cho người dùng để trở lại sau đó hoặc đối với hoạt động được khởi động lại trên thiết bị khác trong số các thiết bị của người dùng. Nhà phát triển cũng có thể viết mã để đưa dữ liệu vào OS để lập chỉ mục để OS có thể phủ dữ liệu, như đáp lại tìm kiếm OS mở rộng (OS-wide).

Để thực hiện chức năng này, nhà phát triển viết mã để cho phép app khôi phục để hoạt động được chỉ báo bởi OS hoặc trạng thái tương tác với đối tượng dữ liệu được xác định bởi OS. Theo một số phương án thực hiện, đối tượng hoạt động được nhận từ OS xác định đầy đủ hoạt động cho app để khôi phục, ngay cả khi việc sao chép cụ thể này của app không được thực hiện hoạt động trước.

Trong lúc đó, đối với các đối tượng dữ liệu, một số app có thể cho phép thể hiện trạng thái của app tương ứng với đối tượng dữ liệu nhất định chỉ khi nào mà đối tượng dữ liệu đó được lưu trữ với bản sao hiện tại của app. Theo các phương án thực hiện khác, hoặc đối với các app nhất định, các đối tượng dữ liệu nhất định có thể xác định đầy đủ trạng thái mà app cần trở lại để xem đối tượng dữ liệu đó. Điều này thường có nghĩa là dữ liệu đối với đối tượng dữ liệu được biết bởi app, truy cập được vào app, hoặc được đề xuất cho app từ OS trong yêu cầu hoạt động người dùng tiếp tục. Đối với các đối tượng dữ liệu không xác định đầy đủ trạng thái của app, các đối tượng dữ liệu này có thể dành riêng cho app trong đó chúng được tạo ra, và không thể được truy cập với các bản sao khác của app.

Cho dù nhà phát triển có thể triển khai, một vài hoặc tất cả trạng thái và cho một vài hoặc tất cả các đối tượng dữ liệu, khả năng để khôi phục hoạt động hoặc điều hướng cho đối tượng dữ liệu đó, không cung cấp cho các bên thứ ba, như các dịch vụ tìm kiếm hoặc các app khác, để liên kết trực tiếp đến các hoạt động hoặc các đối tượng dữ liệu của chúng trong app. Sáng chế nhận dạng các hoạt động được thực hiện bởi nhà phát triển và cung cấp thư viện, khi được thể hiện với liên kết

ngầm thích hợp, có thể dẫn đến việc xử lý hoạt động bên trong của app. Điều này cho phép nỗ lực của nhà phát triển trong tương tác với OS được tác dụng đòn bẩy để cung cấp liên kết ngầm truy cập vào các bên thứ ba.

Đối với một số app, nhà phát triển không thể khai triển chức năng tìm kiếm OS đối với một số trạng thái và một số dữ liệu. Đối với một số app, nhà phát triển không thể khai triển chức năng đó đối với bất kỳ trạng thái nào hoặc bất kỳ dữ liệu nào. Sáng chế, do đó, mô tả cách nhà phát triển có thể xác định các trạng thái nào của app cần có chức năng này được cho phép. Sáng chế mô tả cách mà ứng dụng có thể được bổ sung để cho phép chức năng này của app, như bằng cách ghi các chuỗi của các sự kiện giao diện người dùng có thể được phát lại để đạt các trạng thái ngầm của app.

Khi thư viện định tuyến theo các nguyên lý của sáng chế được sử dụng để cho phép đánh dấu hoạt động và lập chỉ mục dữ liệu, cấu hình của thư viện định tuyến có thể được xác định bởi nhà phát triển. Trong các phương án thực hiện khác nhau, nhân viên của nhà phát triển có thể tạo các quyết định cấu hình, như các trạng thái nào để đánh dấu và dữ liệu nào để lập chỉ mục, và các quyết định cấu hình đó được chuyển tiếp đến thư viện định tuyến. Theo các phương án thực hiện khác nhau, thư viện định tuyến có thể cập nhật theo chu kỳ các cấu hình này để nhà phát triển có thể thay đổi cấu hình thậm chí cho các app được cài đặt sẵn trên các thiết bị.

Thậm chí công nhà phát triển có thể cho phép người không phải lập trình viên làm việc với nhà phát triển để xác định các quy tắc công việc. Ví dụ, chuyên gia về marketing hoặc người dùng có kinh nghiệm làm việc với nhà phát triển có thể xác định xem dữ liệu nào nhập vào OS để lập chỉ mục để tối ưu hóa sự liên quan tìm kiếm của app đối với OS đã cho.

Công nhà phát triển theo sáng chế đề xuất thư viện định tuyến để nhà phát triển để tích hợp vào trong app, và thư viện định tuyến triển khai chức năng trên. Ví dụ, dựa trên dữ liệu cấu hình, thư viện định tuyến có thể đánh dấu các hoạt động nhất định với OS và có thể đưa các đối tượng dữ liệu nhất định để lập chỉ mục cho OS. Dữ liệu cấu hình có thể được tạo ra ở thời điểm cài vào trong app, và có thể được tạo ra thủ công, với trí tuệ nhân tạo, hoặc với trí tuệ nhân tạo được bổ sung bởi người vận hành.

Ngoài ra, thư viện định tuyến có thể phục vụ các yêu cầu hoạt động tiếp tục từ OS, như bằng cách chạy chuỗi các sự kiện giao diện người dùng để để trở lại hoạt động chọn lọc hoặc để hiển thị dữ liệu được chọn. Thư viện định tuyến cũng có thể xử lý các liên kết đến, như từ các app khác hoặc từ các phiên bản trình duyệt.

Thư viện định tuyến có thể có dạng thư viện liên kết động (DLL), dylib, hoặc kit phát triển phần mềm (software development kit - SDK). Thư viện định tuyến có thể được tải xuống bởi nhà phát triển và được nhập vào môi trường phát triển được tích hợp (IDE) của nhà phát triển, như Xcode từ Apple Inc. Theo một số phương án thực hiện, thư viện định tuyến được biên dịch sẵn hoặc bao gồm tạo các tệp tin và các tệp tin cấu hình khác cho phép thư viện định tuyến được biên dịch.

Nhà phát triển có thể tích hợp thư viện định tuyến đơn giản bằng cách nhập IDE để thêm thư viện định tuyến vào dự án app của chúng, như bằng cách thêm thư viện định tuyến vào danh sách nhập hoặc vào danh sách các môđun mã được chứa. Nhà phát triển cũng có thể cập nhật tệp tin cấu hình để chỉ báo là thư viện định tuyến sẽ xử lý các mã nhận dạng tài nguyên đồng nhất (URI) hoặc kiểu hoạt động nhất định. Tệp tin cấu hình có thể có dạng của tệp danh sách tính chất thông tin (Info.plist). Theo các phương án thực hiện khác nhau, tập lệnh có thể được đề xuất cùng với thư viện định tuyến thực hiện cập nhật tệp tin cấu hình cũng như việc nhập thư viện định tuyến vào dự án IDE hiện tại.

Trong lúc đó, công nhà phát triển cung cấp app cho hệ thống phân tích ngoại tuyến xác định cách để truy cập các trạng thái quan tâm. Các trạng thái quan tâm này có thể được xác định bởi nhà phát triển app hoặc được xác định bởi công nhà phát triển. Tóm lại, hệ thống phân tích ngoại tuyến của công nhà phát triển xác định chuỗi các thao tác của giao diện người dùng (UI) nào mà người dùng tiếp cận trạng thái quan tâm. Chuỗi thao tác UI có thể được phát lại bởi thư viện định tuyến trong app trên thiết bị của người dùng để đến trạng thái quan tâm. Theo cách khác, thư viện định tuyến mô phỏng các thao tác UI mà người dùng tự họ thực hiện để đạt trạng thái ngầm.

Bằng cách mô phỏng các thao tác UI, thư viện định tuyến ngăn chặn người dùng thực hiện mỗi thao tác UI và thường sẽ nhanh hơn nhiều so với khi người dùng thực hiện thủ công chuỗi các thao tác UI. Người dùng thậm chí không thể xem

một vài trong số các trạng thái trung gian khi thư viện định tuyến điều hướng đến trạng thái ngầm quan tâm, hoặc các trạng thái trung gian này chỉ có thể hiển thị ngắn gọn.

Các chuỗi thao tác UI đối với các trạng thái ngầm của app có thể được tích hợp vào app cùng với thư viện định tuyến, hoặc có thể được tải xuống bởi thư viện định tuyến ngay khi app này được cài đặt. Theo một số phương án thực hiện, chuỗi thao tác UI có thể được mã hóa là một phần của liên kết chỉ định app và được phục vụ bởi thư viện định tuyến.

Không có sự nỗ lực thêm từ nhà phát triển, giờ đây app có chức năng liên kết ngầm có thể được truy cập từ các nguồn bên ngoài. Ví dụ, hệ thống tìm kiếm có thể cung cấp các kết quả để dẫn trực tiếp đến các trạng thái ngầm của app. Hoặc app bên thứ ba có thể dẫn người dùng dễ dàng vào các trạng thái ngầm của app. Điều này có thể làm tăng độ nhìn thấy của app và cho phép sự tích hợp chặt hơn và sự trải nghiệm của người dùng tốt hơn với app này.

Ngoài ra, giờ đây công nhà phát triển chịu trách nhiệm về liên kết ngầm, các liên kết ngầm có thể được thêm hoặc sửa đổi không yêu cầu hỗ trợ từ các lập trình viên phần mềm của nhà phát triển, có thể làm việc ở các dự án khác và có sự ưu tiên khác. Theo cách khác, người làm kinh doanh, như chuyên gia quảng cáo hoặc marketing làm việc với nhà phát triển của app, có thể sử dụng công nhà phát triển để nhận dạng các trạng thái nào của app có thể liên kết ngầm. Người làm kinh doanh thậm chí có thể loại bỏ các trạng thái khỏi danh sách các trạng thái liên kết ngầm, tất cả không yêu cầu sự hỗ trợ của nhà phát triển phần mềm.

Sơ đồ khối

Trên Fig.2, các chi tiết bổ sung được thể hiện. Như được lưu ý ở trên, nhà phát triển 204 cung cấp app (được gọi là “App A”) cho công nhà phát triển 208. Công nhà phát triển 208 đưa ra bản sao của thư viện định tuyến để nhà phát triển 204 để sát nhập vào App A, tạo phiên bản bổ sung của App A.

Nhà phát triển 204 cung cấp App A bổ sung, sau đây vẫn được gọi là App A cho đơn giản, nền tảng phân phối số 212 để phân phối để giới hạn người dùng. Nền tảng phân phối số 212 cung cấp các app riêng cho các thiết bị người dùng và có thể dành riêng cho OS. Các nền tảng phân phối số làm ví dụ bao gồm nền tảng phân

phối số GOOGLE PLAY bởi Google, Inc., trên tảng phân phối số APP STORE bởi Apple, Inc., và nền tảng phân phối số WINDOWS PHONE bởi Microsoft Corp. Nếu nhà phát triển 204 được cung cấp sẵn App A cho nền tảng phân phối số 212 trước khi bổ sung với thư viện định tuyến, nhà phát triển 204 có thể phát hành App A bổ sung làm phiên bản mới của App A.

Người dùng của thiết bị người dùng 220 cài đặt App A (bổ sung) từ nền tảng phân phối số 212. Trong thiết bị người dùng 220, một hoặc nhiều bộ xử lý (có thể gồm các bộ xử lý đa dụng cũng như các bộ đồng xử lý bổ sung, như các bộ xử lý cảm biến vật lý và các bộ xử lý đồ họa) chạy các lệnh từ bộ nhớ.

Một vài trong số các lệnh này tương ứng với hệ điều hành 224, trình duyệt 228, bản sao được cài đặt của App A (được gọi là App A 232), và app bên thứ ba (App B 236). Hệ điều hành 224 bao gồm đăng ký app được cài đặt 240 và bộ định tuyến liên kết 244. Bộ định tuyến liên kết 244 nhận các liên kết, như các URL (uniform resource locator - địa chỉ tài nguyên đồng nhất) và các URI (các mã nhận dạng tài nguyên đồng nhất), và xác định cách để xử lý các liên kết của chúng.

Nói chung, bộ định tuyến liên kết 244 xử lý liên kết bằng cách chuyển tiếp nó đến bộ nhận được đăng ký. Bộ định tuyến liên kết 244 kiểm tra đăng ký app được cài đặt 240 để xác định xem liệu app có yêu cầu hệ thống, miền cụ thể, hoặc bộ lọc khác khớp với liên kết này hay không. Nếu không, liên kết này có thể được truyền đến trình duyệt 228, có thể đăng ký tập hợp sơ đồ đối với các giao thức như HTTP (Hypertext transfer protocol – giao thức chuyển tải siêu văn bản), HTTPS (HTTP secure – HTTP an toàn), FTP (file transfer protocol – giao thức chuyển tệp tin), Telnet, và Gopher.

Theo một số phương án thực hiện, khi liên kết xác định địa chỉ máy chủ web, bộ định tuyến liên kết 244 có thể liên lạc máy chủ web để có danh sách các app được ủy quyền. Theo cách này, bộ định tuyến liên kết 244 xác minh app được đăng ký miền nhất định được ủy quyền bởi miền đó để xử lý các liên kết. Điều này ngăn chặn các app khỏi trưng dụng các liên kết mà chúng không thích hợp. Nếu app được đăng ký là miền được ủy quyền bởi máy chủ web của miền đó, bộ định tuyến liên kết 244 có thể cung cấp liên kết này đến app được đăng ký.

Thư viện định tuyến 260, được nhận từ cổng nhà phát triển 208, khiến App A 232 đăng ký sơ đồ hoặc miền cụ thể với đăng ký app được cài đặt 240. Sơ đồ này có thể dựa trên mã nhận dạng của cổng nhà phát triển 208 cũng như mã nhận dạng của App A. Ví dụ, chuỗi kiểm tra được liên kết với cổng nhà phát triển 208 (như “portal”) có thể được nối với chuỗi văn bản tương ứng với App A (như “appa”). Theo ví dụ cụ thể, sơ đồ được đăng ký bởi App A 232 có thể là “portal-appa://”. Trong một số ví dụ thực hiện khác, chuỗi văn bản được liên kết với cổng nhà phát triển 208 không thể đọc được bởi người.

Khi bộ định tuyến liên kết 244 nhận liên kết, như từ trình duyệt 228, trong đó sơ đồ khớp với “portal-appa://”, bộ định tuyến liên kết 244 chuyển tiếp liên kết đó đến App A 232, trong đó liên kết được nhận và được xử lý bởi thư viện định tuyến 260. Thư viện định tuyến 260 phân tách liên kết và điều hướng đến trạng thái ngầm được chỉ báo bởi liên kết.

Chỉ với mục đích minh họa, trên Fig.2, Hình A 268-1, Hình B 268-2, và Hình C 268-3 (được gọi chung là, các hình 268) được thể hiện. Trong nhiều app, nhiều hơn ba hình sẽ được biểu diễn. Trong các phương án thực hiện khác nhau, nhiều hoặc tất cả hình có thể được điều khiển bởi một bộ điều khiển hình. Trong các phương án thực hiện khác, mỗi hình có thể được điều khiển bởi bộ điều khiển hình tách biệt.

Chỉ với mục đích minh họa, thư viện định tuyến 260 được thể hiện các sự kiện mô phỏng từ chuỗi sự kiện UI định trước được nhận dạng với liên kết làm ví dụ. Liên kết này được nhận bởi thư viện định tuyến 260 và tương ứng với trạng thái ngầm, cụ thể Hình C 268-3. Chuỗi sự kiện UI bao gồm sự kiện UI thứ nhất, khiến App A 232 chuyển tiếp từ Hình A 268-1 đến Hình B 268-2, sự kiện UI thứ hai, trong đó App A giữ lại trong Hình B 268-2, và sau đó sự kiện UI thứ ba, khiến App A 232 chuyển tiếp từ Hình B 268-2 đến Hình C 268-3. Thư viện định tuyến 260 sẽ mô phỏng các sự kiện UI này theo chuỗi.

Hệ thống tìm kiếm 272 của hệ điều hành 224 nhận dữ liệu và các đối tượng hoạt động từ thư viện định tuyến 260. Thư viện định tuyến 260 đề xuất các đối tượng dữ liệu để lập chỉ mục và các đối tượng hoạt động để đánh dấu. Hệ thống tìm kiếm 272 cho phép người dùng tìm kiếm dữ liệu và/hoặc các hoạt động từ App A

232. Khi người dùng chọn hoạt động tương ứng với App A 232, hệ thống tìm kiếm 272 gửi chỉ báo của hoạt động được chọn đến thư viện định tuyến 260. Hệ thống tìm kiếm 272 được thể hiện chi tiết hơn trên Fig.3A và Fig.3B.

Song song với việc cung cấp thư viện định tuyến từ cổng nhà phát triển 208 đến nhà phát triển 204, cổng nhà phát triển 208 đề xuất App A cho hệ thống phân tích ngoại tuyến 280. Bản sao của App A được cung cấp cho cổng nhà phát triển 208 có thể là thiết kế phiên bản chuẩn, như gói iOS App Store (tệp ipa).

Theo các phương án thực hiện khác, bản sao của App A được đề xuất cho cổng nhà phát triển 208 có thể là thiết kế đặc biệt cho phép App A chạy trong bộ mô phỏng, và có thể bao gồm các ký hiệu, debugging info, và/hoặc mã nguồn thông thường không có sẵn trong app được phân phối bởi nền tảng phân phối số 212. Điều này có thể cho phép hệ thống phân tích ngoại tuyến 280 để phân tích hiệu quả hoặc chính xác hơn App A. Ví dụ, bản sao của App A có thể là thiết kế chỉnh lý được thiết kế để chạy trong ứng dụng bộ mô phỏng từ Apple, Inc. trên hệ điều hành OS X. Ngược lại, phiên bản của App A được phân phối bởi nền tảng phân phối số 212 là thiết kế phiên bản chuẩn.

Hệ thống phân tích ngoại tuyến 280, như được mô tả chi tiết hơn sau đây, xác định các chuỗi sự kiện UI có thể được sử dụng bởi thư viện định tuyến 260 để đạt được các hình cụ thể của App A 232. Chuỗi sự kiện UI có thể được gọi là đường sự kiện giao diện người dùng bao gồm chuỗi theo trình tự của vết tập hợp liên kết riêng biệt (tức là, các sự kiện UI). Để đơn giản, trong mô tả còn lại của phần bộc lộ này, thuật ngữ tập hợp liên kết (breadcrumb) sẽ được sử dụng để đề cập đến vết tập hợp liên kết.

Hệ thống phân tích ngoại tuyến 280 có thể cung cấp các tập hợp liên kết được xác định cho máy chủ dữ liệu 284. Máy chủ dữ liệu 284 bao gồm kho dữ liệu, như cơ sở dữ liệu liên quan, lưu trữ các tập hợp liên kết đối với mỗi app được xử lý bởi hệ thống phân tích ngoại tuyến 280. Mỗi trạng thái ngấm của app được liên kết hợp với tập hợp liên kết tương ứng. Theo một số phương án thực hiện, máy chủ dữ liệu 284 có thể được triển khai làm dịch vụ lưu trữ khối dựa trên đám mây, như dịch vụ S3 (Simple Storage Service - dịch vụ lưu trữ đơn giản) có sẵn từ các dịch vụ web amazon.

Các cơ chế truy cập xác định cách mà các trạng thái ngầm có thể đạt được đối với các mục đích này là các quảng cáo hiển thị và các kết quả tìm kiếm ngầm. Ví dụ, kết quả tìm kiếm có thể tương ứng với trạng thái ngầm cụ thể của app. Kết quả tìm kiếm có thể là thẻ hình ngầm (DVC - Deep View Card, xem dưới đây) với liên kết bao gồm hoặc chỉ báo tập hợp liên kết tương ứng với trạng thái ngầm đó. Các cơ chế truy cập tiềm năng khác cho kết quả tìm kiếm có thể bao gồm liên kết ngầm riêng được chuẩn bị bởi nhà phát triển của chính app đó, hoặc URL chuẩn hướng đến hiệu chỉnh web của app.

Một trong số các cơ chế truy cập có sẵn có thể được chọn bởi hệ thống tìm kiếm và được đề xuất cùng với kết quả tìm kiếm cho máy khách tìm kiếm. Theo một số phương án thực hiện khác, nhiều cơ chế truy cập có thể được đề xuất trong kết quả tìm kiếm, và máy khách tìm kiếm xác định xem cơ chế truy cập nào để sử dụng đáp lại lựa chọn người dùng của kết quả tìm kiếm. Ví dụ, cơ chế truy cập dựa trên tập hợp liên kết có thể được chọn trên cơ chế truy cập dựa trên web khi các quy tắc nghiệp vụ chỉ báo sự ưu tiên của kết quả để đưa ra trong các app riêng thay cho trong các hiệu chỉnh web của các app của chúng.

Cơ chế truy cập dựa trên tập hợp liên kết, tuy nhiên, chỉ có thể có sẵn khi app này được cài đặt sẵn. Theo một số phương án thực hiện, máy khách tìm kiếm có thể kết hợp, hoặc nhận là một phần của kết quả tìm kiếm, mã mà các tập lệnh tải xuống và cài đặt app, được theo sau bởi sự điều khiển của liên kết ngầm. Liên kết ngầm này có thể được truyền động bằng cách gửi URI dựa trên tập hợp liên kết đến app được cài đặt mới.

Máy chủ dữ liệu 284 cũng có thể lưu trữ các cài đặt cấu hình, như các trạng thái ngầm để cho phép các liên kết, các trạng thái ngầm để đánh dấu làm các hoạt động, và dữ liệu nào để lập chỉ mục. Các cấu hình này có thể được xác định bởi nhà phát triển 204 sử dụng cổng nhà phát triển 208. Máy chủ dữ liệu 284 cũng có thể nhận các mục chỉ số từ thư viện định tuyến 260. Ví dụ, thư viện định tuyến 260 có thể báo cáo các đối tượng hoạt động trở lại máy chủ dữ liệu 284.

Sau đó máy chủ dữ liệu 284 có thể phân tích tương tác người dùng với App A 232 và tạo ra thông tin về cách để đưa ra các kết quả tìm kiếm liên quan hơn. Ngoài ra, dữ liệu từ máy chủ dữ liệu 284 có thể được cung cấp ngược lại cho nhà

phát triển 204, chủ yếu đưa ra cách dễ dàng để hướng dẫn App A 232 phân tích sự gắn kết người dùng. Máy chủ dữ liệu 284 có thể liên quan tới hoạt động người dùng qua nhiều app trên các thiết bị người dùng 220, do đó cải thiện thêm các kết quả tìm kiếm.

Tích hợp thư viện định tuyến với ứng dụng

Trên Fig.3A, ví dụ về App A 232 bao gồm thư viện định tuyến 260 và cũng bao gồm bộ xử lý App A 300, được tạo ra bởi nhà phát triển của App A 232. Như đã xem trong ví dụ trên Fig.3A, bộ xử lý App A 300 cho phép tiếp tục các hoạt động người dùng liên quan đến Hình B 268-2, Hình D-268-4, và Hình E-268-5. Bộ xử lý App A 300 phối hợp với bộ tạo lập hoạt động 304, cập nhật hệ thống tìm kiếm 272 với các đối tượng hoạt động chỉ báo các hoạt động người dùng thực hiện với App A 232.

Theo các phương án thực hiện khác nhau, hệ thống tìm kiếm 272 có thể bao gồm các phần tử tương tự với các phần tử được thể hiện trong hệ điều hành 100 trên Fig.1. Bộ tạo lập hoạt động 304 có thể gửi các đối tượng hoạt động đến hệ thống tìm kiếm 272 dựa trên các hoạt động người dùng được kết hợp với Hình B 268-2, Hình D 268-4, và Hình E 268-5. Bộ xử lý App A 300 cũng khớp với bộ lập chỉ mục dữ liệu 308, có thể cung cấp các đối tượng dữ liệu cho hệ thống tìm kiếm 272 liên quan đến Hình B 268-2, Hình D 268-4, và Hình E 268-5.

Thư viện định tuyến 260 có thể dẫn ra Hình A 268-1 và Hình C 268-3 nếu nhà phát triển của App A 232 yêu cầu chức năng đó thông qua cổng nhà phát triển. Thư viện định tuyến 260 có thể cho phép truy cập Hình A 268-1 và Hình C 268-3 bằng cách dẫn các bộ điều khiển hình tương ứng, như được mô tả chi tiết hơn trong đơn sáng chế Mỹ số 15/235,650, nộp ngày 12/08/2016, có tên là “Monitoring và Actuation of Controller Parameter to Reach Deep Link Without Manual Intervention”, với tác giả sáng chế thứ nhất là Kalyan Desineni, và có số hồ sơ 17100-000029-US. Toàn bộ phần bộc lộ của đơn này được đưa vào bằng cách viện dẫn.

Theo các phương án thực hiện khác, thư viện định tuyến 260 có thể truy cập Hình A 268-1 và Hình C 268-3 bằng cách phát lại các chuỗi của các sự kiện giao diện người dùng được xác định trong quy trình ngoại tuyến. Để thêm thông tin, xem

Đơn sáng chế Mỹ số 15/235,859, nộp ngày 12/08/2016, có tiêu đề là “Deep Linking to Mobile Application States Through Programmatic Replay of User Interface Events”, với tác giả sáng chế thứ nhất là Kalyan Desineni, và có số hồ sơ là 17100-000032-US. Toàn bộ phần bộc lộ của đơn này được đưa vào bằng cách viện dẫn.

Trong khi bộ tạo lập hoạt động 304 được phát triển bởi nhà phát triển của App A 232 đề xuất các đối tượng hoạt động cho một vài trong số các hình 268 đến bộ theo dõi hoạt động 124, thư viện định tuyến 260 có thể đề xuất các đối tượng hoạt động từ các hình còn lại trong số các hình 268. Ví dụ, bộ tạo lập hoạt động 304 có thể tạo đối tượng `NSNuserActivity` và dẫn ra phương pháp `becomeCurrent()` để đưa ra đối tượng hoạt động cho bộ theo dõi hoạt động 124. Nó có thể không liên quan đến bộ theo dõi hoạt động 124 liệu thư viện định tuyến hoặc bộ xử lý App A 300 có được gửi đối tượng hoạt động hay không.

Khi bộ theo dõi hoạt động 124 gửi tín hiệu hoạt động người dùng tiếp tục đến App A 232, thư viện định tuyến 260 thu (như do quá tải, hủy bỏ, hoặc thay đổi phương pháp hoạt động người dùng tiếp tục) tín hiệu hoạt động người dùng tiếp tục. Nếu tín hiệu hoạt động người dùng tiếp tục liên quan đến bộ xử lý App A 300, tín hiệu hoạt động người dùng tiếp tục được truyền qua đến bộ xử lý App A 300. Theo cách khác, thư viện định tuyến 260 xử lý tín hiệu hoạt động người dùng tiếp tục.

Tương tự, thư viện định tuyến 260 và bộ lập chỉ mục dữ liệu 308 được phát triển bởi nhà phát triển của App A 232 mỗi trong số đó có thể tạo các đối tượng `CSSearchableItem` và đưa chúng vào chỉ mục tìm kiếm 120, như bằng cách sử dụng phương pháp `indexSearchableItems`. Khi bộ theo dõi hoạt động 124 cung cấp tín hiệu hoạt động người dùng tiếp tục cho thư viện định tuyến 260 liên quan đến đối tượng dữ liệu, thư viện định tuyến 260 tạo đối tượng cho Hình dựa trên đối tượng dữ liệu được chọn hoặc truyền tín hiệu hoạt động người dùng tiếp tục qua bộ xử lý App A 300.

Đối tượng `NSNuserActivity` có thể bao gồm mã nhận dạng duy nhất (như tên) và kiểu hoạt động. Trong tín hiệu hoạt động người dùng tiếp tục, kiểu hoạt động có thể phân biệt đối tượng `NSNuserActivity` với đối tượng `CSSearchableItem`. Đối tượng `NSNuserActivity` cũng có thể bao gồm danh sách các từ khóa văn bản để cho phép đối tượng này được lập chỉ mục. Đối tượng `NSNuserActivity` cũng có

thể bao gồm tập hợp siêu dữ liệu, như `CSSearchableItemAttributeSet`, có thể bao gồm hình ảnh và thông tin khác. Tập hợp siêu dữ liệu của đối tượng có thể được hiển thị khi đối tượng được bao gồm trong tập hợp các kết quả tìm kiếm và cho thông tin người dùng liệu đối tượng có thỏa mãn câu hỏi của họ hay không.

Đối tượng `CSSearchableItem` bao gồm mã nhận dạng duy nhất và tập hợp siêu dữ liệu, có thể có dạng `CSSearchableItemAttributeSet`. Đối tượng `CSSearchableItem` cũng có thể bao gồm ngày hết hạn vượt ra ngoài việc đối tượng sẽ không được lập chỉ mục tự động bởi hệ điều hành. Ngoài ra, đối tượng `CSSearchableItem` có thể bao gồm mã nhận dạng miền đến các đối tượng liên quan đến nhóm.

Theo các phương án thực hiện khác nhau, cả thư viện định tuyến 260 và bộ xử lý App A 300 có thể tạo đối tượng một hoặc nhiều trong số các hình 268. Ví dụ, bộ xử lý App A 300 có thể được lập trình để hướng dẫn bộ tạo lập hoạt động 304 để đánh dấu các hoạt động tương ứng với Hình B 268-2. Trong lúc đó, thư viện định tuyến 260 có thể được lập trình để nhập dữ liệu để lập chỉ mục từ Hình B 268-2.

Khi tín hiệu hoạt động người dùng tiếp tục được nhận bởi thư viện định tuyến 260, nếu tín hiệu hoạt động người dùng tiếp tục liên quan đến hoạt động đánh dấu, thư viện định tuyến 260 truyền qua tín hiệu hoạt động người dùng tiếp tục đến bộ xử lý App A 300. Trong lúc đó, nếu tín hiệu hoạt động người dùng tiếp tục đối với Hình B 268-2 liên quan đến đối tượng dữ liệu, chính thư viện định tuyến 260 điều hướng cho Hình B 268-2 để hiển thị dữ liệu liên quan đến đối tượng dữ liệu.

Mặc dù không được thể hiện trên Fig.3A, App A 232 có thể bao gồm bộ định tuyến liên kết bên trong nhận các liên kết ngầm và các hình tạo đối tượng dựa trên các liên kết ngầm đó. Đối với các liên kết bao gồm sơ đồ được đăng ký bởi thư viện định tuyến 260, khác với sơ đồ được đăng ký bằng bộ định tuyến liên kết bên trong của App A 232, thư viện định tuyến 260 nhận liên kết ngầm và tạo đối tượng một trong số các hình 268 dựa trên liên kết này. Ví dụ, liên kết có thể xác định tập hợp liên kết để theo dõi để đến trạng thái mong muốn.

Thư viện định tuyến 260 có thể hủy bỏ các phương pháp được dẫn ra bởi bộ tạo lập hoạt động 304 và bộ lập chỉ mục dữ liệu 308 để truyền các đối tượng hoạt động và các đối tượng dữ liệu đến hệ thống tìm kiếm 272. Theo cách này, thư viện

định tuyến 260 trước hết có thể kiểm tra các đối tượng hoạt động và sau đó các đối tượng dữ liệu trước khi truyền các đối tượng hoạt động các đối tượng dữ liệu đến hệ thống tìm kiếm 272. Sau đó thư viện định tuyến 260 sẽ nhận biết là các đối tượng hoạt động nào được tạo ra bởi bộ tạo lập hoạt động 304, và có thể, do đó, phục vụ các liên kết ngầm liên quan đến các đối tượng hoạt động của chúng. Tương tự, thư viện định tuyến 260 sẽ biết được các đối tượng dữ liệu nào được lập chỉ mục bởi bộ lập chỉ mục dữ liệu 308, và sau đó có thể phục vụ các liên kết ngầm liên quan đến các đối tượng dữ liệu của chúng.

Để cho phép các app bên thứ ba và phục vụ để truy cập App A 232 sử dụng các liên kết ngầm đó, thư viện định tuyến 260 có thể tải lên thông tin từ các đối tượng hoạt động và các đối tượng dữ liệu đến máy chủ từ xa. Để phục vụ liên kết ngầm liên quan đến các đối tượng hoạt động từ bộ tạo lập hoạt động 304 và các đối tượng dữ liệu từ bộ lập chỉ mục dữ liệu 308, thư viện định tuyến 260 có thể tạo ra tín hiệu hoạt động người dùng tiếp tục và gửi tín hiệu đó đến bộ xử lý App A 300. Bộ xử lý App A 300 có thể không nhận biết là tín hiệu hoạt động người dùng tiếp tục được nhận từ thư viện định tuyến 260 thay cho từ hệ thống tìm kiếm 272. Sau đó, bộ xử lý App A 300 có thể khôi phục hoạt động hoặc tạo đối tượng đánh giá liên quan đến dữ liệu chỉ mục nếu yêu cầu này được đến trực tiếp từ hệ thống tìm kiếm 272.

Trên Fig.3B, app làm ví dụ khác, App B 360, được thể hiện. Trong App B 360, các hình 364-1, 364-2, 364-3, 364-4, và 364-5 (được gọi chung là, các hình 364) được thể hiện để minh họa. App B 360 không bao gồm bộ xử lý cài sẵn tương tự với bộ xử lý App A 300 trên Fig.3A. Do đó, để truy cập hình bất kỳ trong số các hình 364, thư viện định tuyến 260 bao gồm các cơ chế (như các tập hợp liên kết) để truy cập mỗi trong số các hình 364.

Ngoài ra, thư viện định tuyến 260 chỉ chịu trách nhiệm về việc tạo các đối tượng hoạt động và các đối tượng dữ liệu để gửi đến hệ thống tìm kiếm 272. Khi tín hiệu hoạt động người dùng tiếp tục được nhận bởi thư viện định tuyến 260, thư viện định tuyến 260 không thể truyền tín hiệu hoạt động người dùng tiếp tục thông qua và thay vì chịu trách nhiệm về việc xử lý khôi phục hoạt động hoặc việc tạo hình liên quan đến đối tượng dữ liệu.

Thư viện định tuyến

Trên Fig.4, việc thực thi làm ví dụ của thư viện định tuyến 260 bao gồm dữ liệu đăng ký 404. Dữ liệu đăng ký 404 chỉ báo cho hệ điều hành thư viện định tuyến 260 sẽ xử lý các liên kết có sơ đồ liên quan đến thư viện định tuyến 260 và cũng sẽ xử lý các tín hiệu hoạt động người dùng tiếp tục liên quan đến app trong đó thư viện định tuyến 260 được sát nhập. Theo các phương án thực hiện khác nhau, dữ liệu đăng ký 404 được bao gồm trong tệp tin cấu hình của app trong đó thư viện định tuyến 260 được phối hợp.

Bộ xử lý URI (mã nhận dạng tài nguyên đồng nhất) 408 nhận liên kết từ hệ điều hành. Ví dụ, liên kết có thể bắt nguồn từ trình duyệt hoặc từ app bên thứ ba. Bộ xử lý URI 408 trích xuất ID duy nhất từ liên kết, có thể đơn giản là văn bản trong URI về bên phải của cặp dấu gạch chéo lên theo sơ đồ.

Bộ định tuyến nội bộ 412 tham khảo kho dữ liệu chỉ mục 416 để xác định xem liệu ID duy nhất từ URI tương ứng với tập hợp liên kết hoặc tương ứng với cuộc gọi hoạt động sẽ được thực hiện cho bộ xử lý App A 300. Theo cách khác, bộ định tuyến nội bộ 412 quyết định xem liệu tập hợp liên kết có mặt để đạt trạng thái được chỉ báo với ID duy nhất hay không hoặc liệu cuộc gọi hoạt động có cần được tạo ra cho bộ xử lý App A 300 để đạt trạng thái được chỉ báo với ID duy nhất hay không. Khi bộ định tuyến nội bộ 412 xác định ID duy nhất tương ứng với tập hợp liên kết, bộ định tuyến nội bộ 412 gửi ID tập hợp liên kết đến máy trạng thái 420.

Theo các phương án thực hiện khác nhau, ID tập hợp liên kết có thể có cùng ID duy nhất hoặc có thể khác nhau được xác định bằng bản đồ từ kho dữ liệu chỉ mục 416. Như được mô tả chi tiết hơn sau đây, máy trạng thái 420 truy hồi tập hợp liên kết từ kho dữ liệu tập hợp liên kết 424 và thực thi mỗi thao tác giao diện người dùng trong tập hợp liên kết sử dụng bộ truyền động phát lại UI 428. Máy trạng thái 420 có thể giám sát trạng thái của App A 232 để đảm bảo là mỗi sự kiện giao diện người dùng được xử lý bởi App A 232 trước khi chuyển đến sự kiện UI tiếp theo.

Nếu bộ định tuyến nội bộ 412 xác định là ID duy nhất từ bộ xử lý URI 408 tương ứng với hoạt động hoặc đối tượng dữ liệu được xử lý bởi bộ xử lý App A 300, tín hiệu được gửi đến bộ tạo cuộc gọi hoạt động 432. Bộ tạo cuộc gọi hoạt động 432 tạo tín hiệu hoạt động người dùng tiếp tục xác định hoạt động hoặc đối

tượng dữ liệu liên quan để liên kết. Các tham số được yêu cầu để tạo tín hiệu hoạt động người dùng tiếp tục có thể đạt được từ kho dữ liệu chỉ mục 416. Tín hiệu hoạt động người dùng tiếp tục được truyền đến bộ xử lý App A 300 thông qua giao diện bộ xử lý 436, có thể gửi tín hiệu hoạt động người dùng tiếp tục đến bộ xử lý App A 300 nếu bộ xử lý App A 300 nhận được tín hiệu trực tiếp từ hệ điều hành.

Như được đề cập ở trên, thư viện định tuyến 260 nhận phương pháp được gọi bởi hệ điều hành để cung cấp tín hiệu hoạt động người dùng tiếp tục cho app. Sau đó, bộ xử lý hoạt động 440 của thư viện định tuyến 260 nhận tín hiệu hoạt động người dùng tiếp tục. Bộ xử lý hoạt động 440 xác định xem liệu tín hiệu hoạt động người dùng tiếp tục cần được xử lý bởi bộ xử lý App A 300 được triển khai sẵn bởi nhà phát triển. Việc xác định này có thể dựa trên kho dữ liệu chỉ mục 416. Trong các trường hợp này, bộ xử lý hoạt động 440 đề xuất hoạt động người dùng tiếp tục truyền qua bộ xử lý App A 300 thông qua giao diện bộ xử lý 436.

Trong các app trong đó nhà phát triển không mã hóa bộ xử lý như bộ xử lý App A 300, bộ xử lý hoạt động 440 có thể không bao giờ truyền qua các hoạt động và bộ tạo cuộc gọi hoạt động 432 và giao diện bộ xử lý 436 có thể không hoạt động. Nếu bộ xử lý hoạt động 440 xác định là tín hiệu hoạt động người dùng tiếp tục, theo kho dữ liệu chỉ mục 416, có thể được khôi phục sử dụng tập hợp liên kết được lưu trữ, ID tập hợp liên kết được đề xuất cho máy trạng thái 420 để thực thi.

Cho đến nay, các khía cạnh của thư viện định tuyến 260 liên quan đến các liên kết ngầm mới đến, liệu ở dạng của liên kết hoặc tín hiệu hoạt động người dùng tiếp tục từ hệ điều hành, được mô tả. Thư viện định tuyến 260 cũng có thể khai triển chức năng gửi các đối tượng hoạt động và các đối tượng dữ liệu đến hệ điều hành. Điều này cho phép hệ điều hành trả lại các kết quả tìm kiếm từ app.

Bộ lập chỉ mục dữ liệu 460 xác định các đối tượng dữ liệu nào của app để lập chỉ mục dựa trên các quy tắc từ kho dữ liệu các quy tắc nghiệp vụ 464. Chỉ làm ví dụ, bộ lập chỉ mục dữ liệu 460 có thể đáp lại tín hiệu được tải app để thực thi khi app này chạy thành công trên thiết bị. Bộ lập chỉ mục dữ liệu 460 có thể, khi chạy app lần thứ nhất, gửi tập hợp định trước của các đối tượng dữ liệu đến hệ điều hành thông qua giao diện hệ điều hành 468.

Chỉ làm ví dụ, tương ứng với các đối tượng dữ liệu, giao diện hệ điều hành 468 có thể đơn giản là viện dẫn của phương pháp `indexSearchableItems()`. Việc tải tiếp theo của app, bộ lập chỉ mục dữ liệu 460 có thể cập nhật các đối tượng dữ liệu chỉ mục dựa trên sự liên quan liên tiếp của chúng, dựa trên các thay đổi dữ liệu được tạo ra cho, hoặc dựa trên quy tắc thay đổi từ kho dữ liệu các quy tắc nghiệp vụ 464.

Ví dụ, nếu tên của các món ăn được lập chỉ mục, khi tên của các món ăn mới được thêm vào app, bộ lập chỉ mục dữ liệu 460 có thể thêm tên của các món ăn bổ sung vào chỉ mục hệ điều hành. Trong lúc đó, nếu kho dữ liệu các quy tắc nghiệp vụ 464 được cập nhật để chỉ báo tên của các món ăn không được lập chỉ mục nữa, bộ lập chỉ mục dữ liệu 460 có thể xóa các đối tượng dữ liệu tương ứng từ chỉ mục hệ điều hành.

Bộ cập nhật kho dữ liệu 472 chịu trách nhiệm về việc cập nhật một hoặc nhiều kho dữ liệu, bao gồm kho dữ liệu tập hợp liên kết 424, kho dữ liệu chỉ mục 416, và kho dữ liệu các quy tắc nghiệp vụ 464. Bộ cập nhật kho dữ liệu 472 liên lạc máy chủ dữ liệu 284 trên Fig.2 thông qua giao diện mạng 476. Giao diện mạng 476 có thể xử lý tính xác thực và mã hóa để tính hợp lệ của kho dữ liệu cập nhật có thể được xác nhận.

Bộ cập nhật kho dữ liệu 472 có thể hoạt động khi app phối hợp thư viện định tuyến 260 được mở đầu tiên trên thiết bị và ở các khoảng thời gian chu kỳ liên tiếp. Ngoài ra, bộ cập nhật kho dữ liệu 472 có thể thực hiện các cập nhật bổ sung đáp lại nhu cầu dữ liệu, như nếu ID tập hợp liên kết không có mặt trong kho dữ liệu tập hợp liên kết 424.

Bộ tạo lập hoạt động 480 tạo các đối tượng hoạt động để đánh dấu các hoạt động với hệ điều hành. Giao diện hệ điều hành 468 có thể đánh dấu hoạt động bằng cách yêu cầu phương pháp `becomeCurrent()` trên đối tượng hoạt động được tạo ra. Bộ tạo lập hoạt động 480 có thể phản hồi hình xuất hiện, như tín hiệu `viewDidAppear`, hoặc với một số sự kiện khác, như tải hình hoặc xuất hiện.

Bộ tạo lập hoạt động 480 có thể cập nhật các đối tượng hoạt động làm các tương tác với hình xuất hiện. Bộ tạo lập hoạt động 480 có thể, do đó, thu nhận các phương pháp bổ sung cung cấp đầu vào người dùng hình để đối tượng hoạt động

tương ứng có thể được cập nhật. Ví dụ, khi người dùng hiệu chỉnh hộp văn bản, bộ tạo lập hoạt động 480 có thể cập nhật đối tượng hoạt động sau mỗi lần nhấn phím.

Ngoài tạo đối tượng hoạt động ra, bộ tạo lập hoạt động 480 có thể chỉ báo cho bộ lập chỉ mục dữ liệu 460 là các đối tượng dữ liệu nhất định cần được lập chỉ mục. Các quyết định này có thể dựa trên các quy tắc từ kho dữ liệu các quy tắc nghiệp vụ 464. Ví dụ, bộ tạo lập hoạt động 480 có thể tạo đối tượng hoạt động dựa trên việc người dùng xem xét danh sách các nhà hàng đối với món ăn đặc trưng. Liệu đối tượng hoạt động này được tạo ra có được xác định bằng kho dữ liệu các quy tắc nghiệp vụ 464 hay không. Kho dữ liệu các quy tắc nghiệp vụ 464 cũng có thể lệnh cho bộ tạo lập hoạt động 480 báo hiệu cho bộ lập chỉ mục dữ liệu 460 là một hoặc nhiều kết quả hàng đầu từ tìm kiếm cần được lập chỉ mục.

Ví dụ, nếu người dùng xem tập hợp các nhà hàng Thai và kết quả liên quan nhất là nhà hàng Amarin Thai, sau đó người dùng có thể mong muốn tìm kiếm đối với Amarin Thai. Bằng cách lập chỉ mục đối tượng dữ liệu Amarin Thai, hệ điều hành có thể đưa ra kết quả trực tiếp cho app đối với Amarin Thai. Ngoài ra, phụ thuộc vào cách mà chỉ mục hệ điều hành nhận dạng các kết quả tìm kiếm, người dùng tìm kiếm trong hệ điều hành đối với “Thai” có thể xem cả hoạt động đối với app liên quan đến danh sách các nhà hàng Thai cũng như đối tượng dữ liệu cụ thể Amarin Thai.

Khi hệ điều hành gửi tín hiệu hoạt động người dùng tiếp tục liên quan đến một trong số các đối tượng hoạt động từ bộ tạo lập hoạt động 480 hoặc các đối tượng dữ liệu từ bộ lập chỉ mục dữ liệu 460, tập hợp liên kết tương ứng có mặt trong kho dữ liệu tập hợp liên kết 424 để đạt được các đối tượng hoạt động hoặc các đối tượng dữ liệu này vì chính app không được lập trình bởi nhà phát triển để làm phù hợp với chúng.

Các đối tượng hoạt động và các đối tượng dữ liệu được tạo ra với bộ tạo lập hoạt động 480 và bộ lập chỉ mục dữ liệu 460 tương ứng được cung cấp cho kho dữ liệu chỉ mục 416 để kho dữ liệu chỉ mục 416 nhận biết các đối tượng hoạt động và các đối tượng dữ liệu nào có thể được xử lý bởi thư viện định tuyến 260 thay vì được truyền qua bộ xử lý App A 300.

Bộ lập chỉ mục từ xa 484 có thể được hướng dẫn bởi kho dữ liệu các quy tắc nghiệp vụ 464 để phản ánh các đối tượng hoạt động và/hoặc các đối tượng dữ liệu từ kho dữ liệu chỉ mục 416 đến chỉ mục từ xa. Chỉ mục từ xa này có thể được duy trì bởi bộ cung cấp của thư viện định tuyến 260 hoặc với dịch vụ tìm kiếm. Các đối tượng hoạt động và các đối tượng dữ liệu có thể được đánh dấu với mã nhận dạng riêng (hoặc được giả sử là riêng tư trừ khi được đánh dấu là mã nhận dạng công khai). Mã nhận dạng riêng này có thể ngăn chặn đối tượng không được tải lên đến chỉ mục từ xa hoặc có thể ngăn chặn đối tượng không bị lộ với chỉ mục từ xa trong các kết quả tìm kiếm công khai. Chỉ báo riêng tư này có thể được cung cấp bởi nhà phát triển là một phần của quy trình tích hợp của app cho công nhà phát triển và có thể được cập nhật bởi nhà phát triển theo thời gian.

Hệ thống tìm kiếm 288 nhận các cơ chế truy cập từ máy chủ dữ liệu 284 sát nhập với các mã nhận dạng của các tập hợp liên kết hoặc chính các tập hợp liên kết. Hệ thống tìm kiếm 288 có thể có sẵn thông tin về các trạng thái quan tâm của App A và sau đó có thể kết hợp các cơ chế truy cập với các trạng thái tương ứng. Theo các phương án thực hiện khác, hệ thống tìm kiếm 288 có thể thu thập và tìm kiếm App A để thu siêu dữ liệu cho các trạng thái quan tâm, và có thể theo dõi các cơ chế truy cập để tìm nội dung của các trạng thái.

Trong một mô hình sử dụng, người dùng của các thiết bị người dùng 220 thực hiện tìm kiếm đối với chức năng nhất định và/hoặc đối với thực thể nhất định, như các đánh giá về nhà hàng quan tâm. Người dùng có thể thực hiện tìm kiếm này với app tìm kiếm độc lập hoặc, như được thể hiện trên Fig.2, tìm kiếm web được thực hiện thông qua trình duyệt 228. Hệ thống tìm kiếm 288, như được mô tả chi tiết hơn sau đây, đề xuất các kết quả tìm kiếm to trình duyệt 228.

Các kết quả này có thể bao gồm kết quả tương ứng với App A 232, và có thể bao gồm liên kết chỉ báo trạng thái ngầm cụ thể trong App A 232. Nếu người dùng chọn liên kết tương ứng với App A 232, liên kết được chuyển tiếp cho bộ định tuyến liên kết 244 và sau đó được truyền đến thư viện định tuyến 260 của App A 232. App A 232 sau đó được điều khiển bởi thư viện định tuyến 260 để hiển thị trạng thái ngầm được chỉ báo sử dụng tập hợp liên kết tương ứng.

Theo các phương án thực hiện khác nhau, thư viện định tuyến 260 nhận các nội dung của các tập hợp liên kết từ các tham số được mã hóa trong chính liên kết đó. Ví dụ, liên kết (trong trường hợp này, URI) có thể bao gồm sơ đồ (“portal-appa://”) được theo sau bởi mã hóa theo thứ tự của mỗi sự kiện UI của tập hợp liên kết theo chuỗi dẫn đến hình mong muốn. Ví dụ, tập hợp liên kết có thể được xác định trong cấu trúc dữ liệu JSON (ký hiệu đối tượng JavaScript) được mã hóa sử dụng Base64.

Hệ thống tìm kiếm 288, do đó, có thể cung cấp liên kết sẽ dẫn trực tiếp người dùng đến nội dung liên quan trong App A 232. Hệ thống tìm kiếm 288 có thể đưa ra các kết quả tìm kiếm đến các app khác. Ví dụ, app của các khách sạn có thể hỏi hệ thống tìm kiếm 288 đối với các nhà hàng gần khách sạn được chọn của người dùng và hệ thống tìm kiếm 288 có thể đưa ra các kết quả tìm kiếm nhà hàng cho app của các khách sạn. Trong trường hợp trong đó App A 232 là App đánh giá nhà hàng, app của các khách sạn có thể liên kết trực tiếp vào trạng thái ngầm của App A 232 tương ứng với nhà hàng quan tâm. Hệ thống tìm kiếm 288 có thể đưa ra các kết quả tìm kiếm ở dạng của các DVC.

DVC đối với app hoặc trạng thái của app thể hiện thông tin bổ sung, ngoài việc nhận dạng của app hoặc trạng thái của app ra. Ví dụ, thông tin có thể bao gồm tên của trạng thái app hoặc mô tả của trạng thái app, có thể là đoạn văn bản từ trạng thái app. Siêu dữ liệu khác có thể được cung cấp từ trạng thái app, bao gồm các hình ảnh, vị trí, số lượng đánh giá, đánh giá trung bình, và các bộ chỉ báo trạng thái. Ví dụ, bộ chỉ báo trạng thái “mở cửa bây giờ” hoặc “đóng” có thể được áp dụng cho doanh nghiệp phụ thuộc vào liệu thời điểm hiện tại ở trong giờ làm việc của doanh nghiệp.

Một số DVC có thể nhấn mạnh thông tin dẫn đến DVC được chọn làm kết quả tìm kiếm. Ví dụ, văn bản trong DVC thích hợp với câu hỏi của người dùng có thể được biểu diễn đậm hoặc nghiêng. DVC cũng có thể kết hợp các phần tử cho phép các thao tác trực tiếp, như khả năng để gọi ngay sự thiết lập hoặc để chuyển tiếp trực tiếp đến app bản đồ để nhận các hướng điều hướng để thiết lập.

Các tương tác khác với DVC (như gõ hoặc bấm bất kỳ khu vực nào khác của DVC) có thể mang người dùng đến trạng thái được chỉ báo hoặc app. Như được mô

tả chi tiết hơn sau đây, điều này có thể được hoàn thiện bằng cách mở app liên quan hoặc, nếu app không được cài đặt, mở trang web liên quan đến trạng thái mong muốn của app. Trong các phương án thực hiện khác, app không được cài đặt có thể được tải xuống, được cài đặt, và sau đó được chạy để đạt trạng thái mong muốn của app.

Nói cách khác, DVC bao gồm thông tin nhận dạng về app hoặc trạng thái cũng như nội dung bổ sung từ app hoặc chính trạng thái đó. Nội dung bổ sung cho phép người dùng tạo nhiều lựa chọn được thông báo hơn về kết quả nào để chọn, và thậm chí có thể cho phép người dùng thực hiện trực tiếp thao tác mà không điều hướng đến trạng thái của app. Nếu thao tác người dùng muốn thu thông tin, trong một số trường hợp chính DVC có thể cung cấp thông tin cần thiết để hoàn thành thao tác này.

Theo các phương án thực hiện khác nhau, thư viện định tuyến có thể giống với mỗi app với ngoại trừ duy nhất là tên của sơ đồ tùy chỉnh mà thư viện định tuyến sẽ đăng ký nhờ việc cài đặt. Sơ đồ này có thể được tạo ra bằng cách ghép văn bản của công nhà phát triển với văn bản của tên app, và có thể bao gồm dấu ngăn cách như dấu nối hoặc đường gạch dưới. Tên của app có thể thay đổi theo thời gian, nhưng sơ đồ có thể được chỉnh sau khi thiết lập trước để cung cấp sự tương thích ngược với các phiên bản cũ của app mà thư viện định tuyến của chúng chỉ có thể nhận ra sơ đồ ban đầu.

Theo các phương án thực hiện khác nhau, thư viện định tuyến có thể được cập nhật với cập nhật bảo mật, sửa lỗi, và các bổ sung dấu hiệu trong khi duy trì sự tương thích ngược. Các nhà phát triển, như nhà phát triển 204, có thể tải xuống và kết hợp phiên bản mới nhất của thư viện định tuyến mỗi lần chúng phát hành phiên bản mới của app của chúng cho nền tảng phân phối số 212. Theo một số phương án thực hiện, quy trình thiết kế/biên dịch có thể tải xuống tự động phiên bản mới nhất của thư viện định tuyến khi giải quyết các phụ thuộc của các mối liên kết.

Hệ thống phân tích ngoại tuyến 280 có thể cần được viện dẫn mỗi lần nhà phát triển 204 chuẩn bị phiên bản mới của App A đối với nền tảng phân phối số 212. Ngoài ra, khi nhà phát triển 204 bổ sung thêm nội dung vào App A, hệ thống phân tích ngoại tuyến 280 có thể được viện dẫn để xác định các tập hợp liên kết cần

được truy cập nội dung được bổ sung. Sự việ dẫn của hệ thống phân tích ngoại tuyến 280 có thể được thực hiện với yêu cầu của nhà phát triển 204 hoặc có thể được thực hiện trên cơ sở theo chu kỳ. Bất kỳ cập nhật hoặc bổ sung nào cho các tập hợp liên kết được lưu trữ bởi máy chủ dữ liệu 284 và có thể được cung cấp cho hệ thống tìm kiếm 288 để các liên kết trong các kết quả tìm kiếm có các cơ chế truy cập cập nhật nhất.

Theo một số phương án thực hiện, các tập hợp liên kết có thể được truyền thông đến thư viện định tuyến 260 sử dụng cơ chế khác ngoài phép lồng trực tiếp trong URI. Ví dụ, điều này có thể cần thiết khi liên kết có độ dài tối đa ngắn hơn số lượng ký tự và tập hợp liên kết được mã hóa có thể yêu cầu.

Thư viện định tuyến 260 có thể, do đó, theo một số phương án thực hiện, tải xuống gói của các tập hợp liên kết từ máy chủ dữ liệu 284. Các kết quả tìm kiếm từ hệ thống tìm kiếm 288 hoặc các liên kết khác sau đó có thể tham khảo mã nhận dạng duy nhất, thư viện định tuyến 260 tạo sơ đồ cho tập hợp liên kết. Chỉ làm ví dụ, mã nhận dạng duy nhất có thể được tạo ra từ chức năng và thực thể, như “restaurant_reviews” và “Amarin_Thai”. Theo một ví dụ cụ thể, URI “portal-appa://restaurant_reviews/Amarin_Thai” có thể được giải quyết bởi thư viện định tuyến 260 đến tập hợp liên kết đạt được trạng thái đánh giá nhà hàng đối với nhà hàng Amarin Thai từ trạng thái mặc định của App A 232. Tuy nhiên, không có yêu cầu mã nhận dạng duy nhất đọc được bởi người.

Theo các phương án thực hiện khác, thư viện định tuyến 260 có thể tra cứu máy chủ dữ liệu 284 đáp lại nhận liên kết ngầm. Bằng cách cung cấp mã nhận dạng duy nhất, thư viện định tuyến 260 có thể tải xuống tập hợp liên kết cần thiết từ máy chủ dữ liệu 284. Trong các phương án thực hiện này, thư viện định tuyến 260 có thể trừ các sự kiện giao diện người dùng để truy cập mạng không được yêu cầu để giải quyết các liên kết ngầm được thăm gần đây. Việc trừ trước một vài hoặc tất cả các tập hợp liên kết cũng có thể được thực hiện, như khi App A 232 chạy đầu tiên trên thiết bị người dùng 220.

Việc trừ trước thậm chí có thể xảy ra sớm hơn, như khi nhà phát triển 204 chuẩn bị App A để phân phối. Theo cách khác, toàn tập hợp hoặc tập hợp con của các tập hợp liên kết có thể được bao gồm cùng với App A để khi mã nhận dạng duy

nhất được nhận thông qua liên kết, tập hợp liên kết thích hợp có thể được chọn bởi thư viện định tuyến 260 không làm chậm. Việc trữ trước có thể được kết hợp với việc xác minh theo chu kỳ của các tập hợp liên kết bởi thư viện định tuyến 260, như bằng cách kiểm tra số phiên bản của gói của các tập hợp liên kết ở khoảng chu kỳ lịch, như một lần một tháng. Theo các phương án thực hiện khác, máy chủ dữ liệu 284 có thể gửi thông báo thúc đẩy đến thư viện định tuyến 260 chỉ báo là các tập hợp liên kết mới có sẵn.

Đối với phạm vi mà nhà phát triển 204 đã triển khai một số liên kết ngầm trong App A, bộ định tuyến được triển khai bởi nhà phát triển (không được thể hiện) trong App A 232 sẽ nhận các liên kết từ bộ định tuyến liên kết 244. Mã định tuyến của nhà phát triển sẽ không được đăng ký “portal-appa://” nhưng thay vì sơ đồ độc lập với cổng nhà phát triển 208, như “appa.com://”. Theo các phương án thực hiện khác nhau, các URI được xác định bởi nhà phát triển không thể có sẵn công khai hoặc chỉ có thể có sẵn cùng với tương quan được thiết lập với nhà phát triển 204.

Một số hệ điều hành, bao gồm hệ điều hành iOS, có hạ tầng hộp cát với giới hạn phạm vi mà app có thể truy cập các tài nguyên hoặc các vùng của bộ nhớ vượt quá những gì cụ thể được cấp phát cho app. Hộp cát tăng cường độ an toàn, khiến nó khó khăn cho phần mềm gây hại làm tổn hại đến hệ thống người dùng. Vì thư viện định tuyến 260 chạy trong ngữ cảnh của App A 232, hộp cát không gây khó khăn cho các tương tác giữa thư viện định tuyến 260 và các bộ điều khiển hình chịu trách nhiệm về các hình 268.

Lưu đồ cấp độ cao

Trên Fig.5, tổng quan về hoạt động theo sáng chế tương ứng với một nhà phát triển và một app bắt đầu tại 504. Nhà phát triển trước hết phát triển app (ở đây được gọi là “App A”). App A không thể có chức năng mà có thể bộc lộ các liên kết ngầm cho tất cả trong số các trạng thái của nó. Ngoài ra, App A không thể lập chỉ mục chức năng hoặc chức năng khôi phục hoạt động cho một vài hoặc tất cả các trạng thái. Ngoài ra, bất kỳ liên kết ngầm hoặc chức năng tìm kiếm mà App A sở hữu có thể yêu cầu sự can thiệp của lập trình viên để sửa đổi. Do đó, nhà phát triển có thể thu lợi từ thư viện định tuyến được mô tả với sáng chế.

Tại 508, nhà phát triển cung cấp bản sao của app A cho công nhà phát triển. Điều này có thể là thiết kế chuẩn, như thứ cần được cung cấp cho nền tảng phân phối số. Thực tế, nếu App A được xuất bản cho nền tảng phân phối số, công nhà phát triển có thể thu bản sao của App A từ nền tảng phân phối số. Trong khi thiết kế chuẩn thường bị giới hạn để chạy trên các bộ mô phỏng hoặc các thiết bị phần cứng, thay vì nhà phát triển có thể cung cấp thiết kế đặc biệt của App A, như thiết kế được điều chỉnh cho môi trường mô phỏng. Môi trường mô phỏng thậm chí có thể chạy trên cấu trúc khác (như cấu trúc x86) so với app (có thể chạy trên phần cứng thiết bị người dùng dựa trên cấu trúc ARM).

Tại 512, điều khiển bắt đầu phân tích ngoại tuyến của App A. Phân tích ngoại tuyến có thể bao gồm xác định các trạng thái mà các liên kết ngầm có sẵn trong App A. Phân tích ngoại tuyến cũng có thể bao gồm xác định các hoạt động nào có thể được khôi phục trong App A trong đó dữ liệu có thể được lập chỉ mục bởi App A dựa trên mã được viết bởi nhà phát triển. Chỉ làm ví dụ, phân tích ngoại tuyến có thể bao gồm phân tích tĩnh và động của App A, được mô tả chi tiết hơn trong đơn sáng chế Mỹ số 14/984,642, nộp ngày 30/12/2015, có tiêu đề “Static Analysis and Reconstruction of Deep Link Handling and Compiled Applications”, với tác giả sáng chế thứ nhất tên là Kalyan Desineni, và có số hồ sơ 17100-000030-US. Toàn bộ phần bộc lộ của đơn này được đưa vào bằng cách viện dẫn.

Tại 516, nhà phát triển xác định các quy tắc nghiệp vụ cho App A. Ví dụ, nhà phát triển xác định cho công nhà phát triển các trạng thái nào của App A để cung cấp chức năng liên kết ngầm để, các hoạt động nào của App A để đánh dấu, và dữ liệu nào để lập chỉ mục. Nhà phát triển có thể điều hướng bản sao của app A để xác định các trạng thái nào để cung cấp các liên kết ngầm. Ví dụ, nhà phát triển có thể tương tác với App A trong bộ mô phỏng, trong khi nền tảng phân tích ngoại tuyến giám sát các sự kiện giao diện người dùng để phát lại sau đó.

Nhà phát triển có thể đơn giản được yêu cầu điều hướng đến các trạng thái quan tâm, không cần hiểu là các sự kiện giao diện người dùng được ghi cho các mục đích liên kết ngầm. Đối với mỗi trạng thái nhà phát triển điều hướng đến, nhà phát triển có thể xác định xem liệu một hoặc nhiều hoạt động được thực hiện trong đó trạng thái cần được đánh dấu và có thể nhận dạng một hoặc nhiều phần dữ liệu

để lập chỉ mục từ trạng thái đó. Nhà phát triển có thể xác định siêu dữ liệu nào để bao gồm với dữ liệu được lập chỉ mục, và có thể báo cáo vi phạm các hoạt động hoặc dữ liệu là công khai hoặc cá nhân.

Tại 520, song song với phân tích ngoại tuyến, điều khiển đề xuất bản sao của thư viện định tuyến từ công nhà phát triển đến nhà phát triển. Tại 524, nhà phát triển tích hợp thư viện định tuyến vào App A. Ví dụ, nó có thể bao gồm việc thêm tham khảo trong tệp tin cấu hình cho sơ đồ được xử lý bằng khả năng liên kết của thư viện định tuyến cũng như thêm các kiểu hoạt động mà thư viện định tuyến sẽ nhận ra.

Theo một số phương án thực hiện, các hoạt động được nhận dạng bởi thư viện định tuyến sẽ là thừa với các hoạt động được nhận dạng sẵn bởi App A, như khi App A cho phép sẵn cho một số hoạt động khôi phục hoặc lập chỉ mục dữ liệu. Nhà phát triển cũng liên kết nhị phân đến dự án app A trong môi trường phát triển được tích hợp. Trong các phương án thực hiện khác nhau, tập lệnh có thể được cung cấp cho nhà phát triển để thực hiện các thao tác phối hợp của 524.

Tại 528, nhà phát triển biên dịch App A, giờ đây bao gồm thư viện định tuyến, và đề xuất App A cho một hoặc nhiều nền tảng phân phối số. Tại 532, người dùng của các thiết bị người dùng cài đặt App A từ nền tảng phân phối số. Tại 536, là một phần của việc cài đặt, thư viện định tuyến của App A đăng ký sơ đồ liên kết tùy biến được xử lý bởi thư viện định tuyến cũng như các kiểu hoạt động được xử lý bởi thư viện định tuyến. Ví dụ, việc đăng ký này có thể được thực hiện bởi hệ điều hành dựa trên tệp tin cấu hình, như Info.plist.

Tại 540, thư viện định tuyến của ứng dụng được cài đặt hiện tại cập nhật các cơ chế truy cập và các quy tắc nghiệp vụ. Ví dụ, các cơ chế truy cập có thể là các tập hợp liên kết để đạt các trạng thái nhất định. Các quy tắc nghiệp vụ nhận dạng các trạng thái nào cần được truy cập, các hoạt động nào cần được đánh dấu, và dữ liệu nào cần được lập chỉ mục.

Khi ứng dụng chạy đầu tiên, thư viện định tuyến có thể tải xuống toàn bộ tập hợp của các cơ chế truy cập và các quy tắc nghiệp vụ để không có độ trễ truy cập xảy ra khi cơ chế truy cập hoặc quy tắc nghiệp vụ cần thiết. Trong các phương án thực hiện khác, nhờ chạy đầu tiên, thư viện định tuyến chỉ tải xuống tập hợp con

của các cơ chế truy cập, như các cơ chế truy cập đối với các trạng thái phổ biến nhất. Cập nhật tại 540 có thể được thực hiện mỗi lần app được mở hoặc ở khoảng thời gian định trước. Ngoài ra, thư viện định tuyến có thể cập nhật các cơ chế truy cập và/hoặc các quy tắc nghiệp vụ nếu cơ chế truy cập được yêu cầu (ví dụ, tập hợp liên kết) thiếu hoặc nếu cơ chế truy cập xuất hiện không làm việc.

Tại 544, thư viện định tuyến lập chỉ mục dữ liệu được chỉ báo bởi nhà phát triển theo các quy tắc nghiệp vụ. Ví dụ, khi app trước tiên tải, thư viện định tuyến có thể lập chỉ mục tập hợp dữ liệu được chỉ báo bởi nhà phát triển. Sau đó, khi dữ liệu được sửa đổi trong App A, thư viện định tuyến có thể cập nhật chỉ mục hệ điều hành dựa trên dữ liệu được cập nhật. Tại 548, thư viện định tuyến đánh dấu các hoạt động được chọn được tiến hành bởi người dùng. Ví dụ, khi người dùng thực hiện tìm kiếm, các tiêu chuẩn tìm kiếm và các bộ lọc có thể được đánh dấu là hoạt động mà người dùng có thể khôi phục trên thiết bị đó hoặc chuyển đến các thiết bị khác.

Tại 552, thư viện định tuyến trong App A xử lý các liên kết App A độc quyền được chỉ báo bởi sơ đồ được đăng ký) được cung cấp cho thư viện định tuyến từ hệ điều hành. Tại 556, thư viện định tuyến xử lý các hoạt động được gửi bởi hệ điều hành, bao gồm các tín hiệu hoạt động người dùng tiếp tục. Các hoạt động này có thể đề cập đến các hoạt động được đánh dấu hoặc dữ liệu được lập chỉ mục. Điều khiển sau đó trở lại đến 540. Mặc dù 540, 544, 548, 552, và 556 được hiển thị liên tiếp, chúng có thể chạy không đồng bộ và song song.

Hoạt động thư viện định tuyến

Trên Fig.6, hoạt động làm ví dụ của thư viện định tuyến được thể hiện. Tại 604, trong việc cài đặt của App A trên thiết bị, sơ đồ URI App A độc quyền và các kiểu hoạt động trong thư viện định tuyến được đăng ký. Điều này có thể được thực hiện bởi hệ điều hành từ tệp tin cấu hình (cũng được biết là tệp tin kê khai) là một phần của việc xử lý cài đặt. Tại 608, việc điều khiển yêu cầu các tập hợp liên kết và dữ liệu cấu hình từ máy chủ dữ liệu. Điều này có thể được thực hiện nhờ lần chạy thứ nhất của App A trên các thiết bị người dùng.

Tại 612, điều khiển xác định xem liệu hoạt động hoặc URI App A độc quyền được nhận từ hệ điều hành. Nếu vậy, điều khiển chuyển đến 616; mặc khác, điều khiển chuyển đến 620. Tại 616, điều khiển dẫn ra bộ xử lý đối với hoạt động hoặc

URI App A độc quyền. Ví dụ, bộ xử lý có thể được triển khai như được thể hiện trên Fig.7. Chỉ làm ví dụ, bộ xử lý có thể đáp lại cuộc gọi `openURL()` được nhận bởi hệ điều hành hoặc từ tín hiệu hoạt động người dùng tiếp tục được tạo ra bởi hệ điều hành. Sau khi xử lý hoạt động hoặc URI App A độc quyền, điều khiển trở lại đến 612.

Tại 620, điều khiển xác định xem liệu sự kiện kích hoạt hoạt động có xảy ra hay không. Nếu vậy, điều khiển chuyển đến 624; mặc khác, điều khiển chuyển đến 628. Sự kiện kích hoạt hoạt động có thể là việc tải trạng thái hoặc thao tác được thực hiện trong trạng thái. Nhà phát triển có thể xác định sự kiện nào đủ điều kiện làm hoạt động khởi động. Nhà phát triển có thể thiết lập các quy tắc nghiệp vụ lệnh rằng một số trạng thái khởi động hoạt động khi trạng thái được nhập trong khi đối với các trạng thái khác hoạt động được đánh dấu nhờ một số thao tác khác được thực hiện, như đầu vào người dùng.

Tại 624, điều khiển xác định xem liệu tập hợp liên kết có sẵn để trở lại hoạt động đó hay không. Nếu vậy, việc điều khiển chuyển đến 632; mặc khác, điều khiển chuyển đến 636. Nếu tập hợp liên kết trở lại hoạt động đó không có sẵn, thư viện định tuyến không thể khôi phục hoạt động khi được yêu cầu bởi hệ điều hành và do đó sẽ không đánh dấu hoạt động đó. Tại 632, tập hợp liên kết có sẵn và do đó điều khiển gửi đối tượng hoạt động đến hệ điều hành để đánh dấu hoạt động.

Tại 636, điều khiển xác định xem liệu hoạt động được chỉ báo bởi nhà phát triển là công khai. Nếu vậy, điều khiển chuyển đến 640; mặc khác, điều khiển tiếp tục tại 628. Tại 640, hoạt động được chỉ báo là công khai và do đó điều khiển tải lên hoạt động đến chỉ mục từ xa. Điều khiển sau đó tiếp tục tại 628. Trong các phương án thực hiện khác, điều khiển có thể tải lên hoạt động đến chỉ mục từ xa không kể việc hoạt động được chỉ báo là công khai hay không, và thay vì sử dụng bộ chỉ báo công khai để xác định xem liệu có làm lộ hoạt động đó cho bất kỳ người dùng nào của chỉ mục từ xa hay không.

Tại 628, điều khiển xác định xem liệu dữ liệu cần được lập chỉ mục theo các quy tắc nghiệp vụ được thiết lập bởi nhà phát triển. Nếu vậy, điều khiển chuyển đến 644; mặc khác, điều khiển chuyển đến 648. Ví dụ, khi chạy đầu tiên App A, có thể là tập hợp định trước của các đối tượng dữ liệu mà nhà phát triển muốn lập chỉ mục.

Sau đó, các trạng thái nhất định được truy cập, nhà phát triển có thể được xác định dữ liệu liên quan đến các trạng thái đó cần được lập chỉ mục. Ví dụ, trong app phát nhạc, nếu người dùng xem danh sách nghệ sĩ ở thể loại cụ thể, các đối tượng dữ liệu tương ứng với mỗi nghệ sĩ có thể được thêm vào chỉ mục hệ điều hành.

Tại 644, nếu tập hợp liên kết có sẵn đối với đối tượng dữ liệu mà nhà phát triển được chỉ báo cần được lập chỉ mục, điều khiển chuyển đến 652; mặc khác, điều khiển tiếp tục tại 648. Tại 652, tập hợp liên kết để đạt được trạng thái mà có thể truy cập đối tượng dữ liệu đó có sẵn và do đó đối tượng dữ liệu được gửi đến hệ điều hành. Nếu tập hợp liên kết không có sẵn, thư viện định tuyến sẽ không thể phục vụ yêu cầu từ hệ điều hành để hiển thị đối tượng dữ liệu đến người dùng.

Trong khi Fig.6 thể hiện hoạt động của thư viện định tuyến, nhà phát triển có thể có logic được thực thi trong App A mà sẽ lập chỉ mục các đối tượng dữ liệu và đánh dấu các hoạt động, tách khỏi thư viện định tuyến. Việc lập chỉ mục và đánh dấu này không bị loại trừ với hoạt động của thư viện định tuyến. Do đó, ngay cả khi thư viện định tuyến không lập chỉ mục đối tượng dữ liệu hoặc đánh dấu hoạt động, chính App A có thể thực hiện nhiệm vụ đó. Để App A sau đó khôi phục hoạt động khi được lệnh bởi hệ điều hành, App A sẽ có logic được triển khai bởi nhà phát triển để đạt các trạng thái thích hợp. Logic này có thể bao gồm sự dẫn động trực tiếp của các bộ điều khiển hình hoặc có thể bao gồm tập hợp liên kết giao diện người dùng tương tự với việc được sử dụng bởi thư viện định tuyến.

Điều khiển tiếp tục từ 652 đến 656, trong đó nếu đối tượng dữ liệu được chỉ báo là công khai, điều khiển chuyển đến 660; mặc khác, điều khiển tiếp tục tại 648. Tại 660, điều khiển tải lên đối tượng dữ liệu đến chỉ mục từ xa và tiếp tục tại 648. Ngoài ra, mã nhận dạng công khai có thể được bỏ qua ở thời điểm tải lên đối tượng dữ liệu đến chỉ mục từ xa và thay vì được tính khi xác định xem liệu có làm lộ các đối tượng dữ liệu cho người dùng khác.

Tại 648, điều khiển xác định xem liệu cập nhật dữ liệu cuối cùng của các tập hợp liên kết và dữ liệu cấu hình nhiều hơn thời gian định trước trong quá khứ. Nếu vậy, điều khiển chuyển đến 664 để cập nhật dữ liệu này; mặc khác, điều khiển trở lại 612. Mặc dù được thể hiện là lượng thời gian định trước, việc cập nhật dữ liệu có thể được thực hiện động dựa trên tần suất app được sử dụng, tần suất các tập hợp

liên kết hoặc dữ liệu cấu hình được thay đổi theo lịch sử, và/hoặc có thể dựa trên tín hiệu xung từ máy chủ dữ liệu chỉ báo là dữ liệu cần được cập nhật.

Hoạt động/xử lý URI

Trên Fig.7, hoạt động thư viện định tuyến làm ví dụ để xử lý các hoạt động và các URI độc quyền được thể hiện. Trong các phương án thực hiện khác nhau, quy trình trên Fig.7 có thể được gọi là 616 trên Fig.6. Điều khiển bắt đầu tại 704, trong đó nếu URI App A độc quyền được nhận, điều khiển chuyển đến 708; mặc khác, điều khiển chuyển đến 712.

Tại 708, điều khiển xác định xem liệu URI tương ứng với đối tượng dữ liệu có sẵn — tức là, đối tượng dữ liệu App A có thể lập chỉ mục và hiển thị độc lập với thư viện định tuyến. Nếu URI tương ứng với đối tượng dữ liệu có sẵn này, điều khiển tiếp tục tại 716; mặc khác, điều khiển chuyển đến 720. Tại 716, thư viện định tuyến có thể sử dụng cơ chế truy cập có sẵn trong App A để hiển thị trạng thái liên quan đến đối tượng dữ liệu đó. Để hoàn thiện nó, điều khiển có thể chuẩn bị yêu cầu hoạt động tức là tương tự hoặc giống với yêu cầu hoạt động mà hệ điều hành sẽ cung cấp cho App A để hiển thị trạng thái liên quan đến đối tượng dữ liệu đó. Điều khiển sau đó tiếp tục tại 724, trong đó yêu cầu hoạt động được chuẩn bị được truyền đến bộ xử lý App A có sẵn được viết bởi nhà phát triển độc lập với thư viện định tuyến. Điều khiển sau đó tiếp tục tại 712.

Tại 720, điều khiển xác định xem liệu URI tương ứng với hoạt động có sẵn — tức là, hoạt động App A có thể đánh dấu và khôi phục độc lập với thư viện định tuyến. Nếu vậy, điều khiển chuyển đến 728; mặc khác, điều khiển chuyển đến 732. Tại 728, điều khiển chuẩn bị yêu cầu hoạt động nhận dạng hoạt động đó có sẵn. Yêu cầu hoạt động có thể giống hoặc tương tự với yêu cầu mà hệ điều hành sẽ cung cấp cho App A để khôi phục hoạt động đánh dấu. Điều khiển sau đó tiếp tục tại 724.

Tại 732, điều khiển xác định xem liệu URI có tương ứng với tập hợp liên kết hay không. Nếu vậy, điều khiển chuyển đến 736; mặc khác, điều khiển chuyển đến 740. Mặc dù được gọi là tập hợp liên kết, cơ chế truy cập có thể là cơ chế khác, thư viện định tuyến sử dụng để điều hướng đến trạng thái nhất định. Ví dụ, thư viện định tuyến có thể được lập trình với tập hợp các tham số cho phép bộ điều khiển hình cụ thể được dẫn với các tham số đầu vào cụ thể để đến trạng thái nhất định. Tại

736, điều khiển phát lại tập hợp liên kết, như sử dụng quy trình được thể hiện trên Fig.8. Điều khiển sau đó tiếp tục tại 712.

Tại 740, URI App A độc quyền không thể được truyền đến bộ quản lý có sẵn App A và cơ chế truy cập (như tập hợp liên kết) không tồn tại. Do đó, việc xử lý lỗi được thực hiện. Ví dụ, việc xử lý lỗi có thể bao gồm hiển thị tin nhắn cho người dùng chỉ báo liên kết bị lỗi và hướng người dùng đến trạng thái mặc định của app. Trước khi hiển thị tin nhắn, điều khiển có thể làm mới các tập hợp liên kết được tải xuống để đảm bảo là các tập hợp liên kết gần nhất có sẵn. Theo ví dụ khác, điều khiển có thể cố gắng để tìm và sau đó chuyển URI tương thích, như bằng cách hỏi bản đồ của URI quá hạn đến các URI hoạt động. Điều khiển sau đó tiếp tục tại 712.

Tại 712, nếu yêu cầu hoạt động (như cuộc gọi phương pháp hoạt động người dùng tiếp tục) được nhận bởi App A (ví dụ, bởi ủy nhiệm app của App A), điều khiển chuyển đến 744; mặc khác, điều khiển chuyển đến 748. Tại 744, điều khiển xác định xem liệu hoạt động có cần được xử lý bởi thư viện định tuyến hay không. Nếu vậy, điều khiển chuyển đến 752; mặc khác, điều khiển chuyển đến 756.

Tại 752, hoạt động được xử lý bởi thư viện định tuyến vì thư viện định tuyến được đánh dấu hoạt động tương ứng hoặc được lập chỉ mục đối tượng dữ liệu tương ứng. Hoạt động có thể được khôi phục bằng cách phát lại tập hợp liên kết tương ứng với trạng thái được nhận dạng với yêu cầu hoạt động. Ví dụ, điều này có thể được thực hiện như được thể hiện trên Fig.8. Điều khiển sau đó trở lại đến Fig.6.

Tại 756, nếu thư viện định tuyến nhận biết là App A xử lý ít nhất một số yêu cầu hoạt động, điều khiển tiếp tục tại 760. Theo cách khác, nếu thư viện định tuyến chắc chắn là App A mặc khác không thể xử lý các hoạt động bất kỳ, điều khiển chuyển đến 764 trong đó việc xử lý lỗi được thực hiện. Ví dụ, việc xử lý lỗi có thể bao gồm hiển thị tin nhắn lỗi mà yêu cầu hoạt động không thể đáp ứng.

Tại 760, yêu cầu hoạt động được truyền qua đến bộ xử lý App A có sẵn với hi vọng là bộ xử lý App A có thể xử lý yêu cầu hoạt động. Điều khiển sau đó trở lại từ Fig.7. Tại 748, việc xử lý lỗi được thực hiện vì không có URI App A độc quyền được nhận hoặc yêu cầu hoạt động được nhận. Điều khiển không thể nhận biết thứ được nhận khác yêu cầu hoạt động hoặc URI App A độc quyền và do đó không thể

hiển thị bất kỳ thứ gì cho người dùng, thay vì đơn giản dẫn người dùng đến trạng thái mặc định của app. Điều khiển sau đó trở lại từ Fig.7.

Sự truyền tập hợp liên kết

Trên Fig.8, hoạt động làm ví dụ của thư viện định tuyến khi truyền tập hợp liên kết được thể hiện. Điều khiển bắt đầu tại 804, trong đó nếu tập hợp liên kết được nhúng trong URI, điều khiển chuyển đến 808; mặc khác, điều khiển chuyển đến 812.

Như được mô tả ở trên, các tập hợp liên kết cho phép thư viện định tuyến điều hướng đến các trạng thái ngầm nhất định có thể được nhận trong các liên kết hoặc có thể được chỉ báo với ID duy nhất trong liên kết. Khi được chỉ báo với ID duy nhất, thư viện định tuyến có thể phải yêu cầu tập hợp liên kết từ máy chủ dữ liệu. Theo một số phương án thực hiện, thư viện định tuyến do đó cài sẵn trước một vài hoặc tất cả các tập hợp liên kết. Hoạt động này có thể được lược bỏ nếu các tập hợp liên kết sẽ luôn được nhận trong các liên kết.

Theo một số phương án thực hiện, các tập hợp liên kết nhất định khi được mã hóa dài hơn độ dài cho phép tối đa của liên kết. Độ dài tối đa này có thể được lệnh với trình duyệt, với hệ điều hành, hoặc với một số hạn chế công nghệ hoặc kinh doanh khác. Các tập hợp liên kết này do đó có thể thu được từ máy chủ dữ liệu, trong khi các tập hợp liên kết ngắn hơn được chứa trong URI. Theo cách này, bộ tạo của URI (như hệ thống tìm kiếm) chuẩn bị URI bao gồm chính dữ liệu tập hợp liên kết khi độ dài liên kết tối đa sẽ không bị vượt quá, trong khi mặc khác chỉ bao gồm mã nhận dạng duy nhất của sự kiện giao diện người dùng.

Tại 808, điều khiển giải mã cấu trúc dữ liệu giữ tập hợp liên kết từ URI và điều khiển tiếp tục tại 816. Tại 812, vì tập hợp liên kết không được bao gồm trong URI, điều khiển xác định xem liệu tập hợp liên kết có được lưu trữ sẵn trong kho dữ liệu tập hợp liên kết hay không. Nếu vậy, điều khiển sử dụng tập hợp liên kết từ kho dữ liệu tập hợp liên kết và tiếp tục tại 816; mặc khác, điều khiển chuyển đến 820. Tại 820, điều khiển yêu cầu tập hợp liên kết từ máy chủ dữ liệu, và khi tập hợp liên kết được trở lại, điều khiển tiếp tục tại 816.

Theo các phương án thực hiện khác nhau, thư viện định tuyến thu tập hợp ban đầu của các tập hợp liên kết trước khi một trong số các tập hợp liên kết sẽ được

chỉ báo với URI. Ví dụ, thư viện định tuyến có thể được đóng gói trước với tập hợp ban đầu của các tập hợp liên kết hoặc thư viện định tuyến có thể tải xuống tập hợp ban đầu của các tập hợp liên kết nhờ việc cài đặt hoặc lần chạy thứ nhất trên thiết bị người dùng. Thư viện định tuyến có thể (ví dụ, thông qua xử lý thứ cấp) nhận theo chu kỳ các tập hợp liên kết mới để thực hiện chức năng. Theo một số phương án thực hiện, trải nghiệm người dùng được cải thiện có thể dẫn đến các tập hợp liên kết luôn thu được từ kho dữ liệu và không bao giờ được truy hồi từ máy chủ dữ liệu. Điều này làm tối thiểu hóa việc trễ để phản hồi do các tập hợp liên kết từ máy chủ dữ liệu và ngăn chặn các vấn đề có sẵn từ các vấn đề kết nối mạng.

Tại 816, điều khiển chọn sự kiện UI thứ nhất trong chuỗi được xác định bởi tập hợp liên kết. Tại 824, điều khiển phát lại sự kiện UI được chọn từ tập hợp liên kết. Tại 828, điều khiển xác định xem liệu có phát lại sự kiện UI được chọn thành công dẫn đến hình tiếp theo hoặc được cập nhật hình hiện tại như được mong đợi. Theo các phương án thực hiện khác nhau, điều khiển có thể chờ khoảng thời gian định trước trước khi kiểm tra thành công. Khoảng thời gian định trước này cho phép app đáp lại sự kiện UI.

Theo các phương án thực hiện khác nhau, cấu trúc dữ liệu tập hợp liên kết có thể bao gồm thời gian chờ được xác định, có thể áp dụng cho tất cả các sự kiện UI, hoặc có thể có thời gian chờ tách biệt đối với mỗi sự kiện UI. Hệ thống phân tích ngoại tuyến có thể phát hiện các sự kiện UI nhất định lâu hơn đối với app để đáp lại là một phần của quy trình xác định tập hợp liên kết. Trong các trường hợp này, hệ thống phân tích ngoại tuyến có thể chỉ báo là các chu kỳ làm trễ bổ sung cần được thực thi trước khi phát lại sự kiện UI khác.

Tại 828, sự thành công có thể thay vì được nhận dạng khi thư viện định tuyến phát hiện rằng phương pháp được nêu với các tham số nhất định tương ứng với sự kiện được phát lại. Theo cách khác, là một phần của phân tích ngoại tuyến, phản hồi được mong đợi với app cho sự kiện UI có thể được ghi. Khi phản hồi được mong đợi này được thấy, thành công có thể được suy ra. Nếu, tại 828, sự kiện UI xuất hiện được xử lý thành công, điều khiển chuyển đến 832; mặc khác, điều khiển chuyển đến 836.

Tại 836, điều khiển chờ khoảng thời gian định trước khác. Khoảng thời gian bổ sung này có thể cho phép sự phản hồi chậm không được mong đợi đối với sự kiện UI. Điều khiển tiếp tục tại 840, trong đó điều khiển một lần nữa cố gắng để xác minh rằng phát lại sự kiện UI được chọn dẫn đến việc xử lý được mong đợi thành công. Nếu vậy, điều khiển tiếp tục tại 832; mặc khác, điều khiển chuyển đến 844. Tại 844, nếu sự kiện UI được chọn được phát lại hai lần trong nỗ lực để đạt được phản hồi thành công, điều khiển chuyển đến 848.

Tại 848, lỗi được khai báo. Ví dụ, việc xử lý lỗi có thể khiến app trở lại trạng thái chờ và hiển thị tin nhắn thông báo cho người dùng rằng liên kết ngầm không thành công. Ngoài ra, tin nhắn có thể được gửi đến máy chủ dữ liệu chỉ báo là liên kết ngầm không thành công. Điều khiển sau đó trở lại từ Fig.8. Tại 844, nếu sự kiện UI được chọn chỉ được phát lại một lần, không có lỗi thứ hai, và do đó điều khiển trở lại đến 824 để nỗ lực phát lại sự kiện UI được chọn lần thứ hai.

Theo các phương án thực hiện khác nhau, sự thành công không được đánh giá, và các sự kiện UI được phát lại theo chuỗi như mong muốn trong hầu hết các trường hợp, liên kết ngầm mong muốn sẽ đạt được. Do đó, các phần tử 828, 836, 840, 844, và 848 có thể được thay thế với bước chờ đơn giản. Bước chờ có thể chờ trong việc trễ định trước được xác định theo kinh nghiệm để cho phép app phản hồi nhập liệu người dùng. Bước chờ có thể được sử dụng trong các hệ thống trong đó các sự kiện UI không thể được xử lý bởi app nhanh như chúng có thể được phát lại bởi thư viện định tuyến. Theo các phương án thực hiện khác, bước chờ có thể liên quan đến việc chờ cuộc gọi thứ tự do sự kiện UI. Ngay khi phản hồi app cho sự kiện UI được phát hiện, như bằng cách theo dõi cuộc gọi nhất định, điều khiển có thể cho phép sự kiện UI theo sau được phát lại. Tại 832, nếu có các sự kiện UI bổ sung, điều khiển chuyển đến 852; mặc khác, điều khiển trở lại từ Fig.8. Tại 852, điều khiển chọn UI tiếp theo trong tập hợp liên kết và trở lại 824.

Hệ thống tìm kiếm

Trên Fig.9, phương án thực hiện làm ví dụ của hệ thống tìm kiếm 288 bao gồm môđun tìm kiếm 900. Môđun tìm kiếm 900 bao gồm môđun phân tích truy vấn 904 nhận gói truy vấn, có thể có dạng chuỗi truy vấn trong URL được truyền bằng trình duyệt 228 trên Fig 2. Môđun phân tích truy vấn 904 phân tích truy vấn văn bản

từ gói truy vấn. Ví dụ, môđun phân tích truy vấn 904 có thể thể hóa văn bản truy vấn, lọc văn bản truy vấn, và thực hiện nhấn mạnh từ, từ đồng nghĩa, và dừng sự loại bỏ từ. Môđun phân tích truy vấn 904 cũng có thể phân tích dữ liệu bổ sung được lưu trữ trong gói truy vấn. Môđun phân tích truy vấn 904 đề xuất câu hỏi được thể hóa cho môđun tạo tập hợp 908.

Môđun tạo tập hợp 908 nhận dạng tập hợp xem xét của các hồ sơ trạng thái ứng dụng từ kho dữ liệu tìm kiếm 910 dựa trên các thẻ truy vấn. Các hồ sơ trạng thái ứng dụng (tương đương, app) tương ứng với các trạng thái riêng của các app. Theo các phương án thực hiện khác nhau, kho dữ liệu tìm kiếm 910 cũng có thể bao gồm các hồ sơ app. Trong các triển khai khác nhau, hồ sơ app có thể được lưu trữ dưới dạng hồ sơ trạng thái trạng thái app mà đơn giản có giá trị định trước, như rỗng, đối với trạng thái riêng của app.

Các hồ sơ trạng thái app trong kho dữ liệu tìm kiếm 910 có thể được tạo ra bằng cách thu thập và tìm kiếm các app theo các nguyên lý của sáng chế. Một vài hoặc tất cả các nội dung của các hồ sơ của của kho dữ liệu tìm kiếm 910 có thể được lập chỉ mục trong các chỉ mục đảo. Theo một số phương án thực hiện, môđun tạo tập hợp 908 sử dụng thư viện phần mềm APACHE LUCENE với Apache Software Foundation để nhận dạng các hồ sơ từ các chỉ số đảo. Môđun tạo tập hợp 908 có thể tìm kiếm các chỉ số đảo để nhận dạng các hồ sơ chứa một hoặc nhiều các thẻ truy vấn. Khi môđun tạo tập hợp 908 nhận dạng các hồ sơ thích hợp, môđun tạo tập hợp 908 có thể bao gồm ID duy nhất của mỗi hồ sơ được nhận dạng trong tập hợp xem xét. Ví dụ, môđun tạo tập hợp 908 có thể so sánh câu hỏi giới hạn đến tên trạng thái app và các thuộc tính app (như mô tả văn bản và các đánh giá người dùng) của hồ sơ trạng thái app.

Ngoài ra, theo một số phương án thực hiện, môđun tạo tập hợp 908 có thể xác định điểm ban đầu của hồ sơ tương ứng với truy vấn tìm kiếm. Điểm ban đầu có thể chỉ báo các nội dung của hồ sơ phù hợp với truy vấn thể nào. Ví dụ, điểm ban đầu có thể là hàm của các giá trị tần số-tần số nghịch (term frequency-inverse document frequency - TF-IDF) của các giới hạn truy vấn tương ứng.

Môđun xử lý tập hợp 912 nhận ID duy nhất của các hồ sơ trạng thái app được nhận dạng với môđun tạo tập hợp 908 và xác định điểm cuối cùng đối với một vài

hoặc tất cả của các ID. Số điểm cuối cùng chỉ báo sự liên quan của trạng thái app tương ứng với câu hỏi được thể hóa và các tham số ngữ cảnh. Theo các phương án thực hiện khác nhau, số điểm cao hơn chỉ báo sự liên quan đạt được lớn hơn.

Ví dụ, các mục khác trong gói truy vấn có thể đóng vai trò là các tham số ngữ cảnh. Dữ liệu định vị địa lý có thể giới hạn số điểm của (hoặc đơn giản là loại bỏ hoàn toàn) các app không phù hợp cho vị trí của các thiết bị người dùng. Danh sách đen trong gói truy vấn có thể khiến môđun xử lý tập hợp 912 loại bỏ các hồ sơ app và/hoặc các hồ sơ trạng thái app từ tập hợp xem xét khớp với các tiêu chuẩn trong danh sách đen, hoặc để thiết lập số điểm của chúng là giá trị rỗng, như không.

Môđun xử lý tập hợp 912 có thể tạo ra điểm cuối cùng dựa trên một hoặc nhiều các tính năng ghi điểm, như các tính năng ghi điểm vào hồ sơ, các tính năng ghi điểm truy vấn, và các tính năng ghi điểm truy vấn-hồ sơ. Ví dụ các tính năng ghi điểm vào hồ sơ có thể dựa trên các phép đo được kết hợp với hồ sơ, như tần suất hồ sơ được truy hồi trong các tìm kiếm và tần suất các liên kết được tạo ra dựa trên hồ sơ được chọn bởi người dùng. Các tính năng ghi điểm truy vấn có thể bao gồm, nhưng không bị giới hạn với, số lượng từ trong truy vấn tìm kiếm, độ phổ biến của truy vấn tìm kiếm, và tần suất như mong muốn của các từ trong truy vấn tìm kiếm. Các tính năng ghi điểm truy vấn-hồ sơ có thể bao gồm các tham số chỉ báo các thuật ngữ của truy vấn tìm kiếm thích hợp với các thuật ngữ của hồ sơ được chỉ báo với ID tương ứng như thế nào.

Môđun xử lý tập hợp 912 có thể bao gồm một hoặc nhiều mô hình học máy (như mô hình học được giám sát) được tạo cấu hình để nhận một hoặc nhiều tính năng ghi điểm. Một hoặc nhiều mô hình học máy có thể tạo ra các điểm cuối cùng dựa trên ít nhất một trong số các tính năng ghi điểm hồ sơ, các tính năng ghi điểm truy vấn, và các tính năng ghi điểm truy vấn-hồ sơ.

Ví dụ, môđun xử lý tập hợp 912 có thể tạo cặp truy vấn tìm kiếm với mỗi trạng thái của app ID và tính vector của các tính năng cho mỗi cặp {câu hỏi, ID}. Vector của các tính năng có thể bao gồm một hoặc nhiều tính năng ghi điểm vào hồ sơ, một hoặc nhiều tính năng ghi điểm truy vấn, và một hoặc nhiều tính năng ghi điểm truy vấn-hồ sơ. Theo một số phương án thực hiện, môđun xử lý tập hợp 912

chuẩn hóa các tính năng ghi điểm trong vector đặc trưng. Môđun xử lý tập hợp 912 có thể thiết lập các tính năng không phù hợp là giá trị rỗng hoặc không.

Môđun xử lý tập hợp 912 có thể sau đó nhập vector đặc trưng cho một trong số ID trạng thái của app vào mô hình hồi quy đọc được bằng máy để tính điểm cuối cùng cho ID. Trong một số ví dụ, mô hình hồi quy đọc được bằng máy có thể bao gồm tập hợp của các cây quyết định (như các cây quyết định tăng cường độ đọc). Ngoài ra hoặc theo cách khác, mô hình hồi quy đọc được bằng máy có thể bao gồm công thức xác suất logitic. Theo một số phương án thực hiện, tác vụ được học bằng máy có thể được tạo khung làm tác vụ học bán giám sát, trong đó số ít dữ liệu huấn luyện được gắn nhãn số điểm được ghi bởi người và phần còn lại được sử dụng mà không dẫn nhãn bởi người.

Mô hình được học bằng máy kết xuất điểm cuối cùng của ID. Môđun xử lý tập hợp 912 có thể tính các điểm cuối cùng đối với mỗi trong số các ID mà môđun xử lý tập hợp 912 nhận. Môđun xử lý tập hợp 912 kết hợp các điểm cuối cùng với các ID tương ứng và kết xuất các ID ghi điểm liên quan nhất.

Môđun tạo các kết quả 924 có thể chọn các cơ chế truy cập riêng từ các hồ sơ ứng dụng và các hồ sơ trạng thái app được chọn bởi môđun xử lý tập hợp 912. Môđun tạo các kết quả 924 sau đó chuẩn bị tập hợp kết quả để trả lại các thiết bị người dùng. Mặc dù được gọi là “các kết quả trạng thái app” ở đây, một số các cơ chế truy cập có thể tương ứng với trạng thái mặc định (như trang chủ) của app—điều này có thể là trường hợp đặc biệt của hồ sơ trạng thái app hoặc có thể là hồ sơ app.

Môđun tạo các kết quả 924 có thể chọn cơ chế truy cập đối với hồ sơ trạng thái app dựa trên app có được cài đặt trên thiết bị hay không. Nếu app được cài đặt, cơ chế truy cập mở app trực tiếp cho trạng thái được xác định được chọn. Trong lúc đó, nếu app không được cài đặt, cơ chế truy cập được chọn trước tiên tải xuống và cài đặt app, như thông qua tập lệnh, trước khi mở app đến trạng thái được xác định. Mở app ở trạng thái được xác định có thể bao gồm một lệnh hoặc cấu trúc dữ liệu (như dự định trong hệ điều hành Android) để dẫn động trực tiếp trạng thái xác định. Đối với các app khác, tập lệnh hoặc chuỗi khác có thể được sử dụng để mở app ở

trạng thái nhất định (như trạng thái chủ, hoặc trạng thái mặc định) và sau đó điều hướng đến trạng thái xác định.

Môđun tạo các kết quả 924 có thể tạo ra hoặc sửa đổi các cơ chế truy cập dựa trên tính đồng nhất của hệ điều hành và phiên bản cho thiết bị người dùng mà các kết quả được truyền. Ví dụ, tập lệnh để tải xuống, cài đặt, mở, và điều hướng đến trạng thái được chỉ định có thể được tạo ra hoàn toàn đối với hệ điều hành riêng bởi môđun tạo các kết quả 924.

Nếu môđun tạo các kết quả 924 xác định là không có cơ chế nào trong số các cơ chế truy cập riêng có thể tương thích với các thiết bị người dùng, môđun tìm kiếm 900 có thể gửi cơ chế truy cập web đến thiết bị người dùng. Nếu không có cơ chế truy cập web có sẵn, hoặc sẽ không tương thích với các thiết bị người dùng vì một số lý do (ví dụ, nếu cơ chế truy cập web dựa vào ngôn ngữ lập trình JAVA, không được cài đặt trên thiết bị người dùng), môđun tạo các kết quả 924 có thể lược bỏ kết quả này.

Điều khiển bên máy chủ

Trên Fig.10, hoạt động của các thành phần bên máy chủ có thể được thực hiện trong một thiết bị hoặc có thể được thực hiện qua cổng nhà phát triển 208, hệ thống phân tích ngoại tuyến 280, máy chủ dữ liệu 284, và hệ thống tìm kiếm 288 trên Fig.2. Trong các phương án thực hiện khác nhau, cổng nhà phát triển 208, máy chủ dữ liệu 284, và hệ thống tìm kiếm 288 có thể dưới sự điều khiển của cùng thực thể. Hệ thống phân tích ngoại tuyến 280 có thể sử dụng người vận hành để hỗ trợ sự phân tích tĩnh và/hoặc sự phân tích động để đảm bảo là các tham số chính xác và hoàn thiện được trích xuất từ mỗi app.

Điều khiển bắt đầu tại 1002, trong đó thư viện định tuyến được tạo ra khả dụng cho các nhà phát triển của app. Theo thời gian, thư viện định tuyến có thể được cập nhật, và phiên bản được cập nhật nhất có thể là duy nhất khả dụng cho các nhà phát triển của app. Tại 1004, nếu yêu cầu cho tập hợp liên kết được nhận, điều khiển chuyển đến 1006; mặc khác, điều khiển chuyển đến 1008. Tại 1006, điều khiển xác định tập hợp con của các tập hợp liên kết được thay đổi vì yêu cầu gần nhất được nhận từ bộ yêu cầu. Tập hợp con này có thể sau đó được cung cấp tại 1010 để mang dữ liệu cập nhật tại nguồn yêu cầu. Điều khiển sau đó trở lại 1004.

Tại 1008, nếu việc xử lý ngoại tuyến của app được khởi động, điều khiển chuyển đến 1012; mặc khác, điều khiển chuyển đến 1016. Tại 1016, nếu truy vấn tìm kiếm được nhận từ hệ thống tìm kiếm, điều khiển chuyển đến 1020; mặc khác, điều khiển trở lại đến 1004. Tại 1020, điều khiển xác định tập hợp xem xét của các kết quả tương ứng với truy vấn tìm kiếm. Tập hợp xem xét này có thể bao gồm các app mở cho truy vấn tìm kiếm cũng như các trạng thái riêng (các trạng thái ngầm) của các app liên quan đến truy vấn tìm kiếm.

Tại 1024, điều khiển ghi điểm các phần tử trong tập hợp xem xét dựa trên chúng thích hợp với dự định được hiểu của truy vấn tìm kiếm như thế nào. Các kết quả được ghi điểm có thể sau đó được xếp hạng từ liên quan nhiều nhất đến liên quan ít nhất. Tại 1028, các kết quả được xếp hạng cao nhất được định dạng là các thẻ hình ngầm được kết hợp với các liên kết ngầm đến các trạng thái riêng trong các kết quả. Tại 1032, điều khiển trở lại các thẻ hình ngầm đến bộ yêu cầu của truy vấn.

Các thẻ hình ngầm không thể được kết xuất hoàn toàn, và thay vì bao gồm các hình ảnh, văn bản, và các lệnh về cách để kết xuất các thẻ hình ngầm đối với các kích thước màn hình cụ thể, các định hướng, và các yêu cầu khác để yêu cầu ứng dụng hoặc hệ điều hành. Đối với các app trong đó các liên kết ngầm được phục vụ bởi thư viện định tuyến theo các nguyên lý của sáng chế, cơ chế truy cập được trở lại cùng với kết quả tìm kiếm tương ứng có thể bao gồm URI với cấu trúc dữ liệu được mã hóa.

Cấu trúc dữ liệu được mã hóa có thể bao gồm tập hợp liên kết cần thiết để dẫn ra trạng thái riêng từ trong app. URI, là chuỗi, bao gồm phiên bản nối tiếp của cấu trúc dữ liệu đó và được đặt ở trước sơ đồ. Sơ đồ, như “portal-appa://”, sẽ làm URI được chuyển tiếp đến và được nhận dạng bởi thư viện định tuyến của app. Điều khiển sau đó trở lại đến 1004.

Tại 1012, điều khiển chạy App A, như trong bộ mô phỏng. Trong khi App A chạy, điều khiển giám sát các sự kiện UI do việc sử dụng của người vận hành của App A. Tại 1036, điều khiển cho phép người vận hành tương tác với App A để đạt được trạng thái ngầm quan tâm, ghi mỗi tương tác UI để tạo tập hợp liên kết. Theo các phương án thực hiện khác nhau, điều khiển có thể giám sát xem App A mất bao lâu để đáp lại các sự kiện UI nhất định. Các thời gian phản hồi, hoặc các độ trễ dựa

trên các thời gian phản hồi, có thể được mã hóa trong tập hợp liên kết cùng với các sự kiện UI tương ứng. Ví dụ, một số hoạt động có thể yêu cầu App A để thu dữ liệu từ máy chủ, gây ra độ trễ truyền thông mạng. Khi phát lại các sự kiện UI trong tập hợp liên kết, độ trễ dài hơn sau sự kiện UI dẫn đến truy cập mạng trước khi sự kiện UI tiếp theo được phát lại. Tại 1040, nếu có thêm các trạng thái ngầm được quan tâm, điều khiển trở lại đến 1036; mặc khác, điều khiển tiếp tục tại 1044.

Tại 1044, điều khiển xác định các trạng thái song song với các trạng thái được nhận dạng bởi người vận hành. Ví dụ, nếu người vận hành chọn liên kết từ danh sách của các mục tìm kiếm tương tự, và liên kết được chọn dẫn đến trạng thái quan tâm, việc điều khiển có thể giả sử rằng các mục khác trong danh sách cũng được quan tâm. Các trạng thái song song này, và các tập hợp liên kết được sử dụng để đạt được chúng (trong danh sách làm ví dụ, các tập hợp liên kết sẽ chỉ khác nhau trong sự kiện UI cuối cùng), được thêm vào danh sách của các trạng thái có thể liên kết ngầm.

Theo các phương án thực hiện khác nhau, App A có thể được thi hành sử dụng thuật toán thu thập để đạt được một vài hoặc tất cả các trạng thái ngầm của App A. Đối với thông tin bổ sung liên quan đến việc thu thập, xem Đơn sáng chế Mỹ số 14/849,540, nộp ngày 09/09/2015, có tiêu đề “Unguided Application Crawling Structure”, với tác giả sáng chế thứ nhất tên là Kalyan Desineni, và có số hồ sơ 17100-000006-US-01. Toàn bộ phần bộc lộ của đơn này được đưa vào bằng cách viện dẫn.

Tại 1048, điều khiển đề xuất danh sách của các trạng thái có thể liên kết ngầm đến hệ thống tìm kiếm để lập chỉ mục. Tại 1052, đối với mỗi trạng thái có thể liên kết ngầm, điều khiển đề xuất cấu trúc dữ liệu chứa tập hợp liên kết đến máy chủ dữ liệu được sử dụng trong URI làm cơ chế truy cập để truy cập trạng thái ngầm. Sau đó điều khiển trở lại đến 1004.

Tại 1012, người vận hành tương tác với bản sao của app, như trong bộ mô phỏng, để điều hướng đến các trạng thái quan tâm. Theo một số phương án thực hiện, người vận hành điều hướng đến các trạng thái quan tâm được chỉ định bởi nhà phát triển; theo các phương án thực hiện khác, tự nhà phát triển điều hướng đến các

trạng thái quan tâm. Các sự kiện giao diện người dùng được yêu cầu để đạt các trạng thái khác nhau được ghi như được mô tả trên Fig.12.

Điều khiển tiếp tục tại 1040, trong đó điều khiển có thể ngoại suy các trạng thái bổ sung liên quan từ các trạng thái quan tâm được nhận dạng bởi người vận hành. Hoạt động làm ví dụ để ngoại suy được thể hiện trên Fig.13. Tại 1012, trong khi người vận hành nhận dạng các trạng thái quan tâm, người vận hành cũng có thể đánh dấu các trạng thái làm các hoạt động được đánh dấu và có thể nhận dạng dữ liệu trong các trạng thái dưới dạng các đối tượng dữ liệu để lập chỉ mục.

Điều khiển tiếp tục tại 1056, trong đó các tham số cấu hình được thiết lập bởi người vận hành được truyền đến máy chủ dữ liệu. Ví dụ, các tham số cấu hình có thể là các quy tắc nghiệp vụ xác định các trạng thái nào cần được liên kết ngầm các trạng thái nào cần được đánh dấu làm các hoạt động, và dữ liệu nào cần được lập chỉ mục.

Phân tích ngoại tuyến

Trên Fig.11, việc thực thi làm ví dụ của hệ thống phân tích ngoại tuyến 280 bao gồm hệ thống tạo dẫn hướng 1100 cho phép người vận hành tương tác với app quan tâm (như App A). Hệ thống tạo dẫn hướng 1100 cho phép người vận hành xác định các trạng thái nào của app cần có thể liên kết ngầm và có thể sử dụng đầu vào từ người vận hành để xác định cách để đạt được các trạng thái ngầm của chúng. Theo các phương án thực hiện khác nhau, người vận hành là người quản lý của hệ thống phân tích ngoại tuyến 280, tác động đến các thủ tục hoạt động chuẩn cho việc xử lý ngoại tuyến của app hoặc trên các lệnh từ nhà phát triển của app. Ngoài ra hoặc theo các khác, nhân viên của nhà phát triển của app có thể đóng vai trò là người vận hành. Nhân viên này có thể xác định các trạng thái nào hoặc các kiểu trạng thái nào được tạo ra truy cập được dưới dạng các trạng thái ngầm.

Sau khi phân tích ngoại tuyến ban đầu, đại diện có thể sử dụng cổng nhà phát triển 208 để yêu cầu các trạng thái bổ sung của app được thêm vào làm các trạng thái ngầm và yêu cầu rằng các trạng thái được liên kết ngầm tồn tại bị loại bỏ. Điều này có thể được thực hiện thông qua cổng nhà phát triển 208 mà không cần nhân viên sửa đổi mã của app hoặc thậm chí yêu cầu hỗ trợ từ nhà phát triển phần mềm của app. Theo cách khác, nhân viên marketing hoặc các thiết kế trải nghiệm người

dùng chính họ có thể điều khiển phạm vi của liên kết ngầm trong app, thậm chí không xuất bản phiên bản mới của app. Đối với các trạng thái bổ sung, hệ thống phân tích ngoại tuyến 280 sẽ tái xử lý app để xác định các tập hợp liên kết đến các trạng thái bổ sung.

Đối với app được xử lý bởi hệ thống phân tích ngoại tuyến 280, người vận hành điều khiển bản sao của app quan tâm chạy trong bộ mô phỏng 1104. Theo các phương án thực hiện khác nhau, app quan tâm có thể được cài đặt trên thiết bị vật lý hoặc được chạy trong môi trường mô phỏng. Theo các phương án thực hiện khác nhau, bộ mô phỏng 1104 có thể được tạo đối tượng tại bộ vận hành máy chủ đám mây có thể cung cấp các thiết bị máy tính trong đó chạy bộ mô phỏng/mã mô phỏng hoặc có thể trực tiếp cung cấp các bộ mô phỏng cho một hoặc nhiều hệ điều hành thiết bị di động. Để thêm thông tin, xem Đơn sáng chế Mỹ số 14/868,294, nộp ngày 28/09/2015, có tiêu đề “Operator-Guided Application Crawling Structure”, với tác giả sáng chế thứ nhất tên là Kalyan Desineni, và có số hồ sơ là 17100-000006-US-02. Toàn bộ phần bộc lộ của đơn này được đưa vào bằng cách viện dẫn.

Theo một số phương án thực hiện, thiết bị vật lý chạy hệ điều hành có thể được sử dụng thay cho bộ mô phỏng 1104, như đối với hệ điều hành không có bộ mô phỏng thích hợp. Thiết bị vật lý có thể được nối với hệ thống tạo dẫn hướng 1100 sử dụng giao diện không dây hoặc có dây, như USB (universal serial bus – bus tiếp nối đa năng). Ứng dụng điều khiển cấp độ gốc có thể được cài đặt trên thiết bị vật lý để theo dõi đầu vào người dùng. Cài đặt ứng dụng cấp độ gốc có thể yêu cầu các giới hạn an toàn mạch rẽ của phần sụn hoặc hệ điều hành liên quan đến các đặc quyền của các app được cài đặt và có thể yêu cầu là thiết bị bị phá rào.

Sự phá rào đề cập đến việc rẽ nhánh hoặc loại bỏ các giới hạn phần mềm khác nhau được ấn định bởi hệ điều hành và/hoặc phần sụn thiết bị, và có thể được hoàn thành sử dụng kỹ thuật neo thang đặc quyền. Thiết bị phá rào có thể còn được cải biến với các thiết lập được cài đặt trước và các thiết lập cụ thể cho phép các kỹ thuật theo sáng chế, như ghi lại các tương tác UI. Bộ mô phỏng 1104 gần như bị phá rào sẵn. Tuy nhiên, bộ mô phỏng 1104 cũng có thể được cải biến để cho phép điều khiển tốt hơn, như để cho phép ghi các tương tác UI dễ dàng hơn.

Tương tác của người vận hành với app có thể được ghi để tạo dẫn hướng chỉ báo cách mà trạng thái đạt được. Việc dẫn hướng này xác định tập hợp liên kết được sử dụng để đạt trạng thái cuối cùng, và các tập con của việc dẫn hướng xác định các tập hợp liên kết đối với các trạng thái trung gian. Việc dẫn hướng cũng có thể được sử dụng để suy ra các trạng thái quan tâm khác và xác định các tập hợp liên kết dẫn đến các trạng thái đó. Mỗi sự kiện UI có thể được liên kết với thành phần UI cụ thể, được nhận dạng theo chuyên mục định trước với ID duy nhất.

Ví dụ, ID duy nhất có thể được lệnh bằng cách các phần tử UI được tạo ra theo cách lập trình khi kết xuất hình. Theo các phương án thực hiện khác, ID duy nhất có thể được lệnh theo cách các phần tử UI được xem thực sự, với các phần tử UI được đánh số theo thứ tự tăng dần từ trái sang phải và từ trên xuống dưới. Theo các phương án thực hiện khác, các tọa độ x-y có thể được sử dụng để nhận dạng vị trí của tương tác người dùng, hoặc các biên của thành phần UI được điều khiển, hoặc trung tâm của thành phần UI được điều khiển ngay cả nếu thao tác của người vận hành bị lệch tâm.

Theo các phương án thực hiện khác nhau, sự phân tích tĩnh có thể được sử dụng để phân tích các phần tử UI của các trạng thái khác nhau của app. Sự phân tích tĩnh liên quan đến phân tích mã của app không theo dõi app được chạy. Theo một số phương án thực hiện, bộ phân tích tĩnh (không được thể hiện) có thể nhận dạng các phần tử UI của mỗi hình và chỉ định các mã nhận dạng ID cho mỗi thành phần UI trong hình. Sau đó, sự phân tích động sẽ có danh mục định trước khi đề cập đến các tương tác người dùng với các phần tử UI bất kỳ. Ví dụ, sự phân tích tĩnh có thể được thực hiện với việc tách rời và gỡ lỗi được thực thi bởi phần mềm IDA Pro từ Hex-Rays SA.

Ngoài ra hoặc theo cách khác, bộ phân tích tĩnh có thể phân tích các phương pháp và các hình của app để xác định cách trang bị app tốt nhất. Với thông tin này, bộ phân tích động có thể ghép vào các phương pháp chính xác và nghe các sự kiện chính xác để theo dõi chính xác tương tác người dùng với app và loại bỏ các cuộc gọi có phương thức không liên quan đến tương tác người dùng. Để thêm thông tin về phân tích tĩnh, xem Đơn sáng chế Mỹ số 14/843,929, nộp ngày 02/09/2015, có tiêu đề “Static Analysis-Assisted Dynamic Application Crawling Structure” với tác

giả sáng chế thứ nhất tên là Kalyan Desineni, và có số hồ sơ là 17100-000006-US. Toàn bộ phần bộc lộ của đơn này được đưa vào bằng cách viện dẫn.

Trong khi app được chạy trong bộ mô phỏng 1104, hệ thống tạo dẫn hướng 1100 có thể giám sát hàng đợi tin nhắn của bộ mô phỏng 1104 (hoặc của hệ điều hành thực khi chạy trong bộ mô phỏng hoặc thiết bị vật lý), nghe tin nhắn chỉ báo thao tác người dùng của app. Thao tác thủ công này bao gồm, ví dụ, nhấp, nhấn, kẹp, hoặc vuốt giao diện người dùng.

Ngoài việc thu mã nhận dạng duy nhất cho mỗi thành phần UI ra mà người vận hành tương tác, hệ thống tạo dẫn hướng 1100 cũng có thể trích xuất mỗi loại thành phần theo các kiểu xác định được xác định bởi hệ điều hành, và còn trích xuất tính đồng nhất của bộ điều khiển hình liên quan.

Ví dụ, các phương pháp sau đây có thể được dẫn ra, bằng cách sử dụng phương pháp swizzling:

- (void)sendEvent:(UIEvent *)event method in the UIApplication class
- (void)sendAction:(SEL)action to:(id)target forEvent:(UIEvent *)event in UIControl class
- (void)viewDidAppear:(BOOL)animated for all the controllers that were loaded

Bộ xử lý có thể được triển khai để ghi lại mỗi kiểu thành phần UI. Theo một ví dụ, mã để ghi lại nút bấm có thể được triển khai như sau:

```
-(void)sendEvent:(UIEvent *)event
{
    for (UITouch *touch in event.allTouches)
    {
        if (touch.phase == UITouchPhaseBegan)
        {
            if ([touch.view isKindOfClass:[UIButton class]]) {
                NSLog(@"UIButton:id:{%@}", [((UIButton *)touch.view)
currentTitle]);
            }
        }
    }
}
```

```

    }
}

```

Các kiểu phần tử UI khác có thể được ghi bởi mã bộ xử lý tương tự. Theo các phương án thực hiện khác nhau, sau khi dẫn hàm objc_msgSend, bộ theo dõi dẫn hướng 1108 có thể nghe lớp appDelegate và phương pháp (BOOL) application:(id)application ~didFinishLaunchingWithOptions: ~(NSDictionary *)launchOptions. Ngoài ra, bộ theo dõi dẫn hướng 1108 có thể tạo ra một cách động cho người nghe đối với các bộ điều khiển.

Thay vì cố gắng phát hiện mọi trạng thái của app thông qua việc thu thập không được dẫn hướng của app, người vận hành có thể hỗ trợ sự tập trung phân tích trên các trạng thái mà các liên kết ngầm mong muốn—nói chung, các trạng thái quan tâm hoặc quan trọng nhất. Người vận hành có thể bắt đầu với trạng thái chủ của app và tiến triển đến một trạng thái quan tâm đối với mỗi hạng mục trạng thái mà liên kết ngầm mong muốn. Hệ thống phân tích ngoại tuyến 280 có thể sau đó ngoại suy để tìm các thao tác tương tự/song song, mỗi trong số đó có thể tương ứng với trạng thái quan tâm khác. Các trạng thái song song này có thể được thêm vào danh sách trạng thái xác định các trạng thái nào sẽ có các tập hợp liên kết tương ứng được lưu trữ.

Ví dụ, nếu app bao gồm thông tin về các nhà hàng, người vận hành có thể duyệt trạng thái mà liệt kê các nhà hàng, và sau đó chọn một trong số các nhà hàng. Người vận hành có thể dừng lại sau khi tìm trạng thái mà thể hiện chi tiết về nhà hàng thứ nhất. Dựa trên chuỗi các thao tác được thực hiện bởi người vận hành trong khi duyệt, hệ thống phân tích ngoại tuyến 280 có thể tìm các trạng thái chi tiết nhà hàng khác có thể đạt được theo cách này.

Ví dụ, sau khi chọn mục nhà hàng với sơ đồ nhất định và các tính chất nhất định (ví dụ hình ảnh có tính chất X gần hộp văn bản với tính chất Y) từ danh sách, hệ thống phân tích ngoại tuyến 280 có thể phát hiện nhiều mục nhà hàng bổ sung (có cùng sơ đồ và/hoặc các tính chất) trong hình, mà người vận hành có thể chọn. Sau đó dự đoán là việc chọn danh sách nhà hàng khác sẽ dẫn đến việc tìm kiếm các trạng thái thông tin nhà hàng bổ sung. Đối với thông tin bổ sung về phép ngoại suy, xem Đơn sáng chế Mỹ số 14/869,127, nộp ngày 29/09/2015, có tiêu đề “State

Extrapolation for Automated and Semi-Automated Crawling Architecture”, với tác giả sáng chế thứ nhất là Kalyan Desineni, và có số hồ sơ 17100-000006-US-03. Toàn bộ phần bộc lộ của đơn này được đưa vào bằng cách viện dẫn.

Trong hệ thống tạo dẫn hướng 1100, bộ theo dõi dẫn hướng 1108 ghi lại tương tác người vận hành với app trong bộ mô phỏng 1104 để tạo các dẫn hướng được xác định bởi người vận hành. Ví dụ, sự dẫn hướng có thể bao gồm mỗi tương tác giao diện người dùng được thực hiện bởi người vận hành, bắt đầu ở trạng thái chủ (hoặc, mặc định) của app. Trong các phương án thực hiện khác nhau, bộ theo dõi trạng thái đích 1112 có thể cho phép người vận hành báo cáo trạng thái thấy được hiện tại trong bộ mô phỏng 1104 là trạng thái quan tâm. Ví dụ, bộ theo dõi trạng thái đích 1112 có thể cung cấp thành phần giao diện người dùng (như nút) trong bộ mô phỏng 1104 hoặc là một phần của phần mềm qua đó người vận hành điều khiển bộ mô phỏng 1104.

Đối với mỗi trạng thái quan tâm, bộ điều khiển trích xuất liên kết 1116 tạo ra tập hợp liên kết xác định chuỗi các sự kiện UI dẫn đến trạng thái quan tâm. Bộ điều khiển trích xuất liên kết 1116 nhận biết các trạng thái liên quan đến người vận hành, rõ ràng (như qua bộ theo dõi trạng thái đích 1112) hoặc hoàn toàn (như qua bộ theo dõi dẫn hướng 1108). Bộ điều khiển trích xuất liên kết 1116 có thể cố gắng để nhận dạng các trạng thái tương tự—ví dụ, các trạng thái đạt được sử dụng các phần tử UI (giao diện người dùng) tương tự.

Các trạng thái đích có thể được nhận dạng rõ ràng bởi người vận hành sử dụng bộ theo dõi trạng thái đích 1112. Nếu các trạng thái đích không được xác định bởi người vận hành, bộ điều khiển trích xuất liên kết 1116 có thể giả sử rằng trạng thái cuối cùng đạt được khi người vận hành tạo dẫn hướng là trạng thái đích. Theo cách khác, bộ điều khiển trích xuất liên kết 1116 có thể giả sử rằng mọi trạng thái mà người vận hành được điều hướng cần là trạng thái đích. Thủ tục giải sao chép có thể được sử dụng để phát hiện khi người vận hành duyệt trong vòng lặp, do đó ngăn chặn việc ghi các dẫn hướng sao chép và/hoặc đường đi không ngắn nhất. Việc giải sao chép có thể ít cần thiết nếu người vận hành đánh dấu rõ ràng các trạng thái đích.

Bộ điều khiển trích xuất liên kết 1116 vận hành bộ mô phỏng 1120, như được mô tả ở trên, có thể thay vì là bộ mô phỏng hoặc thiết bị vật lý. Về mặt tỷ lệ,

nhiều bộ mô phỏng, bộ mô phỏng, và/hoặc thiết bị vật lý có thể được điều khiển bởi bộ điều khiển trích xuất liên kết 1116 để phân tích cùng ứng dụng hoặc các ứng dụng khác. Theo một ví dụ, ngân hàng điện thoại thông minh vật lý tất cả có thể được nối qua USB đến thẻ giao diện là được điều khiển bởi bộ điều khiển trích xuất liên kết 1116. Để dễ minh họa, một bộ mô phỏng (bộ mô phỏng 1120) được thể hiện trên Fig.4. Trong các phương án thực hiện khác nhau, bộ mô phỏng 1120 và bộ mô phỏng 1104 có thể là một bộ mô phỏng được chia sẻ bởi hệ thống tạo dẫn hướng 1100 và bộ điều khiển trích xuất liên kết 1116.

Ứng dụng quan tâm được chạy trong bộ mô phỏng 1120. Theo các phương án thực hiện khác nhau, app được chạy bởi bộ mô phỏng 1120 có thể được bổ sung với thư viện định tuyến, như được mô tả trên Fig.3A hoặc Fig.3B. Với thư viện định tuyến, bộ điều khiển trích xuất liên kết 1116 có thể khiến app trong bộ mô phỏng 1120 phát lại các sự kiện UI để theo dõi dẫn hướng và nhận dạng các trạng thái quan tâm bổ sung. Trong các phương án thực hiện khác, vì app trong bộ mô phỏng 1120 không cần phiên bản của app được phân phối công khai, khả năng truy cập hoặc khung tự động có thể được sử dụng để thực hiện điều khiển UI. Thư viện định tuyến có thể được dự trữ cho phiên bản của app tức là được phân phối công khai.

Bộ điều khiển trích xuất liên kết 1116 nhận dạng các trạng thái quan tâm tương ứng với mỗi trong số các dẫn hướng được xác định bởi bộ theo dõi dẫn hướng 1108. Để đạt được trạng thái trong app, bộ điều khiển trích xuất liên kết 1116 gửi đường ngắn nhất được xác định bởi bộ theo dõi dẫn hướng 1108 đến bộ mô phỏng 1120 được phát lại. Bộ điều khiển trích xuất liên kết 1116 nhận dạng các trạng thái quan tâm tương ứng với mỗi trong số các dẫn hướng được xác định bởi bộ theo dõi dẫn hướng 1108.

Theo các phương án thực hiện khác nhau, hệ thống phân tích ngoại tuyến 280 có thể bao gồm bộ tìm kiếm (không được thể hiện) để trích xuất nội dung từ mỗi trạng thái quan tâm. Nội dung có thể bao gồm văn bản, các hình ảnh, và siêu dữ liệu, như gợi ý vị trí, định dạng, giao diện, v.v. Nội dung này có thể được sử dụng bởi hệ thống tìm kiếm 288 trên Fig.2 để xác định các trạng thái ngầm nào của app liên quan đến truy vấn tìm kiếm. Trong các phương án thực hiện khác, hệ thống tìm kiếm 288 có thể thu thập và tìm kiếm app một cách tách biệt, nhưng có thể sử dụng

các tập hợp liên kết từ hệ thống phân tích ngoại tuyến 280 để điều hướng đến các trạng thái khác nhau.

Nếu có các trường UI trong đó văn bản cần được nhập, người vận hành có thể nhận dạng hệ thống tạo dẫn hướng 1100 và các kiểu đầu vào văn bản nào cần được nhập. Các kiểu này có thể là, ví dụ, tên thành phố, tên món ăn, v.v. Bộ điều khiển trích xuất liên kết 1116 sau đó tham khảo cơ sở tri thức để nhận danh sách các giá trị khả dĩ của các loại này (đối với các thành phố, danh sách có thể bao gồm “Seattle,” “San Jose,” v.v.) và sau đó phát lại mỗi một trong số các giá trị này vào trường nhập vào văn bản.

Bộ điều khiển trích xuất liên kết 1116 có thể phát hiện là trạng thái quan tâm tiềm năng đã đạt được bằng cách theo dõi tập hợp liên kết khác. Bộ điều khiển trích xuất liên kết 1116 sau đó có thể chọn một trong số các tập hợp liên kết để kết hợp với trạng thái đó. Ví dụ, tập hợp liên kết có số lượng các sự kiện UI ít nhất có thể được chọn. Theo cách khác, tập hợp liên kết có các thay đổi hình ít nhất có thể được chọn, và tổng số lượng các sự kiện UI có thể được sử dụng làm hiệp phụ.

Để nhận dạng trạng thái nào đã đạt được, bộ điều khiển trích xuất liên kết 1116 có thể lưu trữ dấu vân tay đối với mỗi trạng thái cùng với tập hợp liên kết. Dấu vân tay này có thể là việc biểu diễn thu gọn của các thành phần của trạng thái. Ví dụ, việc biểu diễn của các đối tượng thấy được có thể được tạo ra, như bằng cách liệt kê các đối tượng trong cấu trúc dữ liệu XML và sau đó tính toán hàm băm toán học (như với MD5) của cấu trúc dữ liệu. Sau đó hàm băm là dấu vân tay, và nếu hàm băm của trạng thái mới khớp với hàm băm của trạng thái có sẵn, các trạng thái được xem xét là giống nhau.

Khi bộ điều khiển trích xuất liên kết 1116 cài đặt tập hợp các trạng thái quan tâm, bộ điều khiển trích xuất liên kết 1116 có thể lưu trữ mã nhận dạng của trạng thái trong kho dữ liệu cấu hình 1124. Đối với mỗi trạng thái quan tâm, bộ điều khiển trích xuất liên kết 1116 xác định tập hợp liên kết để đạt trạng thái đó và lưu trữ tập hợp liên kết trong kho dữ liệu cấu hình 1124. Ngoài ra, nội dung hoặc dấu vân tay từ các trạng thái được lưu trữ trong kho dữ liệu cấu hình 1124. Dữ liệu từ kho dữ liệu cấu hình 1124 có thể được cung cấp cho hệ thống tìm kiếm 288 trên Fig.2.

Việc tạo tập hợp liên kết

Trên Fig.12, hoạt động làm ví dụ của việc tạo tập hợp liên kết bởi người vận hành bắt đầu tại 1204, trong đó điều khiển mở app đối tượng (subject app) trong bộ mô phỏng đến trạng thái chủ của app. Tại 1208, điều khiển bắt đầu theo dõi tương tác người dùng với app, bao gồm theo dõi các thành phần giao diện người dùng mà người vận hành tương tác.

Tại 1212, nếu người vận hành đánh dấu trạng thái hiện tại làm đích liên kết ngầm, điều khiển chuyển đến 1216; mặc khác, điều khiển chuyển đến 1220. Tại 1216, trạng thái được thêm vào danh sách các trạng thái đích và điều khiển tiếp tục tại 1224. Tại 1224, điều khiển lưu trữ tương tác người vận hành dẫn đến trạng thái hiện tại làm tập hợp liên kết. Điều khiển tiếp tục tại 1220. Tại 1220, nếu người vận hành đánh dấu trạng thái là hoạt động để nhằm đánh dấu, điều khiển chuyển đến 1228; mặc khác, điều khiển chuyển đến 1232.

Tại 1228, điều khiển thêm trạng thái vào các quy tắc đối với hoạt động đánh dấu để khi trạng thái này được bắt gặp bởi người dùng, hoạt động tương ứng với trạng thái được đánh dấu bằng cách gửi đối tượng hoạt động đến hệ điều hành. Điều khiển tiếp tục tại 1236, trong đó tương tác người vận hành dẫn đến trạng thái hiện tại được lưu trữ làm tập hợp liên kết. Nếu tập hợp liên kết này được lưu trữ sẵn tại 1224 đối với trạng thái hiện tại, bản sao bổ sung của tập hợp liên kết có thể được loại bỏ khỏi việc lưu trữ. Điều khiển tiếp tục tại 1232.

Tại 1232, điều khiển tiếp tục theo dõi tương tác người vận hành với app. Tại 1240, điều khiển xác định xem liệu người vận hành có chỉ dẫn app để thiết lập lại trạng thái chủ của app hay không. Nếu vậy, điều khiển chuyển đến 1244; mặc khác, điều khiển trở lại đến 1212. Thiết lập lại trạng thái chủ có thể được hoàn thành với thành phần giao diện người dùng thuộc về bộ mô phỏng trong đó app đang hoạt động. Trong các giới hạn khác, chính app này có thể có cơ chế để trở lại trạng thái chủ và điều này cũng được nhận dạng bởi 1240. Bằng cách thiết lập lại trạng thái chủ, người vận hành chỉ báo tập hợp mới của các sự kiện giao diện người dùng cần được theo dõi dẫn đến trạng thái quan tâm khác.

Tại 1244, nếu trạng thái hiện tại không được đánh dấu sẵn là đích liên kết ngầm, điều khiển lưu trữ tương tác người vận hành dẫn đến trạng thái hiện tại là tập

hợp liên kết và thêm trạng thái hiện tại vào danh sách của các trạng thái đích. Điều khiển tiếp tục tại 1248, trong đó nếu người vận hành kết thúc, điều khiển kết thúc. Theo cách khác, điều khiển trở lại đến 1208.

Ngoại suy tập hợp liên kết

Tham chiếu đến Fig.13, người vận hành có thể thiết lập chuỗi các tập hợp liên kết nhưng không được gắn nhãn bất kỳ các trạng thái cụ thể nào làm các trạng thái đích. Ví dụ, theo một số phương án thực hiện, người vận hành không thể có mặt với lựa chọn để chỉ định các trạng thái đích. Tuy nhiên, các tập hợp liên kết được tạo bởi người vận hành thường bắt đầu ở trạng thái chủ và kết thúc ở trạng thái cuối (hoặc, cuối cùng). Trạng thái cuối cùng này có thể được giả sử là trạng thái quan tâm đối với liên kết ngầm. Theo cách khác, chỉ có các trạng thái đầu cuối được giả sử là các trạng thái đích. Theo các phương án thực hiện khác, mỗi trạng thái được bắt gặp cùng với tập hợp liên kết được giả sử là trạng thái đích.

Ví dụ, khi người vận hành tạo các tập hợp liên kết cho ứng dụng đánh giá nhà hàng, người vận hành có thể tạo một tập hợp liên kết bằng cách điều hướng từ trạng thái chủ đến trạng thái chứa thông tin và các đánh giá về nhà hàng. Điều này được giả sử là trạng thái đích mong muốn, và bộ điều khiển trích xuất liên kết cố gắng để tìm các trạng thái bổ sung mà có dữ liệu tương tự để tìm kiếm.

Điều khiển bắt đầu tại 1304, trong đó tập hợp liên kết thứ nhất từ danh sách của các tập hợp liên kết được tạo bởi người vận hành được chọn. Tại 1308, điều khiển mở app đến trạng thái chủ trong bộ mô phỏng. Tại 1312, điều khiển theo dõi tập hợp liên kết được chọn, dừng ở trạng thái ngay trước trạng thái cuối cùng. Tại 1316, điều khiển thực hiện ngoại suy để nhận dạng các trạng thái tương tự với trạng thái cuối cùng. Ví dụ, điều khiển nhận dạng các công cụ UI song song với công cụ UI, khi được theo sau theo tập hợp liên kết được chọn, sẽ đạt đến trạng thái cuối cùng.

Tại 1320, điều khiển thêm trạng thái cuối cùng, cũng như các trạng thái đạt được từ các công cụ UI được nhận dạng là song song, vào danh sách trạng thái. Mỗi trong số các trạng thái được thêm được chú thích với tập hợp liên kết dựa trên tập hợp liên kết được chọn. Tập hợp liên kết đối với các trạng thái ngoài trạng thái cuối cùng sẽ làm trệch đi khỏi tập hợp liên kết đối với trạng thái cuối cùng ở sự kiện UI

cuối cùng của mỗi tập hợp liên kết. Tại 1324, điều khiển xác định xem liệu các tập hợp liên kết bổ sung có mặt trong danh sách của các tập hợp liên kết hay không. Nếu vậy, điều khiển chuyển đến 1328; mặc khác, điều khiển kết thúc. Tại 1328, điều khiển chọn tập hợp liên kết tiếp theo từ danh sách của các tập hợp liên kết và trở lại đến 1308.

Kết luận

Mô tả trước chỉ làm ví dụ về bản chất và không theo cách dự định làm giới hạn sáng chế, ứng dụng, hoặc việc sử dụng của nó. Việc hiểu rộng của sáng chế có thể được triển khai trong các dạng khác nhau. Do đó, trong khi sáng chế bao gồm các ví dụ cụ thể, phạm vi thực của sáng chế không bị giới hạn vì việc sửa đổi khác sẽ trở nên rõ ràng nhờ nghiên cứu các hình vẽ, bản mô tả, và các yêu cầu bảo hộ sau đây. Cần hiểu rằng một hoặc nhiều bước trong phương pháp có thể được thực hiện theo thứ tự khác (hoặc đồng thời) không thay đổi các nguyên lý của sáng chế. Ngoài ra, mặc dù mỗi trong số các phương án được mô tả ở trên có các dấu hiệu nhất định, bất kỳ một hoặc nhiều dấu hiệu đó được mô tả tương ứng với phương án bất kỳ của sáng chế có thể được triển khai và/hoặc được kết hợp với các dấu hiệu của phương án bất kỳ trong số các phương án khác, ngay cả nếu kết hợp đó không được mô tả rõ ràng. Theo cách khác, các phương án được mô tả không loại trừ lẫn nhau, và các hoán vị của một hoặc nhiều phương án trong phạm vi bảo hộ của sáng chế.

Các mối tương quan không gian và chức năng giữa các phần tử (ví dụ, giữa các môđun) được mô tả sử dụng các thuật ngữ khác nhau, bao gồm “được nối,” “được gắn,” “được ghép nối”, và “được ghép”. Trừ khi được mô tả rõ ràng là “trực tiếp” khi tương quan giữa các phần tử thứ nhất và thứ hai được mô tả trong phần bộc lộ ở trên, tương quan gồm tương quan trực tiếp trong đó không có các phần tử xen kẽ khác có mặt giữa các phần tử thứ nhất và thứ hai, và cũng tương quan gián tiếp trong đó một hoặc nhiều phần tử xen kẽ có mặt (không gian hoặc chức năng) giữa các phần tử thứ nhất và thứ hai. Khi được sử dụng ở đây, các cụm từ ít nhất một trong số A, B, và C cần hiểu nghĩa là logic (A OR B OR C), sử dụng logic không loại trừ OR, và không được hiểu nghĩa là “ít nhất một trong số A, ít nhất một trong số B, và ít nhất một trong số C.”

Trong sáng chế này, bao gồm các định nghĩa sau đây, thuật ngữ ‘môđun’ hoặc thuật ngữ ‘bộ điều khiển’ có thể được thay thế với thuật ngữ ‘mạch’. Thuật ngữ ‘môđun’ có thể đề cập đến, là một phần của, hoặc bao gồm phần cứng bộ xử lý (được chia sẻ, dành riêng, hoặc nhóm) chạy mã và phần cứng bộ nhớ (được chia sẻ, dành riêng, hoặc nhóm) để lưu trữ mã được chạy bởi phần cứng bộ xử lý.

Môđun có thể bao gồm một hoặc nhiều mạch giao diện. Theo một số ví dụ, các mạch giao diện có thể bao gồm các giao diện có dây hoặc không dây được nối với mạng cục bộ (local area network - LAN), Internet, mạng toàn cục (wide area network - WAN), hoặc các kết hợp của chúng. Chức năng của môđun đã cho bất kỳ theo sáng chế có thể được phân phối trong số nhiều môđun được nối qua các mạch giao diện. Ví dụ, nhiều môđun có thể cho phép việc cân bằng tải. Theo ví dụ khác, môđun máy chủ (cũng được biết là từ xa, hoặc đám mây) có thể hoàn thiện một số chức năng thay mặt cho môđun máy khách.

Thuật ngữ mã, như được sử dụng ở trên, có thể bao gồm phần mềm, phần sụn, và/hoặc vi mã, và có thể đề cập đến các chương trình, đoạn chương trình, chức năng, phân loại, cấu trúc dữ liệu, và/hoặc các đối tượng. Phần cứng bộ xử lý dùng chung gồm một bộ vi xử lý chạy một vài hoặc tất cả mã từ nhiều môđun. Phần cứng bộ xử lý nhóm gồm bộ vi xử lý, trong kết hợp với các bộ vi xử lý bổ sung, chạy một vài hoặc tất cả mã từ một hoặc nhiều môđun. Tham chiếu đến nhiều bộ vi xử lý gồm nhiều bộ vi xử lý trên các khuôn riêng biệt, nhiều bộ vi xử lý trên một khuôn, nhiều lõi trong số một bộ vi xử lý, nhiều chuỗi của một bộ vi xử lý, hoặc kết hợp các thành phần trên.

Phần cứng bộ nhớ dùng chung gồm một thiết bị bộ nhớ lưu trữ một vài hoặc tất cả mã từ nhiều môđun. Phần cứng bộ nhớ nhóm gồm thiết bị bộ nhớ, kết hợp với các thiết bị bộ nhớ khác, lưu trữ một vài hoặc tất cả mã từ một hoặc nhiều môđun.

Thuật ngữ phần cứng bộ nhớ là tập con của thuật ngữ vật ghi đọc được bằng máy tính. Thuật ngữ vật ghi đọc được bằng máy tính, như được sử dụng ở đây, không gồm các tín hiệu điện hoặc điện từ lan truyền qua vật ghi (như trên sóng mang); thuật ngữ vật ghi đọc được bằng máy tính do đó được xét là hữu hình và không tạm thời. Các ví dụ không giới hạn của vật ghi đọc được bằng máy tính không tạm thời là các thiết bị bộ nhớ không khả biến (như thiết bị bộ nhớ chớp,

thiết bị bộ nhớ chỉ đọc lập trình xóa được, hoặc thiết bị bộ nhớ chỉ đọc mặt nạ), các thiết bị bộ nhớ khả biến (như thiết bị bộ nhớ truy cập ngẫu nhiên tĩnh hoặc thiết bị bộ nhớ truy cập ngẫu nhiên động), vật ghi lưu trữ từ (như băng từ tương tự hoặc kỹ thuật số hoặc ổ đĩa cứng), và các vật ghi lưu trữ quang học (như CD, DVD, hoặc đĩa Blu-ray).

Các thiết bị và phương pháp được mô tả trong đơn sáng chế này có thể được triển khai một phần hoặc toàn phần với máy tính chuyên dụng được tạo ra bằng cách tạo cấu hình máy tính đa dụng để thực hiện một hoặc nhiều chức năng cụ thể được bao gồm trong các chương trình máy tính. Các khối chức năng và các phần tử lưu đồ được mô tả ở trên đóng vai trò là các đặc tả phần mềm, có thể được dịch sang là các chương trình máy tính với công việc thường trình của chuyên gia kỹ thuật hoặc lập trình viên.

Các chương trình máy tính bao gồm các lệnh chạy bằng bộ xử lý được lưu trữ trên ít nhất một vật ghi đọc được bằng máy tính không tạm thời. Các chương trình máy tính cũng có thể bao gồm hoặc dựa trên dữ liệu được lưu trữ. Các chương trình máy tính có thể gồm hệ thống đầu vào/đầu ra cơ bản (BIOS) tương tác với phần cứng của máy tính chuyên dụng, thiết bị điều khiển để tương tác với các thiết bị cụ thể của máy tính chuyên dụng, một hoặc nhiều hệ điều hành, các ứng dụng người dùng, các dịch vụ nền tảng, các ứng dụng nền tảng, v.v.

Các chương trình máy tính có thể bao gồm: (i) văn bản mô tả được phân tích, như HTML (Hypertext markup language – ngôn ngữ đánh dấu siêu văn bản) hoặc XML (extensible markup language – ngôn ngữ đánh dấu mở rộng), (ii) mã ngữ, (iii) mã đối tượng được tạo ra từ mã nguồn bởi bộ biên dịch, (iv) mã nguồn để chạy bởi bộ biên dịch, (v) mã nguồn cho biên dịch và chạy bởi các bộ biên dịch đúng thời hạn, v.v. Theo các ví dụ, mã nguồn có thể được viết sử dụng cú pháp từ các ngôn ngữ bao gồm C, C++, C#, Objective-C, Swift, Haskell, Go, SQL, R, Lisp, Java®, Fortran, Perl, Pascal, Curl, OCaml, Javascript®, HTML5 (Hypertext Markup Language 5th revision - ngôn ngữ đánh dấu siêu văn bản xét duyệt thứ 5), Ada, ASP (Active Server Pages – các trang máy chủ hoạt động), PHP (PHP: Hypertext Preprocessor), Scala, Eiffel, Smalltalk, Erlang, Ruby, Flash®, Visual Basic®, Lua, MATLAB, SIMULINK, và Python®.

Không có phần tử nào trong số các phần tử được nêu trong các yêu cầu bảo hộ nhằm vào thành phần chức năng bổ sung với nghĩa của 35 U.S.C. §112(f) trừ khi thành phần được nêu rõ ràng là sử dụng các cụm từ “các phương tiện để” hoặc, trong trường hợp của phương pháp nêu, sử dụng các cụm từ “hoạt động để” hoặc “bước để”.

YÊU CẦU BẢO HỘ

1. Hệ thống phát triển ứng dụng di động bao gồm:

công nhà phát triển được tạo cấu hình để:

nhận bản sao của ứng dụng thứ nhất từ nhà phát triển thứ nhất; và

cung cấp thư viện định tuyến cho nhà phát triển thứ nhất để sát nhập vào ứng dụng thứ nhất trước khi nhà phát triển thứ nhất phân phối ứng dụng bổ sung thông qua nền tảng phân phối số, trong đó thư viện định tuyến được sát nhập vào ứng dụng thứ nhất để tạo ứng dụng bổ sung; và

hệ thống phân tích ngoại tuyến được tạo cấu hình để:

phân tích ứng dụng thứ nhất để (i) xác định tập hợp của các hoạt động mà bộ xử lý trong ứng dụng thứ nhất được lập trình để khôi phục nhằm đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ và (ii) xác định các tham số cho mỗi hoạt động trong số tập hợp các hoạt động; và

tạo tập hợp của các liên kết, trong đó mỗi liên kết trong số tập hợp các liên kết tương ứng với hoạt động tương ứng trong số tập hợp của các hoạt động,

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, theo sau việc cài đặt ứng dụng bổ sung từ nền tảng phân phối số trên các thiết bị người dùng:

nhận liên kết từ hệ điều hành của thiết bị người dùng, trong đó liên kết này nhận dạng hoạt động thứ nhất;

tạo yêu cầu khôi phục thứ nhất dựa trên các tham số tương ứng với hoạt động thứ nhất; và

truyền yêu cầu khôi phục thứ nhất đến bộ xử lý của ứng dụng bổ sung.

2. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó các tham số tương ứng với hoạt động thứ nhất bao gồm kiểu hoạt động và tên hoạt động.

3. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó yêu cầu khôi phục thứ nhất được truyền với cuộc gọi `continueuserActivity` đến ủy nhiệm app của ứng dụng bổ sung.

4. Hệ thống phát triển ứng dụng di động theo điểm 3, trong đó hoạt động thứ nhất bao gồm đối tượng `NSNuserActivity`.

5. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó:

hệ thống phân tích ngoại tuyến được tạo cấu hình để (i) phân tích ứng dụng thứ nhất để xác định tập hợp của các đối tượng dữ liệu mà bộ xử lý trong ứng dụng thứ nhất được lập trình để truy cập nhằm đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ, (ii) xác định các tham số cho mỗi đối tượng dữ liệu trong số tập hợp của các đối tượng dữ liệu, và (iii) tạo tập hợp liên kết thứ hai, trong đó mỗi liên kết trong số tập hợp liên kết thứ hai tương ứng với đối tượng dữ liệu tương ứng trong số tập hợp các đối tượng dữ liệu; và

thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, theo sau việc cài đặt của ứng dụng bổ sung trong thiết bị người dùng:

nhận liên kết thứ hai từ hệ điều hành của thiết bị người dùng, trong đó liên kết thứ hai nhận dạng đối tượng dữ liệu thứ nhất,

tạo yêu cầu khôi phục thứ hai dựa trên các tham số tương ứng với đối tượng dữ liệu thứ nhất, và

truyền yêu cầu khôi phục thứ hai đến bộ xử lý của ứng dụng bổ sung.

6. Hệ thống phát triển ứng dụng di động theo điểm 5, trong đó đối tượng dữ liệu thứ nhất là đối tượng `CSSearchableItem`.

7. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó:

bằng cách sạt nhập thư viện định tuyến, ứng dụng bổ sung được tạo cấu hình để đăng ký mẫu mã nhận dạng tài nguyên đồng nhất (uniform resource identifier - URI) thứ nhất với hệ điều hành nhờ việc cài đặt; và

liên kết thích hợp cho mẫu URI thứ nhất.

8. Hệ thống phát triển ứng dụng di động theo điểm 7, trong đó:

mẫu URI thứ nhất xác định sơ đồ bao gồm chuỗi văn bản kết thúc với dấu hai chấm và hai dấu gạch lên;

sơ đồ là duy nhất cho ứng dụng bổ sung;

chuỗi văn bản là sự nối mã nhận dạng văn bản được liên kết với cổng nhà phát triển và tên của ứng dụng thứ nhất; và

liên kết bắt đầu với sơ đồ.

9. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó:

hệ thống phân tích ngoại tuyến được tạo cấu hình để:

chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi các sự kiện giao diện người dùng (UI) được khởi động bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất; và

lưu trữ chuỗi các sự kiện UI làm cấu trúc dữ liệu thứ nhất tương ứng với trạng thái thứ nhất;

thư viện định tuyến bao gồm các lệnh được tạo cấu hình để:

đáp lại người dùng của thiết bị người dùng bắt gặp trạng thái thứ nhất trong lúc sử dụng ứng dụng bổ sung, truyền đánh dấu đến hệ điều hành; và

đáp lại việc nhận yêu cầu khôi phục chỉ báo trạng thái thứ nhất từ hệ điều hành, phát lại chuỗi của các sự kiện UI từ cấu trúc dữ liệu thứ nhất.

10. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó:

hệ thống phân tích ngoại tuyến được tạo cấu hình để:

chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi các sự kiện giao diện người dùng (UI) được khởi động bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất tương ứng với đối tượng dữ liệu thứ nhất; và

lưu trữ chuỗi các sự kiện UI làm cấu trúc dữ liệu thứ nhất tương ứng với đối tượng dữ liệu thứ nhất;

thư viện định tuyến bao gồm các lệnh được tạo cấu hình để:

đáp lại việc chạy ứng dụng bổ sung, truyền chọn lọc đối tượng dữ liệu thứ nhất đến hệ điều hành; và

đáp lại việc nhận yêu cầu khôi phục chỉ báo đối tượng dữ liệu thứ nhất từ hệ điều hành, phát lại chuỗi các sự kiện UI từ cấu trúc dữ liệu thứ nhất.

11. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó:

liên kết bao gồm chọn lọc dữ liệu nối tiếp; và

thư viện định tuyến bao gồm các lệnh được tạo cấu hình để giải mã các tham số tương ứng với hoạt động thứ nhất từ dữ liệu nối tiếp trong liên kết.

12. Hệ thống phát triển ứng dụng di động theo điểm 1, trong đó:

liên kết bao gồm chọn lọc mã nhận dạng duy nhất; và

thư viện định tuyến bao gồm các lệnh được tạo cấu hình để truy hồi các tham số tương ứng với hoạt động thứ nhất dựa trên mã nhận dạng duy nhất.

13. Hệ thống phát triển ứng dụng di động theo điểm 12, hệ thống này còn bao gồm:

máy chủ dữ liệu được tạo cấu hình để lưu trữ các tham số tương ứng với các hoạt động được xác định với hệ thống phân tích ngoại tuyến,

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, nhờ lần chạy thứ nhất của ứng dụng bổ sung, tải xuống các tham số tương ứng với ít nhất hoạt động thứ nhất từ máy chủ dữ liệu để lưu trữ trong bộ phận kho dữ liệu đến thư viện định tuyến, và

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để truy hồi các tham số tương ứng với hoạt động thứ nhất từ kho dữ liệu đáp lại mã nhận dạng duy nhất.

14. Hệ thống phát triển ứng dụng di động bao gồm:

hệ thống phát triển ứng dụng di động theo điểm 1; và

hệ thống tìm kiếm được tạo cấu hình để trả lại các kết quả cho thiết bị người dùng nhằm đáp lại tìm kiếm được ủy quyền bởi người dùng của thiết bị người dùng,

trong đó kết quả thứ nhất trong số các kết quả được trả lại bao gồm liên kết, và

trong đó, đáp lại việc chọn kết quả thứ nhất bởi người dùng, liên kết này được truyền đến thư viện định tuyến bởi hệ điều hành.

15. Phương pháp điều hành hệ thống phát triển ứng dụng di động, phương pháp này bao gồm các bước:

nhận bản sao của ứng dụng thứ nhất từ nhà phát triển thứ nhất;

cung cấp thư viện định tuyến cho nhà phát triển thứ nhất để sát nhập vào ứng dụng thứ nhất trước khi nhà phát triển thứ nhất phân phối ứng dụng bổ sung thông qua nền tảng phân phối số, trong đó thư viện định tuyến được sát nhập vào ứng dụng thứ nhất để tạo ứng dụng bổ sung;

phân tích ứng dụng thứ nhất để (i) xác định tập hợp các hoạt động mà bộ xử lý trong ứng dụng thứ nhất được lập trình để khôi phục nhằm đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ và (ii) xác định các tham số cho mỗi hoạt động trong số tập hợp của các hoạt động; và

tạo ra tập hợp các liên kết, trong đó mỗi liên kết trong số tập hợp các liên kết tương ứng với hoạt động tương ứng trong số tập hợp của các hoạt động,

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, theo sau việc cài đặt ứng dụng bổ sung từ nền tảng phân phối số trên thiết bị người dùng:

nhận liên kết từ hệ điều hành của thiết bị người dùng, trong đó liên kết nhận dạng hoạt động thứ nhất;

tạo yêu cầu khôi phục thứ nhất dựa trên các tham số tương ứng với hoạt động thứ nhất; và

truyền yêu cầu khôi phục thứ nhất đến bộ xử lý của ứng dụng bổ sung.

16. Phương pháp theo điểm 15, trong đó các tham số tương ứng với hoạt động thứ nhất bao gồm kiểu hoạt động và tên hoạt động.

17. Phương pháp theo điểm 15, trong đó yêu cầu khôi phục thứ nhất được truyền với cuộc gọi `continueuserActivity` đến ủy nhiệm app của ứng dụng bổ sung.

18. Phương pháp theo điểm 15, phương pháp này còn bao gồm các bước:

phân tích ứng dụng thứ nhất để xác định tập hợp của các đối tượng dữ liệu mà bộ xử lý trong ứng dụng thứ nhất được lập trình để truy cập nhằm đáp lại các yêu cầu khôi phục tương ứng từ hệ điều hành chủ;

xác định các tham số cho mỗi đối tượng dữ liệu trong số tập hợp các đối tượng dữ liệu; và

tạo ra tập hợp liên kết thứ hai, trong đó mỗi liên kết trong số tập hợp liên kết thứ hai tương ứng với đối tượng dữ liệu tương ứng của tập hợp các đối tượng dữ liệu,

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, theo sau việc cài đặt ứng dụng bổ sung trong thiết bị người dùng:

nhận liên kết thứ hai từ hệ điều hành của thiết bị người dùng, trong đó liên kết thứ hai nhận dạng đối tượng dữ liệu thứ nhất,

tạo yêu cầu khôi phục thứ hai dựa trên các tham số tương ứng với đối tượng dữ liệu thứ nhất, và

truyền yêu cầu khôi phục thứ hai đến bộ xử lý của ứng dụng bổ sung.

19. Phương pháp theo điểm 18, trong đó đối tượng dữ liệu thứ nhất là đối tượng CSSearchableItem.

20. Phương pháp theo điểm 15, trong đó:

bằng cách sát nhập thư viện định tuyến, ứng dụng bổ sung được tạo cấu hình để đăng ký mẫu mã nhận dạng tài nguyên đồng nhất (URI) thứ nhất với hệ điều hành nhờ việc cài đặt; và

liên kết thích hợp với mẫu URI thứ nhất.

21. Phương pháp theo điểm 15, phương pháp này còn bao gồm các bước:

chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi các sự kiện giao diện người dùng (UI) được khởi động bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất; và

lưu trữ chuỗi các sự kiện UI làm cấu trúc dữ liệu thứ nhất tương ứng với trạng thái thứ nhất,

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để:

đáp lại người dùng của thiết bị người dùng bắt gặp trạng thái thứ nhất trong lúc sử dụng ứng dụng bổ sung, truyền đánh dấu đến hệ điều hành; và

đáp lại việc nhận yêu cầu khôi phục chỉ báo trạng thái thứ nhất từ hệ điều hành, phát lại chuỗi các sự kiện UI từ cấu trúc dữ liệu thứ nhất.

22. Phương pháp theo điểm 15, phương pháp này còn bao gồm các bước:

chạy ứng dụng thứ nhất và, trong lúc chạy, giám sát chuỗi các sự kiện giao diện người dùng (UI) được khởi động bởi người vận hành bắt đầu từ trạng thái ban đầu và tiếp tục đến trạng thái thứ nhất tương ứng với đối tượng dữ liệu thứ nhất; và

lưu trữ chuỗi các sự kiện UI làm cấu trúc dữ liệu thứ nhất tương ứng với đối tượng dữ liệu thứ nhất;

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để:

đáp lại việc chạy ứng dụng bổ sung, truyền chọn lọc đối tượng dữ liệu thứ nhất đến hệ điều hành; và

đáp lại việc nhận yêu cầu khôi phục chỉ báo đối tượng dữ liệu thứ nhất từ hệ điều hành, phát lại chuỗi các sự kiện UI từ cấu trúc dữ liệu thứ nhất.

23. Phương pháp theo điểm 15, trong đó:

liên kết bao gồm chọn lọc dữ liệu nối tiếp; và

thư viện định tuyến bao gồm các lệnh được tạo cấu hình để giải mã các tham số tương ứng với hoạt động thứ nhất từ dữ liệu nối tiếp trong liên kết.

24. Phương pháp theo điểm 15, trong đó:

liên kết bao gồm chọn lọc mã nhận dạng duy nhất; và

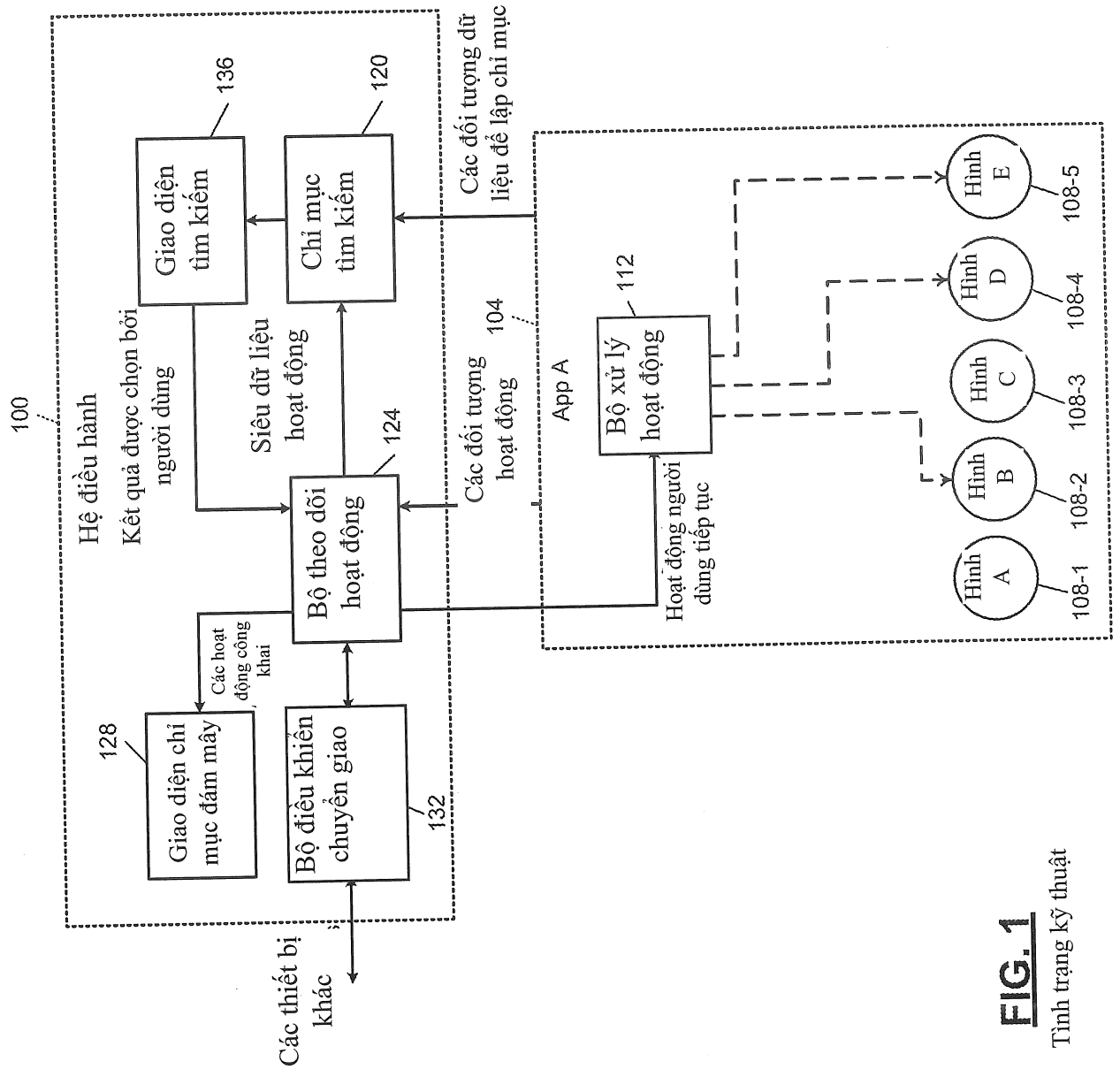
thư viện định tuyến bao gồm các lệnh được tạo cấu hình để truy hồi các tham số tương ứng với hoạt động thứ nhất dựa trên mã nhận dạng duy nhất.

25. Phương pháp theo điểm 24, phương pháp này còn bao gồm các bước:

lưu trữ, trong kho dữ liệu, các tham số tương ứng với tập hợp các hoạt động được xác định,

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để, nhờ lần chạy thứ nhất của ứng dụng bổ sung, tải xuống các tham số tương ứng với ít nhất hoạt động thứ nhất từ kho dữ liệu để lưu trữ trong bộ phận kho dữ liệu đến thư viện định tuyến, và

trong đó thư viện định tuyến bao gồm các lệnh được tạo cấu hình để truy hỏi các tham số tương ứng với hoạt động thứ nhất từ kho dữ liệu đáp lại mã nhận dạng duy nhất.

**FIG. 1**

Tình trạng kỹ thuật

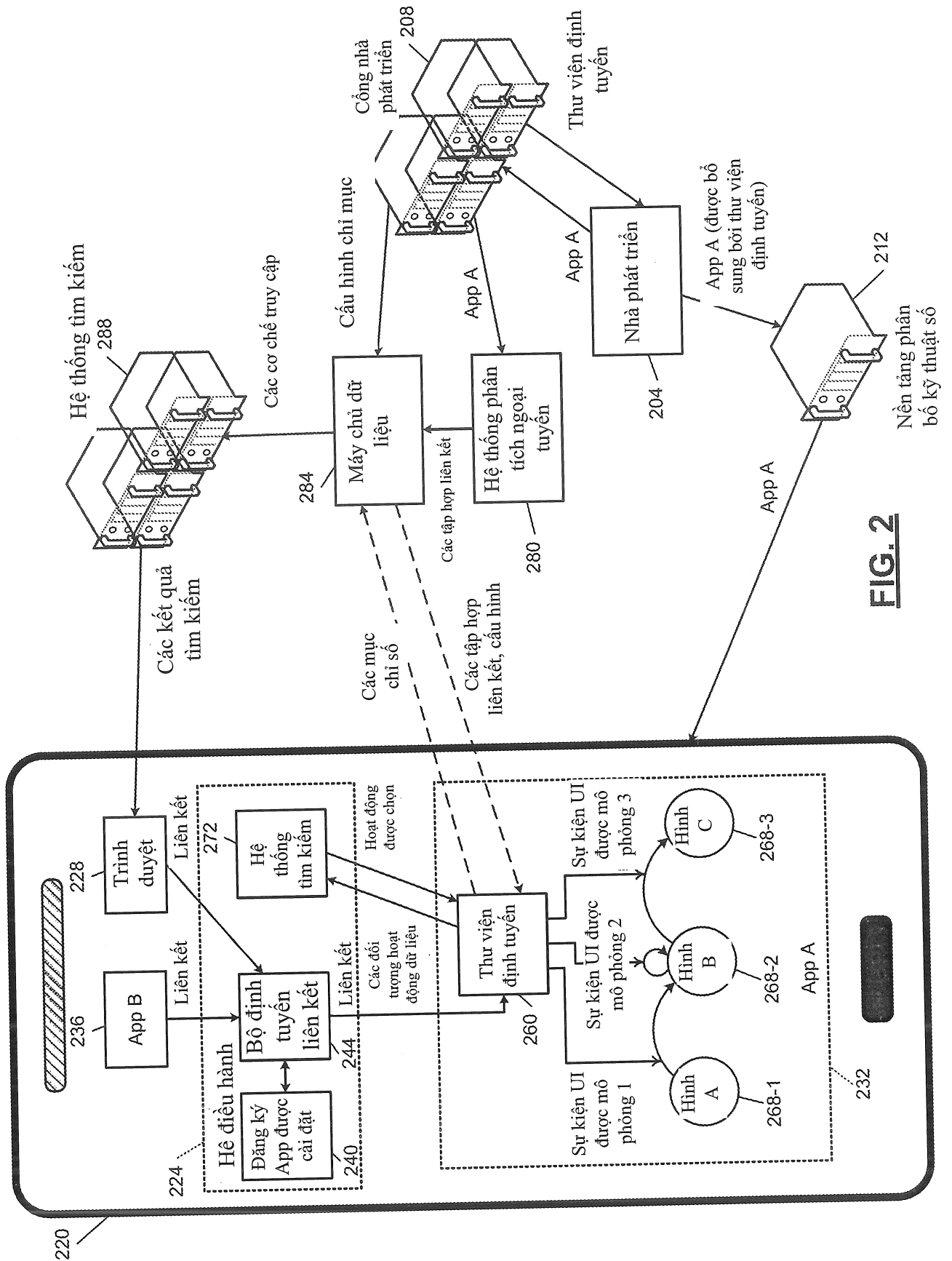


FIG. 2

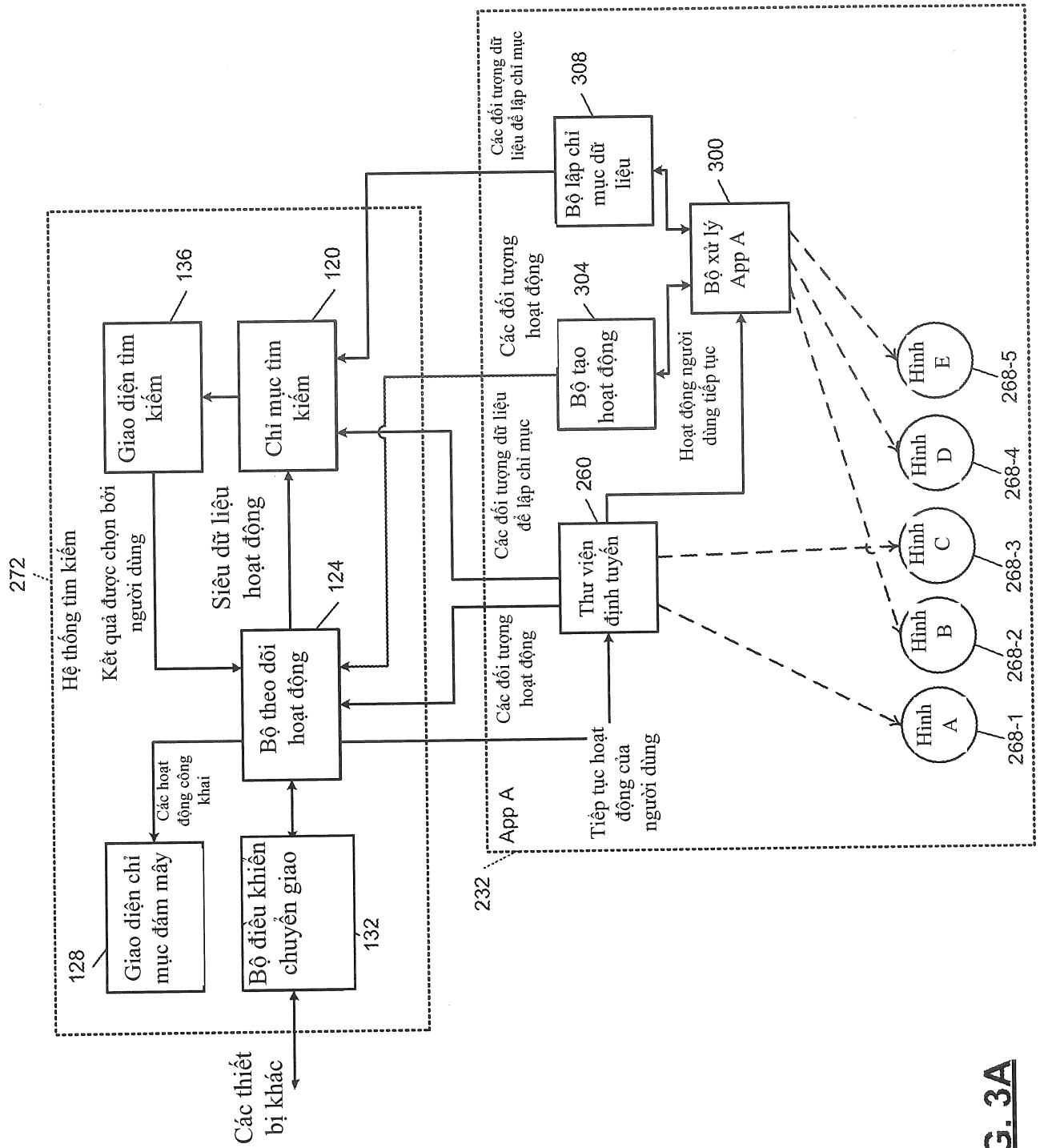


FIG. 3A

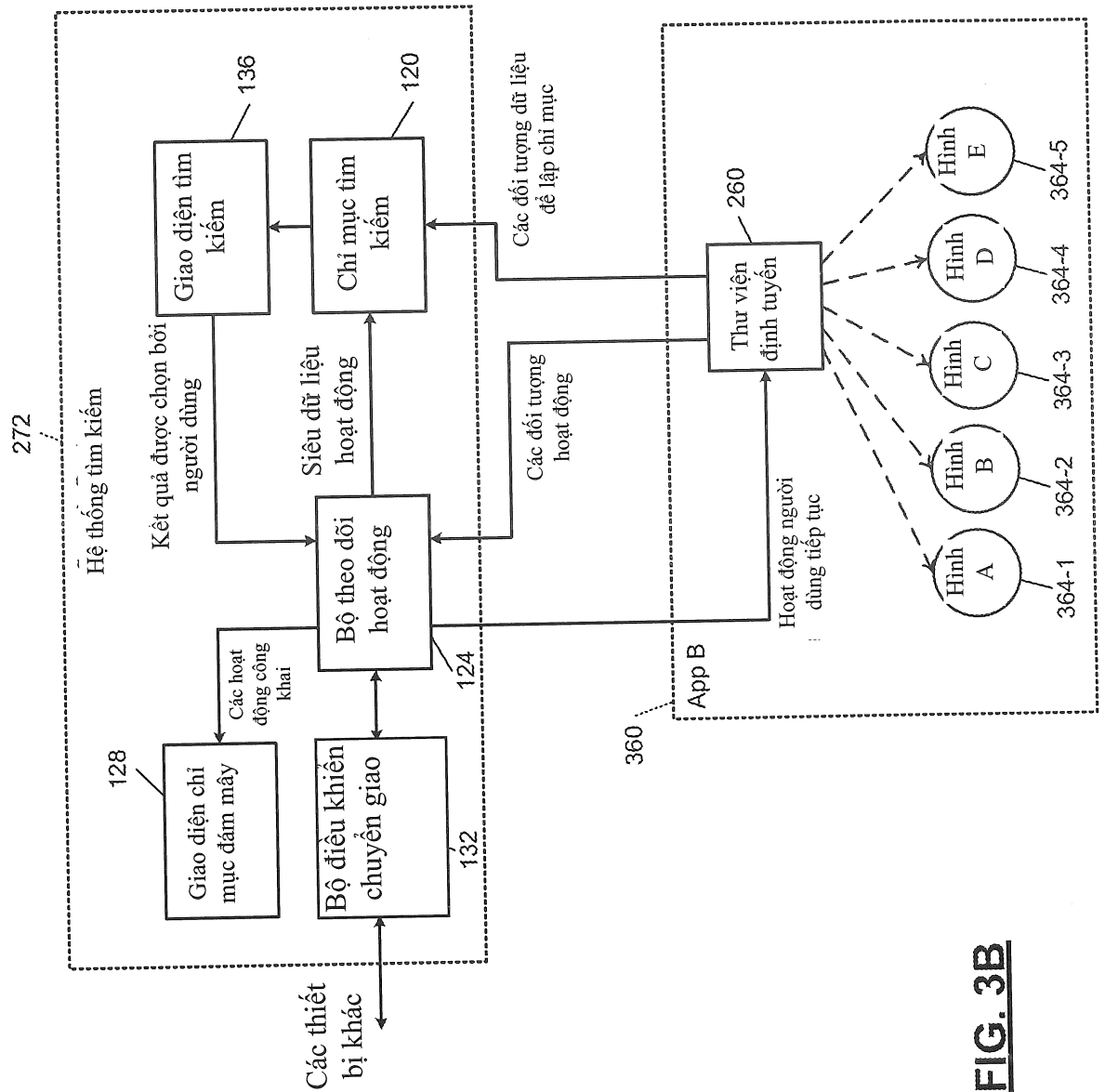


FIG. 3B

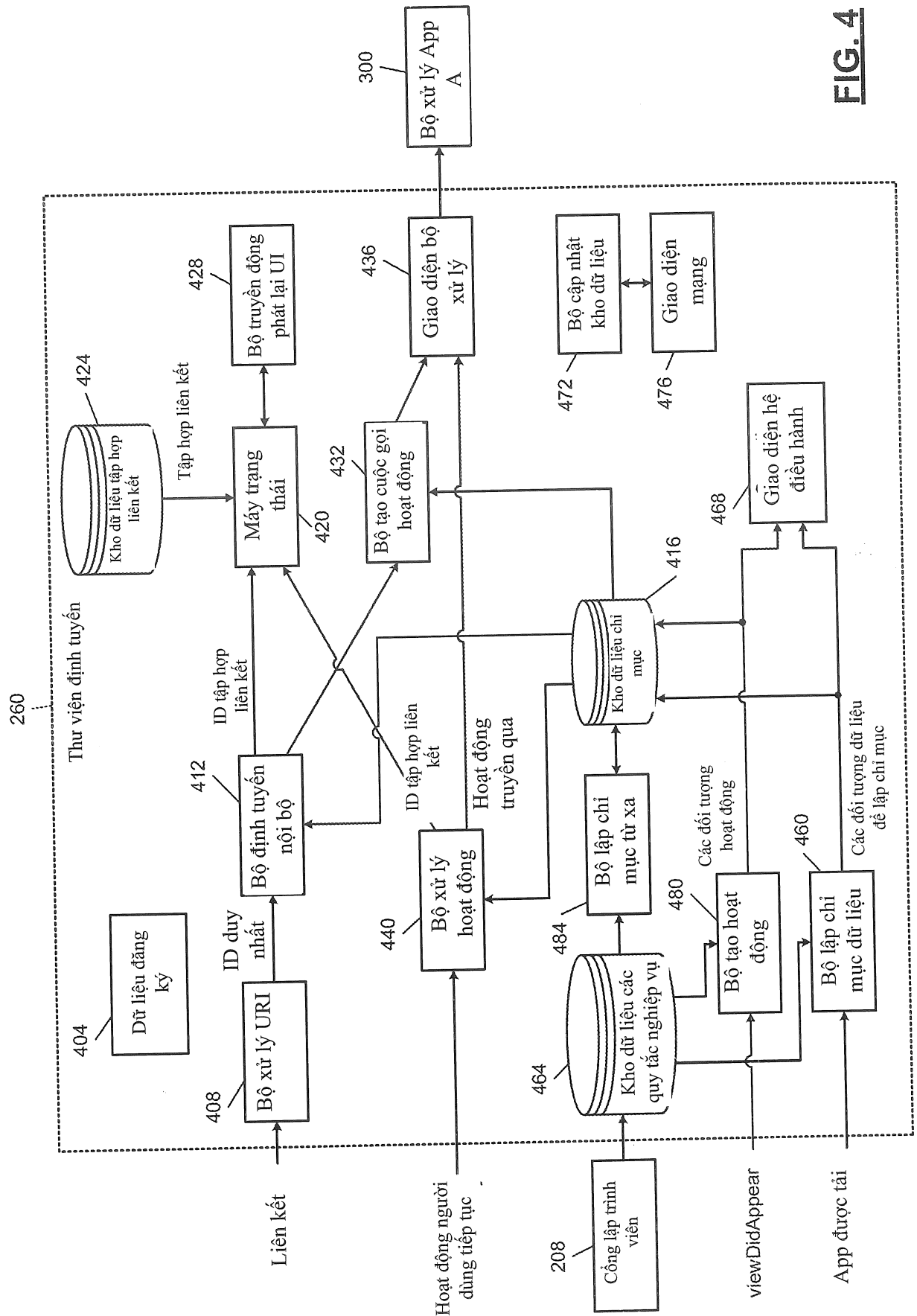
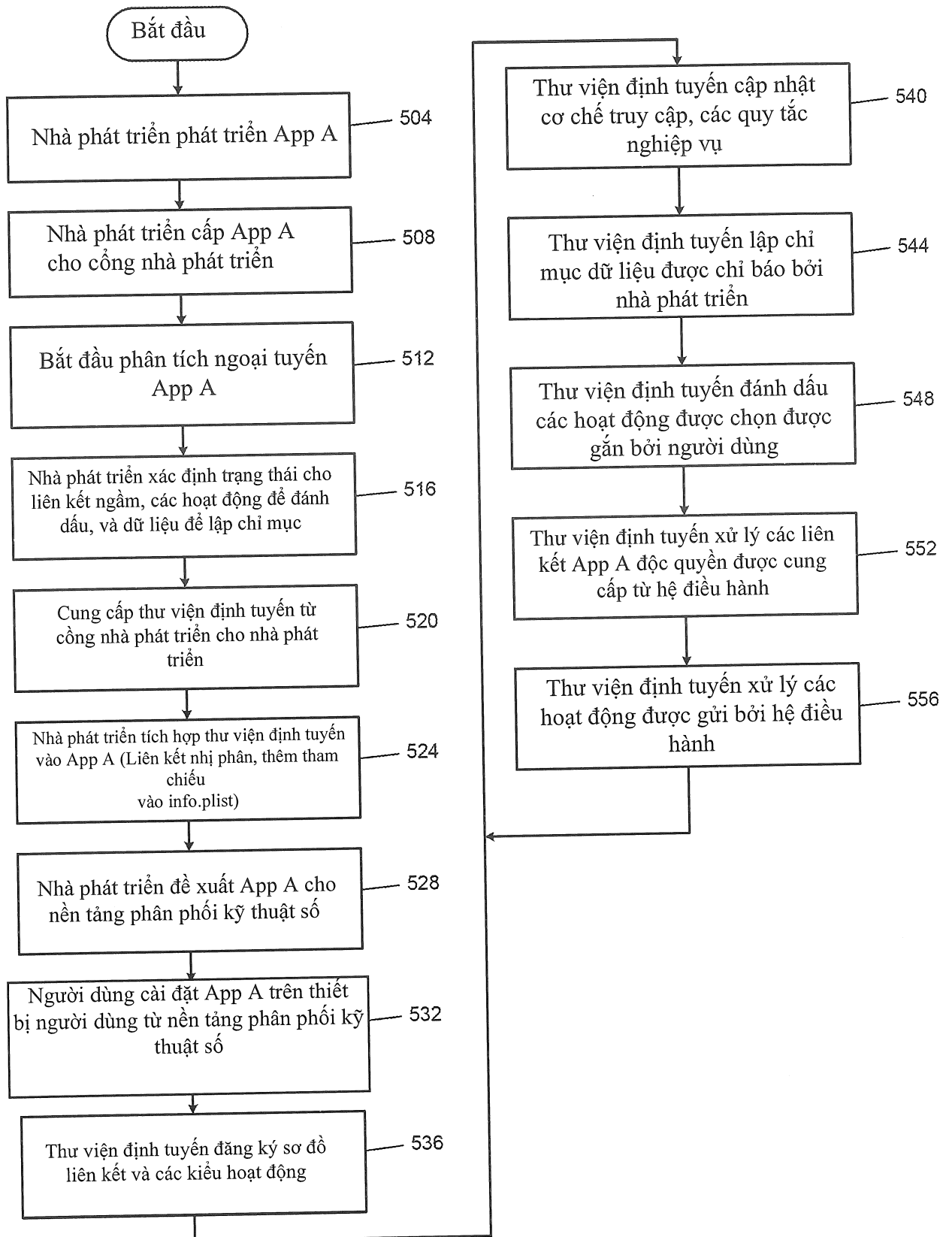
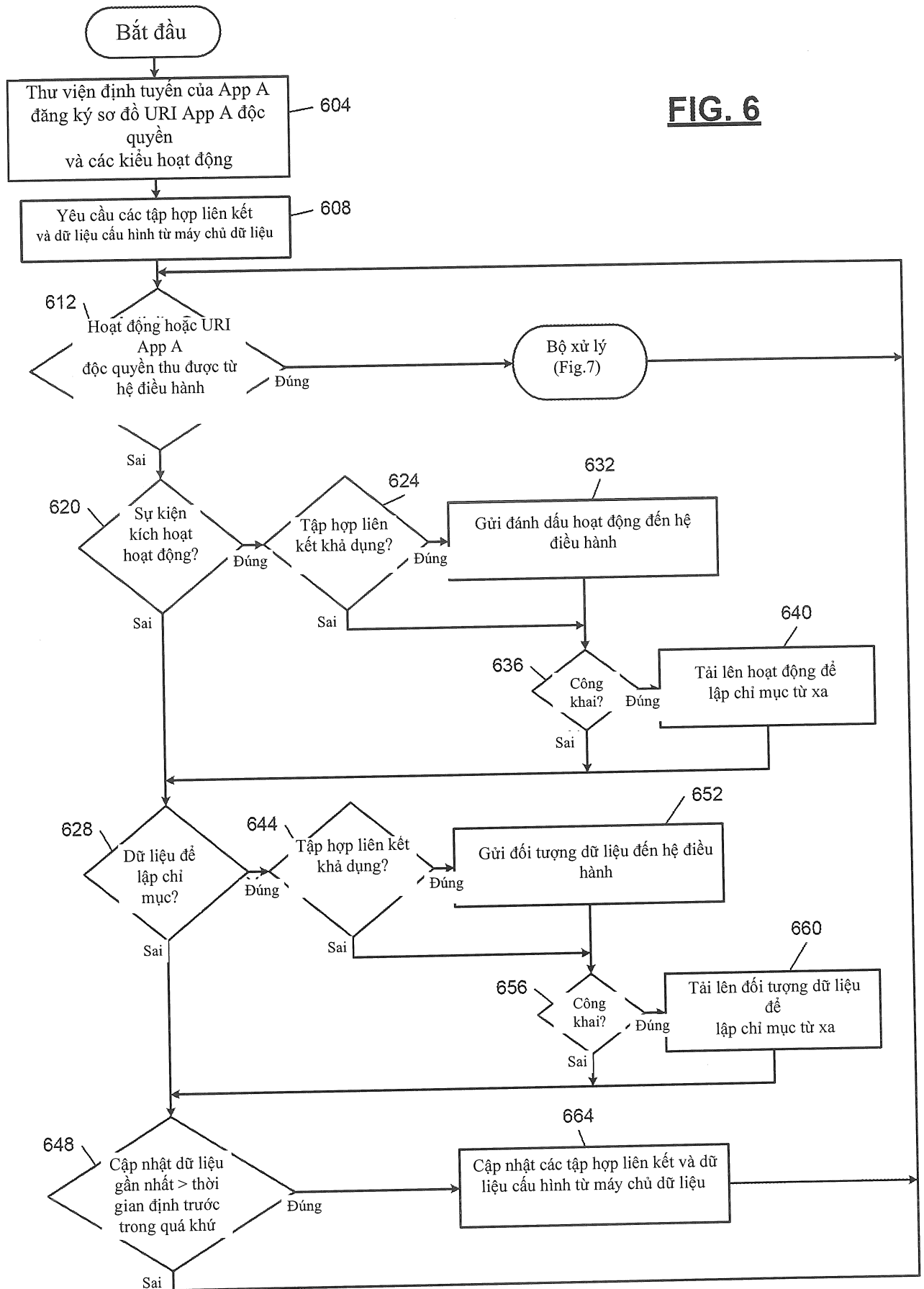


FIG. 4

**FIG. 5**



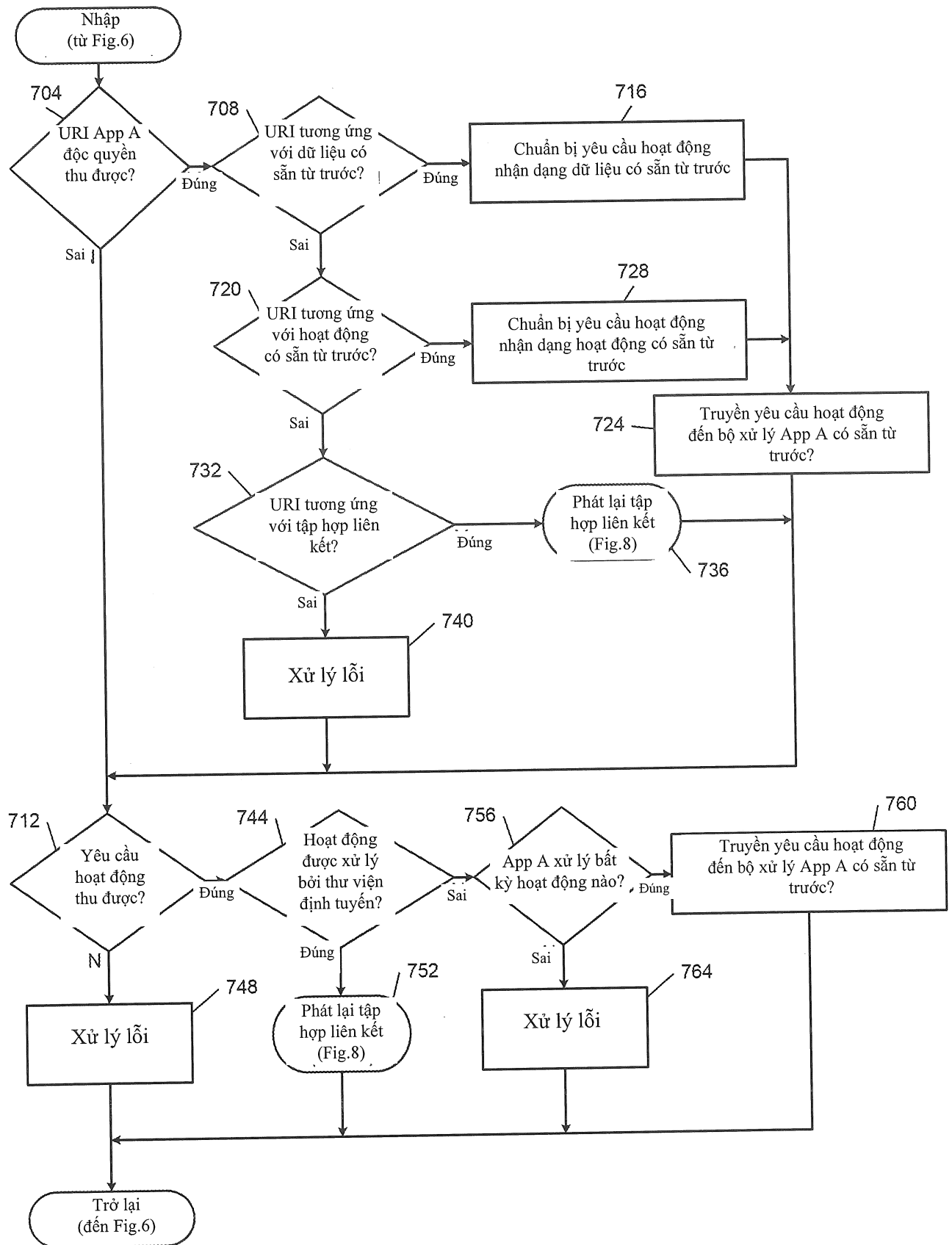
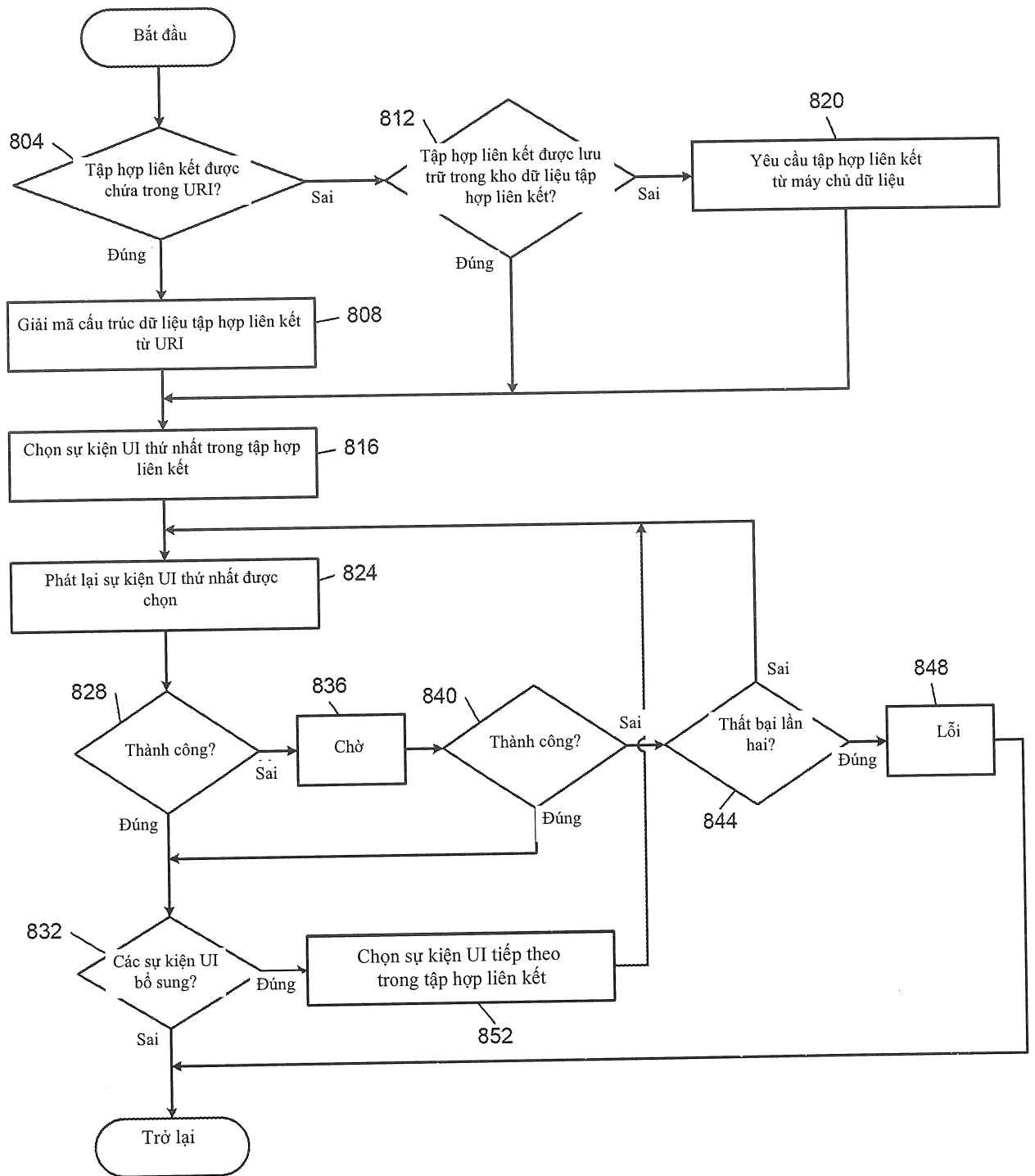
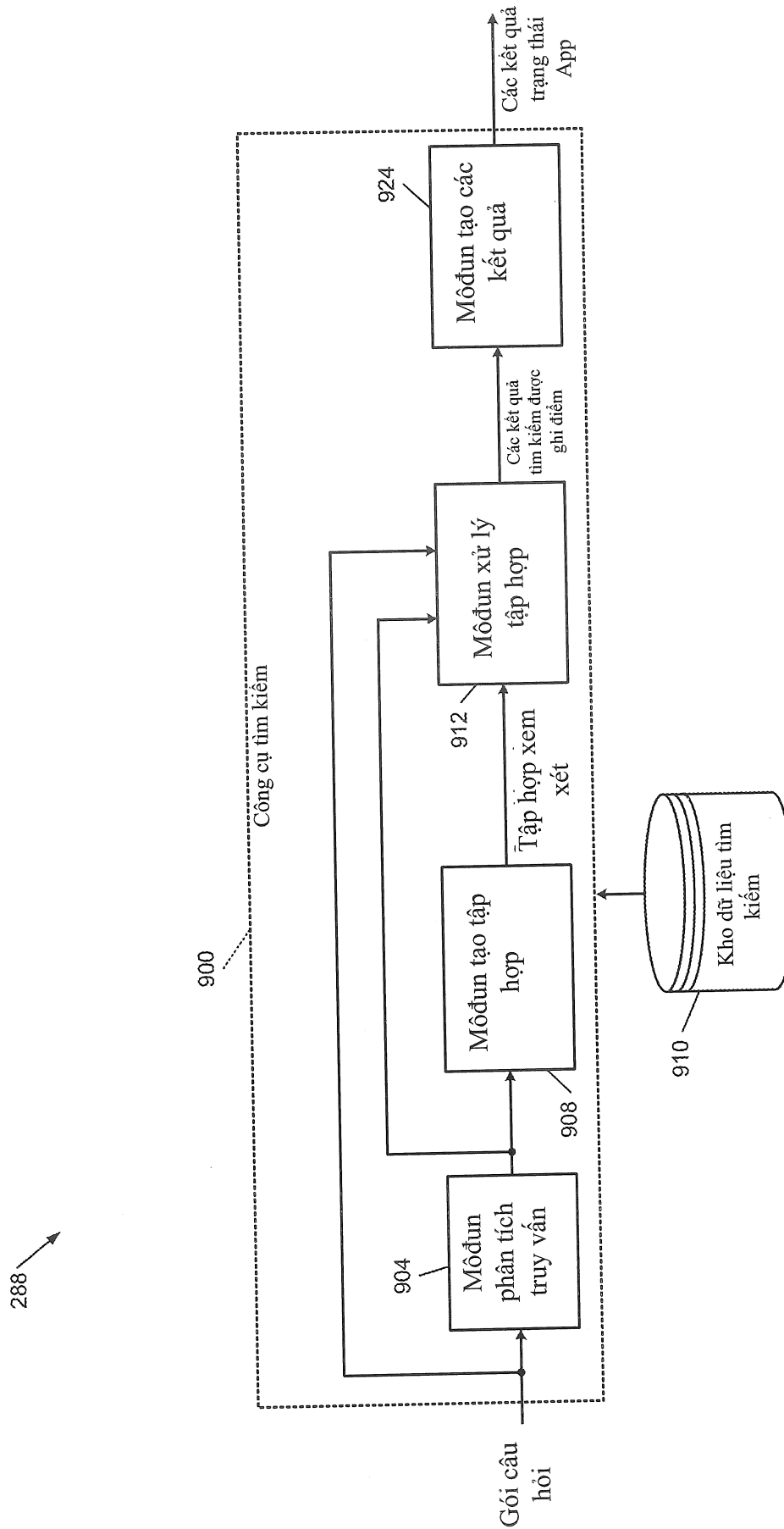
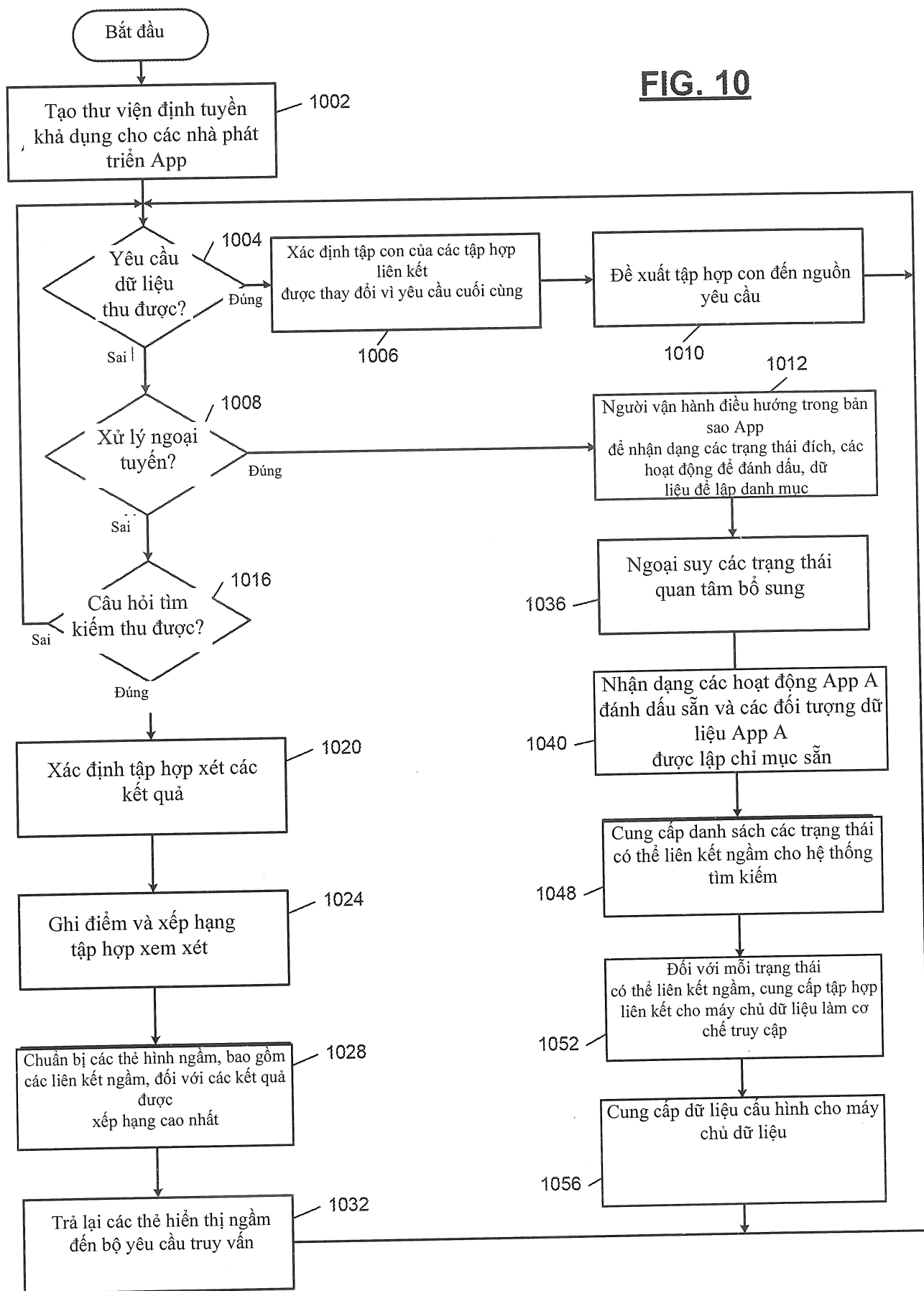
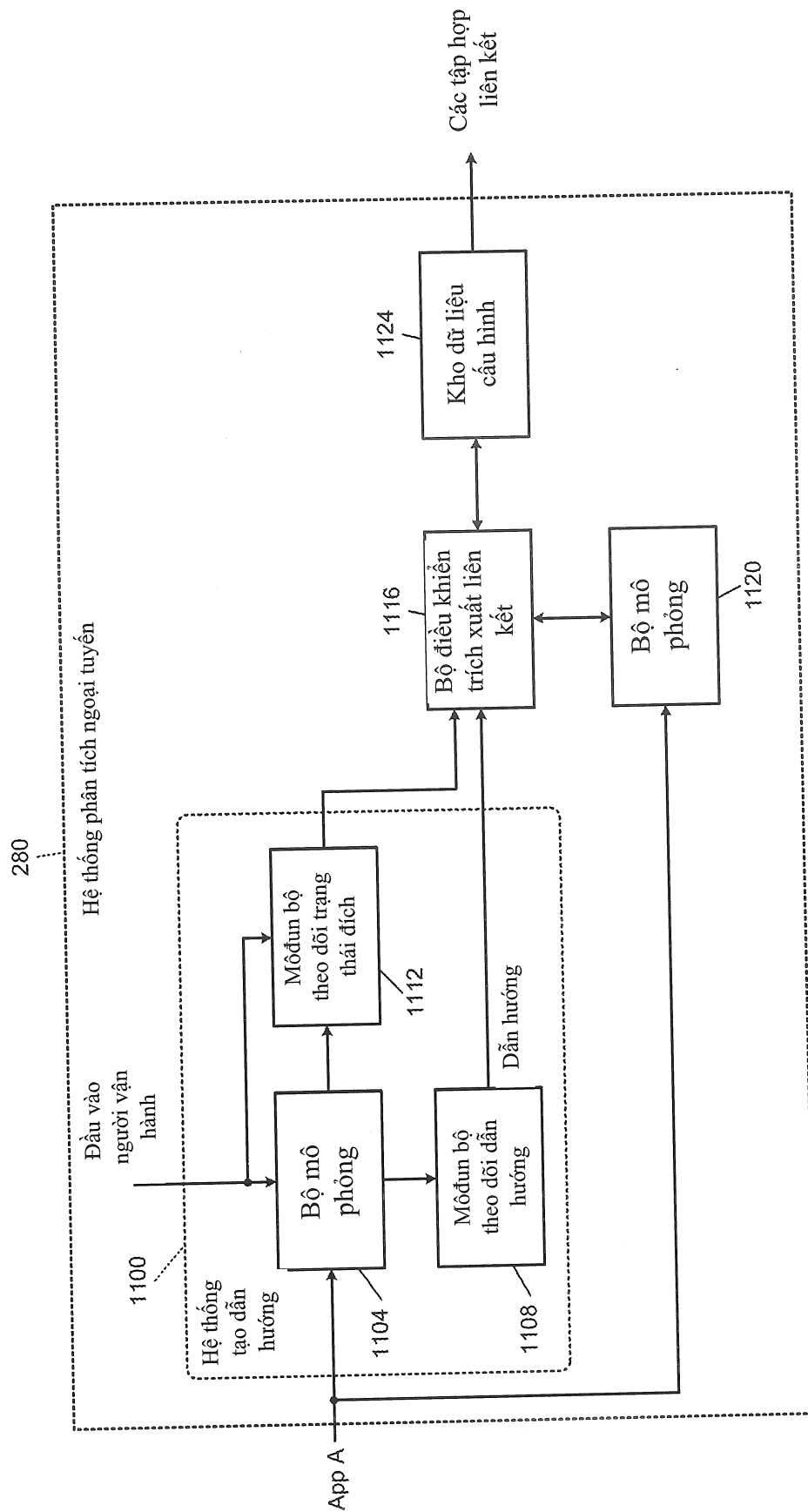


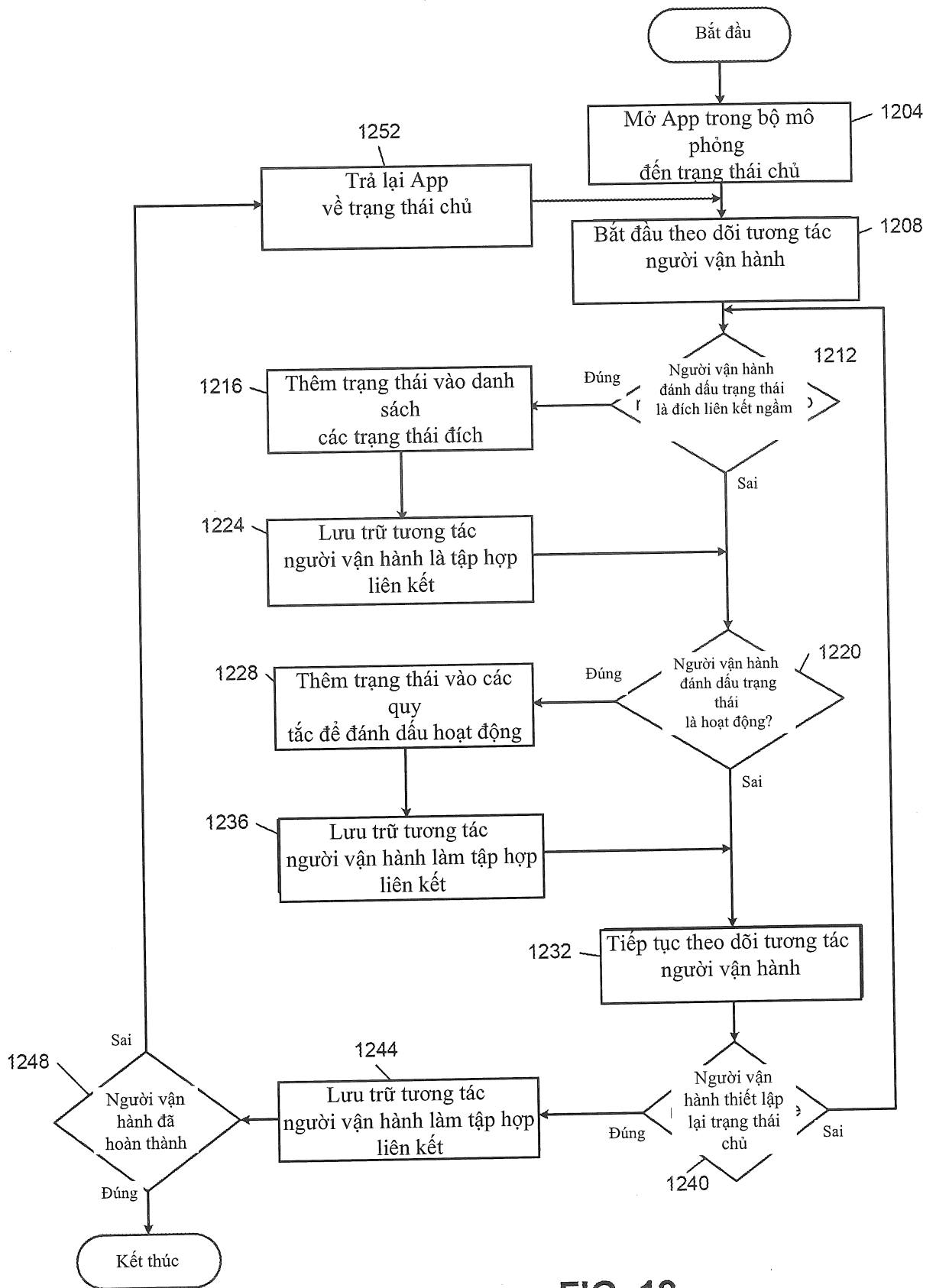
FIG. 7

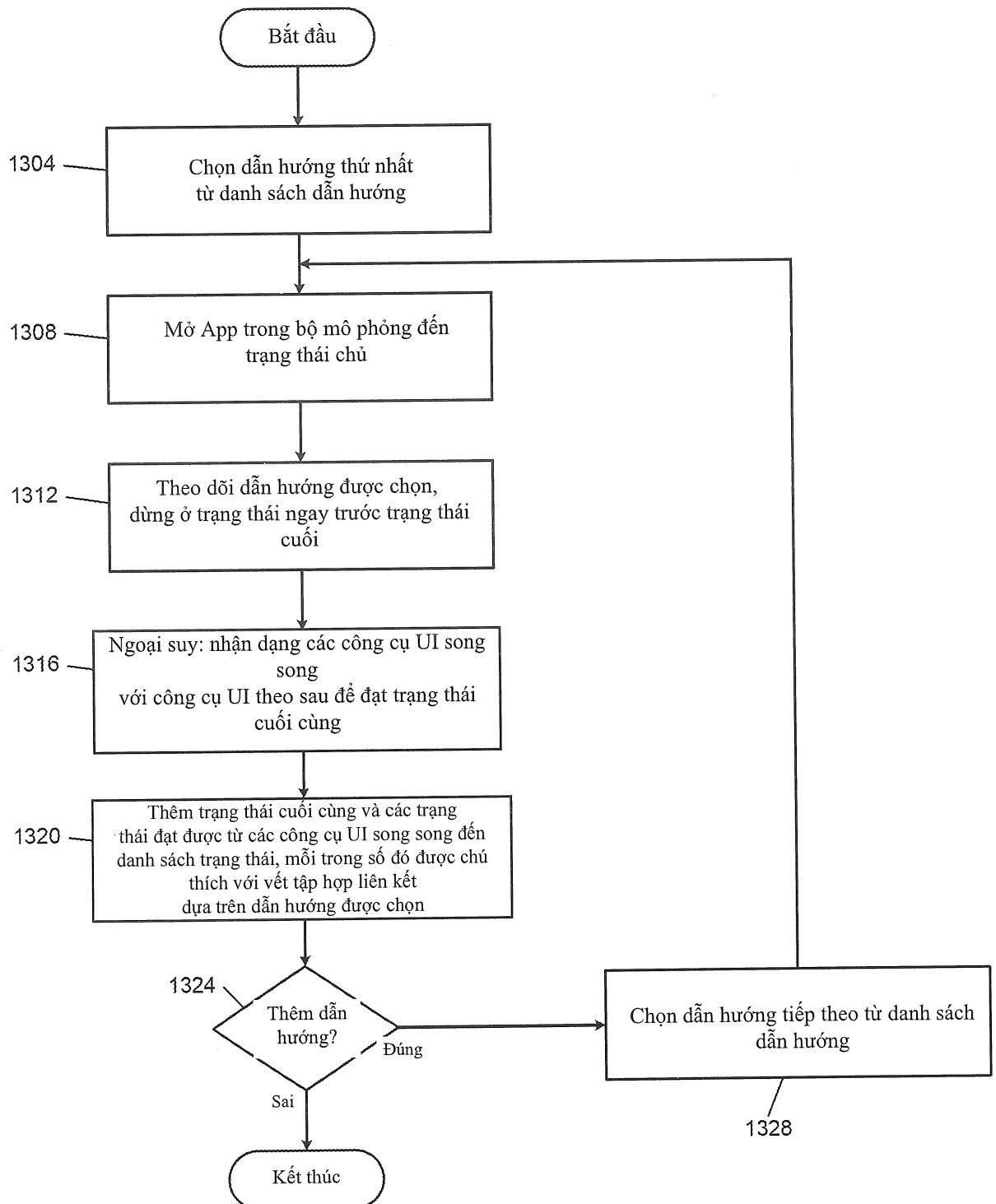
**FIG. 8**

**FIG. 9**



**FIG. 11**

**FIG. 12**

**FIG. 13**