



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2020년12월23일

(11) 등록번호 10-2194055

(24) 등록일자 2020년12월16일

(51) 국제특허분류(Int. Cl.)

G06F 9/44 (2018.01)

(52) CPC특허분류

G06F 9/44 (2013.01)

(21) 출원번호 10-2018-7028698

(22) 출원일자(국제) 2017년03월03일

심사청구일자 2020년01월10일

(85) 번역문제출일자 2018년10월04일

(65) 공개번호 10-2018-0125498

(43) 공개일자 2018년11월23일

(86) 국제출원번호 PCT/CN2017/075520

(87) 국제공개번호 WO 2017/157186

국제공개일자 2017년09월21일

(30) 우선권주장

201610150785.X 2016년03월16일 중국(CN)

(56) 선행기술조사문헌

KR101543236 B1*

Kapoor Sangel. Android Activity Lifecycle.
2016.03.10.

US20150248723 A1

US20150371363 A1

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

어드밴스드 뉴 테크놀로지스 씨오., 엘티디.

케이만 군도, 그랜드 케이만 케이와이1-9008, 조지 타운, 27 하스피탈 로드, 케이만 코포레이트 센터

(72) 발명자

자오 시양유

중국 저지앙 311121 유 항 디스트릭트 항저우 넘버 969 웨스트 웬 이 로드 빌딩 3 알리바바 그룹 리갈 디파트먼트 5층

딩 리양지

중국 저지앙 311121 유 항 디스트릭트 항저우 넘버 969 웨스트 웬 이 로드 빌딩 3 알리바바 그룹 리갈 디파트먼트 5층

(74) 대리인

김태홍, 김진희

전체 청구항 수 : 총 8 항

심사관 : 최정권

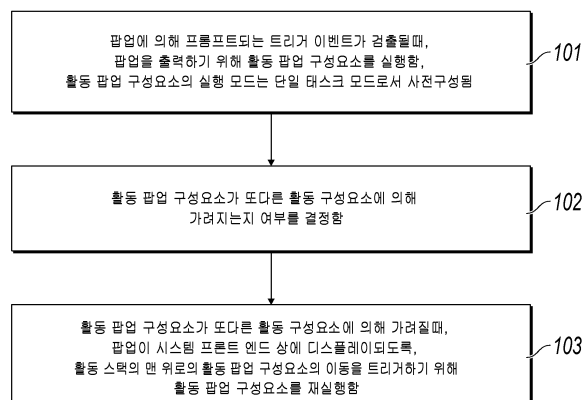
(54) 발명의 명칭 안드로이드 기반의 팝업 프롬프트 방법 및 디바이스

(57) 요약

본 출원은 안드로이드 기반의 팝업 프롬프트 방법 및 디바이스를 제공한다. 방법은, 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 팝업을 출력하기 위해 활동 팝업 구성요소를 실행함, 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성됨

(뒷면에 계속)

대표도 - 도1



구성요소에 의해 가려지는지 여부를 결정하는 단계; 및 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 팝업이 시스템 프론트 엔드 상에 디스플레이되도록, 활동 스택의 맨 위로의 활동 팝업 구성요소의 이동을 트리거하기 위해 활동 팝업 구성요소를 재실행하는 단계를 포함한다. 본 출원에 따르면, 사용자에게 팝업을 출력하기 위해 활동 팝업 구성요소가 실행된 후에, 팝업이 가려지지 않으며 항상 시스템 프론트 엔드 상에 디스플레이되는 것이 보장될 수 있다.

명세서

청구범위

청구항 1

팝업 구성요소(pop-up components)를 실행하는(launch) 컴퓨터 구현(computer-implemented) 방법에 있어서,

팝업 윈도우의 프리젠테이션에 대한 요청의 검출에 의해 생성된 트리거 이벤트를 수신하는 단계;

상기 수신된 트리거 이벤트에 기초하여, 상기 팝업 윈도우를 출력하도록 구성된 활동 팝업 구성요소(activity pop-up component)를 실행하는 단계로서, 상기 활동 팝업 구성요소의 실행 모드는 사용자 입력에 의해 제공된 팝업 윈도우 제한에 의해 팝업 윈도우 출력을 금지하는 단일 태스크 모드로서 사전구성되어 있는 것인, 상기 실행하는 단계;

상기 활동 팝업 구성요소의 onStop 메소드(method) - 상기 onStop 메소드는 상기 활동 팝업 구성요소의 팝업 윈도우 출력이 기존의 팝업 윈도우에 의해 완전히 가려지도록(obscured) 함 - 가 상기 단일 태스크 모드 동안 호출되는지 여부를 결정하는 단계;

상기 활동 팝업 구성요소의 팝업 윈도우 출력이 기존의 팝업 윈도우에 의해 완전히 가려진다고 결정하는 것에 응답하여, onPause 메소드가 호출되었다고 결정하는 단계; 및

상기 onPause 메소드가 호출되었다고 결정하는 것에 응답하여, 활동 스택(activity stack)과 적어도 부분적으로 중첩하는 상기 팝업 윈도우의 가려지지 않은(non-obscured) 디스플레이를 강행하도록 상기 활동 스택의 맨 위(top)로의 상기 팝업 윈도우의 이동을 트리거하기 위해 상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소가 재실행되었다고 결정하는 단계; 및

상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소가 재실행되었다고 결정하는 것에 응답하여, 상기 onStop 메소드에 로드된 재실행 로직의 실행(execution)을 사용중지시키는(disabling) 단계를 포함하는 컴퓨터 구현 방법.

청구항 2

청구항 1에 있어서, 상기 트리거 이벤트는 안드로이드 디바이스 상에서 수신되는 것인 컴퓨터 구현 방법.

청구항 3

청구항 1에 있어서, 상기 트리거 이벤트는 단일 디바이스 로그인 이벤트인 것인 컴퓨터 구현 방법.

청구항 4

팝업 구성요소를 실행하는 동작들을 수행하도록 컴퓨터 시스템에 의해 실행가능한 하나 이상의 명령어를 저장한 비일시적 컴퓨터 판독가능 매체에 있어서, 상기 동작들은,

팝업 윈도우의 프리젠테이션에 대한 요청의 검출에 의해 생성된 트리거 이벤트를 수신하는 동작;

상기 수신된 트리거 이벤트에 기초하여, 상기 팝업 윈도우를 출력하도록 구성된 활동 팝업 구성요소를 실행하는 동작으로서, 상기 활동 팝업 구성요소의 실행 모드는 사용자 입력에 의해 제공된 팝업 윈도우 제한에 의해 팝업 윈도우 출력을 금지하는 단일 태스크 모드로서 사전구성되어 있는 것인, 상기 실행하는 동작;

상기 활동 팝업 구성요소의 onStop 메소드 - 상기 onStop 메소드는 상기 활동 팝업 구성요소의 팝업 윈도우 출력이 기존의 팝업 윈도우에 의해 완전히 가려지도록 함 - 가 상기 단일 태스크 모드 동안 호출되는지 여부를 결정하는 동작;

상기 활동 팝업 구성요소의 팝업 윈도우 출력이 기존의 팝업 윈도우에 의해 완전히 가려진다고 결정하는 것에 응답하여, onPause 메소드가 호출되었다고 결정하는 동작;

상기 onPause 메소드가 호출되었다고 결정하는 것에 응답하여, 활동 스택과 적어도 부분적으로 중첩하는 상기 팝업 윈도우의 가려지지 않은 디스플레이를 강행하도록 상기 활동 스택의 맨 위로의 상기 팝업 윈도우의 이동을

트리거하기 위해 상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소가 재실행되었다고 결정하는 동작; 및

상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소가 재실행되었다고 결정하는 것에 응답하여, 상기 onStop 메소드에 로드된 재실행 로직의 실행을 사용중지시키는 동작을 포함하는 것인 비일시적 컴퓨터 판독가능 매체.

청구항 5

청구항 4에 있어서, 상기 트리거 이벤트는 안드로이드 디바이스 상에서 수신되는 것인 비일시적 컴퓨터 판독가능 매체.

청구항 6

청구항 4에 있어서, 상기 트리거 이벤트는 단일 디바이스 로그인 이벤트인 것인 비일시적 컴퓨터 판독가능 매체.

청구항 7

팝업 구성요소를 실행하는 컴퓨터 구현 시스템에 있어서,

하나 이상의 컴퓨터; 및

상기 하나 이상의 컴퓨터와 상호동작가능하게 연결되며, 상기 하나 이상의 컴퓨터에 의해 실행될 때 하나 이상의 동작을 수행하는 하나 이상의 명령어를 저장한 유형의(tangible) 비일시적 기계 판독가능 매체를 갖는 하나 이상의 컴퓨터 메모리 디바이스를 포함하고, 상기 하나 이상의 동작은,

팝업 윈도우의 프리젠테이션에 대한 요청의 검출에 의해 생성된 트리거 이벤트를 수신하는 동작;

상기 수신된 트리거 이벤트에 기초하여, 상기 팝업 윈도우를 출력하도록 구성된 활동 팝업 구성요소를 실행하는 동작으로서, 상기 활동 팝업 구성요소의 실행 모드는 사용자 입력에 의해 제공된 팝업 윈도우 제한에 의해 팝업 윈도우 출력을 금지하는 단일 태스크 모드로서 사전구성되어 있는 것인, 상기 실행하는 동작;

상기 활동 팝업 구성요소의 onStop 메소드 - 상기 onStop 메소드는 상기 활동 팝업 구성요소의 팝업 윈도우 출력이 기존의 팝업 윈도우에 의해 완전히 가려지도록 함 - 가 상기 단일 태스크 모드 동안 호출되는지 여부를 결정하는 동작;

상기 활동 팝업 구성요소의 팝업 윈도우 출력이 기존의 팝업 윈도우에 의해 완전히 가려진다고 결정하는 것에 응답하여, onPause 메소드가 호출되었다고 결정하는 동작; 및

상기 onPause 메소드가 호출되었다고 결정하는 것에 응답하여, 활동 스택과 적어도 부분적으로 중첩하는 상기 팝업 윈도우의 가려지지 않은 디스플레이를 강행하도록 상기 활동 스택의 맨 위로의 상기 팝업 윈도우의 이동을 트리거하기 위해 상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소가 재실행되었다고 결정하는 동작; 및

상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소가 재실행되었다고 결정하는 것에 응답하여, 상기 onStop 메소드에 로드된 재실행 로직의 실행을 사용중지시키는 동작을 포함하고는 것인 컴퓨터 구현 시스템.

청구항 8

청구항 7에 있어서, 상기 트리거 이벤트는 안드로이드 디바이스 상에서 수신되는 것인 컴퓨터 구현 시스템.

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

발명의 설명

기술 분야

- [0001] 본 출원은, 2016년 3월 16일 출원되어 발명의 명칭이 “ANDROID-BASED POP-UP PROMPT METHOD AND DEVICE” 인 중국 특허 출원 번호 제201610150785.X호의 우선권을 주장하며, 이는 그 전체가 참조에 의해 여기에 포함된다.
- [0002] 본 출원은 통신 기술분야에 관한 것으로, 보다 상세하게는 안드로이드 기반의 팝업(pop-up) 프롬프트(prompt) 방법 및 디바이스에 관한 것이다.

배경 기술

- [0003] 안드로이드 시스템에서, 사용자에게 팝업이 출력될 필요가 있는 일부 이벤트에 대하여, 시스템은 사용자에게 프롬프트 텍스트로 채워진 팝업을 출력할 수 있다. 예를 들어, 이벤트는 단일 디바이스 로그인 이벤트일 수 있다. 단일 디바이스 로그인은 보통, 사용자가 계정을 사용함으로써 오로지 하나의 디바이스에만 로그인하는 것이 허용됨을 의미한다. 사용자가 동일 계정을 사용함으로써 또다른 디바이스에 로그인하면, 사용자는 강제로 현재 디바이스로부터 로그인 상태를 종료하여야 한다. 단일 디바이스 로그인 이벤트를 검출할 때, 시스템은 현재 로그인 계정이 또다른 디바이스에 로그인하는 데에 이미 사용됨을 사용자에게 프롬프트하기 위해 사용자에게 팝업을 출력할 수 있다. 이 경우, 팝업에서의 프롬프트 텍스트는 사용자가 강제로 로그인 상태를 종료해야 하는 이유를 포함할 수 있다. 그러나, 관련 기술에서, 안드로이드 시스템이 사용자에게 팝업을 출력할 때, 출력 팝업은 어쩌면 또다른 팝업 또는 또다른 페이지에 의해 가려지며(obscured), 사용자에게 의한 시스템 단말기의 기본 기능 또는 APP의 사용에 영향을 미친다.

발명의 내용

- [0004] 본 출원은 안드로이드 기반의 팝업 프롬프트 방법을 제공하며, 방법은, 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 상기 팝업을 출력하기 위해 활동 팝업 구성요소(activity pop-up component)를 실행하는 (launch) 단계로서, 상기 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성되어 있는 것인, 상기 실행하는 단계; 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부를 결정하는 단계; 및 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 상기 팝업이 시스템 프론트 엔드 상에 디스플레이되도록, 활동 스택의 맨 위(top)로의 상기 활동 팝업 구성요소의 이동을 트리거하기 위해 상기 활동 팝업 구성요소를 재실행하는(relaunch) 단계를 포함한다.
- [0005] 선택적으로, 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부를 결정하는 단계는, 상기 활동 팝업 구성요소의 onPause 또는 onStop 메소드(method)가 호출되는지 여부를 결정하는 단계; 및 상기 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출될 때, 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려진다고 결정하는 단계를 포함한다.
- [0006] 선택적으로, 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 상기 활동 팝업 구성요소를 재실행하는 단계는, 상기 활동 팝업 구성요소의 onPause 메소드가 호출될 때, 상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소를 재실행하는 단계; 및 상기 활동 팝업 구성요소의 onStop 메소드가 호출될 때, 상기 활동 팝업 구성요소의 onPause 메소드가 이미 호출되어 있는지 여부를 결정하고, onPause 메소드가 호출되지 않은 경우 상기 onStop 메소드에 기초하여 상기 활동 팝업 구성요소를 재실행하는 단계를 포함한다.
- [0007] 선택적으로, 상기 onPause 메소드 또는 상기 onStop 메소드에 기초하여 상기 활동 팝업 구성요소를 재실행하는 단계는, 상기 onPause 메소드 또는 상기 onStop 메소드의 호출이 상기 활동 팝업 구성요소에 대한 사용중지(disabling) 이벤트에 의해 트리거되는지 여부를 결정하는 단계; 및 상기 onPause 메소드 또는 상기 onStop 메

소드의 호출이 상기 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거된 것이 아닌 경우, 상기 onPause 메소드 또는 상기 onStop 메소드에 기초하여 상기 활동 팝업 구성요소를 재실행하는 단계를 포함한다.

[0008] 선택적으로, 방법은, 상기 onPause 메소드 또는 상기 onStop 메소드의 호출이 상기 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거된 경우, 상기 활동 팝업 구성요소를 정상적으로 사용중지시키는 단계를 더 포함한다.

[0009] 본 출원은 안드로이드 기반의 팝업 프롬프트 디바이스를 더 제공하며, 디바이스는, 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 상기 팝업을 출력하기 위해 활동 팝업 구성요소를 실행하도록 구성된 실행 모듈로서, 상기 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성되어 있는 것인, 상기 실행 모듈; 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부를 결정하도록 구성된 결정 모듈; 및 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 상기 팝업이 시스템 프론트 엔드 상에 디스플레이되도록, 활동 스택의 맨 위로의 상기 활동 팝업 구성요소의 이동을 트리거하기 위해 상기 활동 팝업 구성요소를 재실행하도록 구성된 재실행 모듈을 포함한다.

[0010] 선택적으로, 상기 결정 모듈은, 상기 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출되는지 여부를 결정하고; 상기 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출될 때, 상기 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려진다고 결정하도록, 구성된다.

[0011] 선택적으로, 상기 재실행 모듈은, 상기 활동 팝업 구성요소의 onPause 메소드가 호출될 때, 상기 onPause 메소드에 기초하여 상기 활동 팝업 구성요소를 재실행하고; 상기 활동 팝업 구성요소의 onStop 메소드가 호출될 때, 상기 활동 팝업 구성요소의 onPause 메소드가 이미 호출되어 있는지 여부를 결정하고, onPause 메소드가 호출되지 않은 경우 상기 onStop 메소드에 기초하여 상기 활동 팝업 구성요소를 재실행하도록, 구성된다.

[0012] 선택적으로, 상기 재실행 모듈은 또한, 상기 onPause 메소드 또는 상기 onStop 메소드의 호출이 상기 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거되는지 여부를 결정하고; 상기 onPause 메소드 또는 상기 onStop 메소드의 호출이 상기 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거된 것이 아닌 경우, 상기 onPause 메소드 또는 상기 onStop 메소드에 기초하여 상기 활동 팝업 구성요소를 재실행하도록, 구성된다.

[0013] 선택적으로, 디바이스는, 상기 onPause 메소드 또는 상기 onStop 메소드의 호출이 상기 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거된 경우, 상기 활동 팝업 구성요소를 정상적으로 사용중지시키도록 구성된 사용중지 모듈을 더 포함한다.

[0014] 본 출원에서, 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 팝업을 출력하기 위해 활동 팝업 구성요소가 실행되고, 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성되어 있다. 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부가 결정된다. 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 활동 스택의 맨 위로의 활동 팝업 구성요소의 이동을 트리거하도록 활동 팝업 구성요소가 재실행되며, 그리하여 팝업이 시스템 프론트 엔드 상에 디스플레이된다. 따라서, 사용자에게 팝업을 출력하도록 활동 팝업 구성요소가 실행된 후에, 팝업이 가려지지 않으며 항상 시스템 프론트 엔드 상에 디스플레이되는 것이 보장될 수 있다.

도면의 간단한 설명

[0015] 도 1은 본 출원의 구현에 따른 안드로이드 기반의 팝업 프롬프트 방법을 예시한 흐름도이다.

도 2는 본 출원의 구현에 따른 안드로이드 기반의 팝업 프롬프트 디바이스를 예시한 논리 블록도이다.

도 3은 본 출원의 구현에 따른 안드로이드 기반의 팝업 프롬프트 디바이스를 포함하는 단말기의 하드웨어를 예시한 구조도이다.

발명을 실시하기 위한 구체적인 내용

[0016] 관련 기술에서, 안드로이드 시스템은 보통 다음 구현에서 사용자에게 팝업을 출력할 수 있다.

[0017] 구현에서, 안드로이드 시스템은 윈도우 구성요소(window component)를 사용함으로써 팝업을 출력할 수 있다.

[0018] 윈도우 구성요소를 사용함으로써 출력된 팝업은 임의의 페이지 상에 출력될 수 있고 또다른 페이지에 의해 가려지지 않는다. 그러나, 보통 사용자가 윈도우 구성요소를 사용함으로써 출력된 팝업을 구성하는 것이 허용된다.

따라서 사용자가 수동으로 시스템에서의 팝업을 금지할 때 시스템은 팝업을 출력할 수가 없다.

- [0019] 다른 구현에서, 안드로이드 시스템은 활동 구성요소(activity component)를 사용함으로써 팝업을 출력할 수 있다.
- [0020] 안드로이드 시스템에서, 활동 구성요소는 사용자와 상호작용하는 일을 담당한다. 구성요소는 상호작용 인터페이스를 제공하고 상호작용 인터페이스에 일부 상호작용 위젯(버튼 및 팝업과 같은)을 디스플레이할 수 있다. 사용자는 단말기와 상호작용하도록 이들 상호작용 위젯을 동작시킬 수 있다. 단말기는 본 상호작용에 대응하는 태스크를 완료하기 위해, 백 엔드에서 이들 상호작용 위젯에 대한 사용자의 이벤트를 검출할 수 있다. 단말기는 활동 구성요소를 사용함으로써 팝업을 출력하며, 그리하여 시스템에서의 임의의 활동 구성요소는 임의의 페이지 상에 팝업을 출력하도록 팝업을 위한 메인 바디로서 사용될 수 있다. 그러나, 활동 구성요소는 보통 활동 스택에서 관리된다. 활동 구성요소가 활동 스택에서 또다른 활동 구성요소에 의해 가려지면, 즉 다른 활동 구성요소가 활동 스택의 맨 위에 있고 활동 구성요소가 더 이상 활동 스택의 맨 위에 있지 않으면, 활동 구성요소에 의해 출력된 팝업은 또다른 팝업 또는 인터페이스에 의해 가려지고 시스템 프론트 엔드 상에 디스플레이될 수 없으며, 시스템의 기본 기능 또는 APP의 사용에 영향을 미친다.
- [0021] 다른 구현에서, 안드로이드 시스템은 고정 페이지 상에 팝업을 출력할 수 있다.
- [0022] 그러나, 팝업이 고정 페이지 상에 출력되면, 이는 융통적이지 못하고 열악한 상호작용 경험을 가져다준다. 예를 들어, 팝업이 로그인 페이지 상에서 출력될 때, 출력된 팝업은 로그인 페이지 상에서만 디스플레이될 수 있다. 시스템이 사용자에게 팝업을 출력한 후에, 시스템 프론트 엔드가 또다른 페이지를 디스플레이하는 경우, 사용자가 능동적으로 로그인 페이지로 전환하지 않는 한, 사용자는 시스템에 의해 출력된 팝업을 볼 수 없다. 결과적으로, 시스템에 의해 출력된 팝업은 제 시간에 사용자에게 알릴 수 없다.
- [0023] 이러한 점에 비추어, 본 출원은 안드로이드 기반의 팝업 프롬프트 방법을 제공한다. 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 팝업을 출력하도록 활동 팝업 구성요소가 실행되고, 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성되어 있다. 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부가 결정된다. 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 활동 스택의 맨 위로의 활동 팝업 구성요소의 이동을 트리거하기 위해 활동 팝업 구성요소가 재실행되며, 그리하여 팝업이 시스템 프론트 엔드 상에 디스플레이된다. 본 출원에서 활동 팝업 구성요소가 단일 태스크 모드로 있기 때문에, 활동 팝업 구성요소는 재실행된 후에 활동 스택의 맨 위로 즉시 이동될 수 있다. 따라서 활동 팝업 구성요소가 사용자에게 팝업을 출력하도록 실행된 후에, 팝업이 가려지지 않으며 항상 시스템 프론트 엔드 상에 디스플레이되는 것이 보장될 수 있다.
- [0024] 또한, 본 출원에서, 팝업은 더 이상 안드로이드 시스템에서의 윈도우 구성요소에 기초하여 출력되지 않고, 사용자가 더 이상 팝업을 구성하는 것이 허용되지 않으며, 이는 사용자가 시스템에서의 팝업을 수동으로 금지하기 때문에 팝업이 출력될 수 없었던 문제를 피한다.
- [0025] 또한, 본 출원에서는 팝업이 활동 팝업 구성요소를 사용함으로써 출력되기 때문에, 시스템에서의 임의의 활동 구성요소가 임의의 페이지 상에 팝업을 출력하도록 메인 바디로서 사용될 수 있으며, 높은 융통성을 달성한다.
- [0026] 본 출원은 특정 응용 시나리오를 참조하여 특정 구현을 사용함으로써 아래에 기재된다.
- [0027] 도 1을 참조하면, 도 1은 본 출원의 구현에 따른 안드로이드 기반의 팝업 프롬프트 방법을 도시한다. 방법은 단말기에 적용되고 다음 단계들을 포함한다.
- [0028] 단계 101: 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 팝업을 출력하기 위해 활동 팝업 구성요소를 실행하는데, 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성된다.
- [0029] 본 구현에서, 단말기는 안드로이드 시스템을 실행하는 지능형 단말기, 예를 들어 스마트폰 또는 태블릿 컴퓨터와 같은 이동 단말기를 포함할 수 있다.
- [0030] 팝업에 의해 프롬프트되는 트리거 이벤트는 단말기 시스템에 의해 검출된 이벤트를 포함할 수 있고, 팝업을 출력함으로써 사용자에게 프롬프트되어야 한다. 팝업에 의해 프롬프트되는 트리거 이벤트는 상이한 응용 시나리오에서 상이한 유형의 이벤트들을 포함할 수 있다.
- [0031] 예를 들어, 단일 디바이스 로그인 응용 시나리오에서, 팝업에 의해 프롬프트되는 트리거 이벤트는 단일 디바이스 로그인 이벤트를 포함할 수 있다. 이러한 응용 시나리오에서, 사용자가 동일 로그인 계정을 사용함으로써

또다른 단말기에 로그인할 때, 현재 단말기는 로그인 상태를 종료할 수 있고, 현재 로그인 상태를 종료하는 이유로 사용자에게 프롬프트하도록 사용자에게 팝업을 출력할 수 있다.

- [0032] 또다른 예를 들어, 착신 호 응답 응용 시나리오에서, 팝업에 의해 프롬프트되는 트리거 이벤트는 착신 호 이벤트를 포함할 수 있다. 착신 호를 검출할 때, 단말기 시스템은 사용자에게 팝업을 출력할 수 있다. 이 경우에, 사용자는 착신 호에 응답하거나 거절하도록 팝업에 대한 대응하는 동작(예컨대, 위로 또는 아래로 슬라이딩)을 수행할 수 있다.
- [0033] 본 구현에서, 단말기는 여전히 미리 결정된 활동 팝업 구성요소를 사용함으로써 사용자에게 팝업을 출력할 수 있다. 팝업에 의해 프롬프트되는 트리거 이벤트를 검출한 후에, 단말기는 팝업을 출력하는 데에 사용되는 활동 인스턴스를 생성하도록 활동 팝업 구성요소를 실행할 수 있고, 그 다음 사용자에게 프롬프트 텍스트를 출력하기 위해 인스턴스에 기초하여 팝업에 프롬프트 텍스트를 채울 수 있다.
- [0034] 안드로이드 시스템에서, 활동 구성요소의 실행 모드(launchMode)는 보통 4가지 모드를 포함한다: 표준 모드, singleTop 모드, singleTask 모드(즉, 단일 태스크 모드), 및 singleInstance 모드.
- [0035] 활동 구성요소가 singleTask 모드에 있을 때, 활동 팝업 구성요소는 고유의 활동 인스턴스만 실행될 수 있게 한다.
- [0036] singleTask 모드에서, 활동 팝업 구성요소가 실행된 후에, 시스템은 활동 팝업 구성요소에 대응하는 활동 인스턴스가 활동 스택에 존재하는지 여부를 결정한다. 활동 팝업 구성요소에 대응하는 활동 인스턴스가 이미 현재 활동 스택에 존재하기 때문에, singleTask 모드에서, 시스템은 새로운 활동 인스턴스를 생성하지 않고, 활동 스택으로부터 활동 팝업 구성요소를 가릴 또다른 활동 구성요소를 제거하며, 기존의 활동 인스턴스에 대응하는 활동 팝업 구성요소를 다시 활동 스택의 맨 위로 이동시킨다. 이 경우, 활동 인스턴스에 기초하여 출력된 팝업이 시스템 프론트 엔드 상에 다시 디스플레이되며 가려지지 않는다.
- [0037] 따라서, 본 구현에서, 활동 팝업 구성요소에 의해 출력된 팝업이 또다른 팝업 또는 페이지에 의해 가려지지 않게 유지하기 위해, 활동 팝업 구성요소의 실행 모드는 초기 상태에서 singleTask 모드로 설정될 수 있다.
- [0038] 단계 102: 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부를 결정한다.
- [0039] 안드로이드 시스템에서, 활동 구성요소의 수명주기는 실행(running) 상태, 일시정지(pause) 상태, 정지(stop) 상태, 및 폐기(destroy)(사용중지(disable)) 상태를 포함한다.
- [0040] 실행 상태는, 활동 구성요소가 보이고 사용자와 상호작용할 수 있는 상태이다. 활동 구성요소가 실행된 후에, 초기 상태는 실행 상태이다. 이 경우, 활동 구성요소는 활동 스택의 맨 위에 배치되고 시스템 프론트 엔드(스크린의 프론트엔드) 상에 디스플레이된다.
- [0041] 일시정지 상태는, 활동 구성요소가 또다른 전체화면 크기가 아닌 또는 반투명 활동 구성요소에 의해 가려지며 더 이상 활동 스택의 맨 위에 있지는 않지만, 여전히 보이고 사용자와 상호작용할 수 있는 상태이다.
- [0042] 정지 상태는, 활동 구성요소가 또다른 활동 구성요소에 의해 완전히 가려지며 더 이상 활동 스택의 맨 위에 있지 않고 활동 구성요소의 상호작용 인터페이스가 완전히 숨겨진 상태이다.
- [0043] 폐기 상태는 활동 구성요소가 사용중지되어 있는 상태이다.
- [0044] 4개의 실행 상태는 활동 구성요소에 대한 사용자의 동작을 사용함으로써 전환될 수 있다.
- [0045] 안드로이드 시스템에서, 활동 구성요소가 실행된 후에, 사용자가 동작을 수행함으로써 새로운 활동 구성요소를 실행하는 경우, 새로운 활동 구성요소가 활동 스택의 맨 위로 이동되어 원래 활동 구성요소를 가리게 된다. 이 경우, 시스템은 원래 활동 구성요소를 일시정지 상태로 전환하도록 onPause 메소드(method)를 호출한다.
- [0046] 원래 활동 구성요소가 완전히 가려질 때, 시스템은 원래 활동 구성요소를 정지 상태로 전환하도록 onStop 메소드를 호출한다.
- [0047] 원래 활동 구성요소가 일시정지 상태 또는 정지 상태에 있고 시스템 메모리가 불충분할 때, 시스템은 활동 구성요소를 사용중지하기 위해 활동 구성요소를 폐기 상태로 전환하도록 onDestroy 메소드를 호출한다.
- [0048] 따라서, 구현에서, 시스템은 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부를 결정하기 위해 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출되는지 여부를 결정할 수 있다.

- [0049] 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출될 때, 활동 팝업 구성요소가 현재 또다른 활동 구성 요소에 의해 가려지며 더 이상 스택의 맨 위에 있지 않다고 결정될 수 있다.
- [0050] 단계 103: 활동 팝업 구성요소가 또다른 활동 구성 요소에 의해 가려질 때, 활동 스택의 맨 위로의 활동 팝업 구성 요소의 이동을 트리거하도록 활동 팝업 구성 요소를 재실행하며, 그리하여 팝업이 시스템 프론트 엔드 상에 디스플레이된다.
- [0051] 본 구현에서, 활동 팝업 구성요소가 또다른 활동 구성 요소에 의해 가려진다고 결정할 때, 시스템은 활동 팝업 구성 요소를 재실행할 수 있다. 활동 팝업 구성요소가 이미 singleTask 모드로서 사전구성되어 있기 때문에, 활동 팝업 구성요소가 재실행된 후에, 시스템은 활동 스택으로부터 활동 팝업 구성 요소를 가릴 또다른 활동 구성 요소를 제거한다.
- [0052] 이 경우, 활동 팝업 구성요소가 다시 활동 스택의 맨 위로 이동되고, 활동 팝업 구성 요소의 고유의 활동 인스턴스에 기초하여 출력된 팝업이 다시 시스템 프론트 엔드 상에 디스플레이되며 가려지지 않는다.
- [0053] 구현에서, 시스템이 활동 팝업 구성요소의 onPause 메소드가 호출된다고 결정하는 경우, 시스템은 활동 팝업 구성요소가 또다른 활동 구성 요소에 의해 가려진다는 것을 확인할 수 있다.
- [0054] 이 경우, 활동 팝업 구성 요소를 재실행하기 위한 로직이 미리 onPause 메소드에서 로딩될 수 있다. 예를 들어, 개발자는 onPause() 메소드에서 활동 팝업 구성 요소를 재실행하는 데에 사용되는 실행 코드를 편집할 수 있다. onPause 메소드를 호출한 후에, 시스템은 활동 팝업 구성 요소를 재실행할 수 있다.
- [0055] 또다른 구현에서, 시스템이 활동 팝업 구성요소의 onStop 메소드가 호출된다고 결정하는 경우, 시스템은 활동 팝업 구성요소가 또다른 활동 구성 요소에 의해 가려진다는 것을 확인할 수 있다. 활동 팝업 구성 요소를 재실행하기 위한 로직이 또한 미리 onStop 메소드에서 로딩될 수 있다.
- [0056] 안드로이드 시스템에서, 시스템이 onPause 메소드 및 onStop 메소드를 호출할 때, 시스템은 보통 onStop 메소드 전에 onPause 메소드를 호출한다(호출 시퀀스는 안드로이드 시스템의 기본 프로세싱 메커니즘임). 그러나, 동시 팝업 시나리오와 같은 일부 응용 시나리오에서(즉, 활동 팝업 구성요소가 팝업을 출력할 때에 활동 팝업 구성 요소에 의해 출력되는 팝업이 또다른 활동 구성 요소의 팝업에 의해 가려짐), onPause 메소드 및 onStop 메소드를 호출하는 시퀀스 동안 시스템에 이상(abnormal) 거동이 발생할 수 있다. 구체적으로, 시스템은 onPause 메소드를 무시하면서 onStop 메소드를 바로 호출할 수 있다.
- [0057] 따라서, 이러한 이상 응용 시나리오에 대하여, onStop 메소드에서 로딩된 재실행 로직을 사용함으로써 활동 팝업 구성 요소를 반복으로 처리하는 것을 피하기 위해, 시스템이 활동 팝업 구성요소의 onStop 메소드가 호출된다고 결정할 때, 시스템은 활동 팝업 구성요소의 onPause 메소드가 호출되는지 여부를 더 결정할 수 있다. 활동 팝업 구성요소의 onPause 메소드가 이미 호출된 경우, 이는 활동 팝업 구성요소가 이미 onPause 메소드에서 로딩된 재실행 로직을 실행함으로써 처리되어 있음을 나타낸다. 이 경우, 시스템은 더 이상 onStop 메소드에서 로딩된 재실행 로직을 반복으로 실행하여 활동 팝업 구성 요소를 반복으로 처리하지 않을 수 있다.
- [0058] 물론, 활동 팝업 구성요소의 onPause 메소드가 호출되지 않은 경우, 시스템에서 이상 호출이 발생할 수 있고, 시스템은 활동 팝업 구성 요소를 재실행하도록 onStop 메소드에서 로딩된 재실행 로직을 정상적으로 실행시킬 수 있다.
- [0059] 활동 팝업 구성요소의 onPause 메소드가 이미 호출되어 있는지 여부를 결정하기 위해, 시스템이 활동 팝업 구성 요소의 클래스 파일에 마크를 추가하고 그 다음 클래스 파일에 대응하는 마크가 존재하는지 여부를 체크할 수 있다는 것을 주목할 만하다.
- [0060] 예를 들어, 구현 동안, 시스템이 onPause 메소드에서 로딩된 재실행 로직을 호출함으로써 활동 팝업 구성 요소를 완전히 처리한 후에, 시스템은 활동 팝업 구성 요소의 클래스 파일에 마크를 추가할 수 있다. 마크는 onPause 메소드에서 로딩된 재실행 로직에 기초하여 활동 팝업 구성요소가 이미 처리됨을 나타내는 데에 사용된다.
- [0061] 시스템이 활동 팝업 구성요소의 onStop 메소드가 호출되어 있다고 결정할 때, 활동 팝업 구성요소의 onPause 메소드가 이미 호출되어 있는지 여부를 확인하기 위해, 시스템은 활동 팝업 구성 요소의 클래스 파일에 마크가 존재하는지 여부를 더 체크할 수 있다. 클래스 파일에 마크가 존재하는 경우, 이는 onPause 메소드에서 로딩된 재실행 로직에 기초하여 활동 팝업 구성요소가 이미 처리되고 더 이상 반복으로 처리될 필요가 없음을 나타낸다.

- [0062] 메소드를 사용함으로써, onPause 메소드 및 onStop 메소드를 호출하는 시퀀스에서 이상이 발생하는 응용 시나리오에서, onStop 메소드에서 로딩된 재실행 로직을 사용함으로써 활동 팝업 구성요소가 반복으로 처리되는 것을 막을 수 있다는 것을 알 수 있다.
- [0063] 앞서 기재된 구현은, 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부를 결정하기 위해 시스템이 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출되는지 여부를 결정하는 상세한 프로세스를 기재한 것이다. 시스템이 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려진다고 결정할 때, 시스템은 onPause 또는 onStop 메소드에서 로딩된 재실행 로직을 실행함으로써 활동 팝업 구성요소를 재실행한다. 그리하여 활동 팝업 구성요소가 singleTask 모드로서 사전구성되어 있을 때, 활동 팝업 구성요소는 다시 활동 스택의 맨 위로 이동될 수 있고, 출력된 팝업은 시스템 프론트 엔드 상에 다시 디스플레이되며 가려지지 않을 수 있다.
- [0064] 그러나, 안드로이드 시스템에서, 시스템은 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때 뿐만 아니라, 사용자가 활동 팝업 구성요소를 사용중지하도록 정상적인 사용중지 이벤트를 실행할 때에도 onPause 메소드 또는 onStop 메소드를 호출한다.
- [0065] 예를 들어, 활동 구성요소가 실행(running) 상태에 있을 때, 시스템이 활동 구성요소에 대한 사용자의 사용중지 이벤트를 검출하는 경우, 시스템은 활동 구성요소를 일시정지 상태 및 정지 상태로 연속으로 전환하도록 onPause 메소드 및 onStop 메소드를 순차적으로 호출하고, 그 다음 활동 구성요소를 사용중지시키기 위해 활동 구성요소를 폐기 상태로 전환하도록 onDestroy 메소드를 자동으로 호출한다.
- [0066] 사용중지 이벤트는, 활동 팝업 구성요소에 대하여 사용자에게 의해 수행된 사용중지 동작에 대응하는 이벤트를 포함할 수 있다. 예를 들어, 안드로이드 시스템을 실행하는 터치스크린 스마트폰에 대하여, 사용중지 이벤트는 이동 전화 상의 백(back) 키에 대한 사용자의 탭 이벤트를 포함할 수 있다. 구체적으로, 사용자는 활동 팝업 구성요소를 사용중지하도록 이동 전화 상의 백 키를 탭할 수 있다. 백 엔드에서 백 키에 대한 사용자의 탭 이벤트를 검출할 때, 이동 전화 시스템은 활동 구성요소를 일시정지 상태 및 정지 상태로 연속으로 전환하도록 onPause 메소드 및 onStop 메소드를 순차적으로 호출할 수 있고, 그 다음 활동 구성요소를 사용중지시키도록 onDestroy 메소드를 자동으로 호출할 수 있다.
- [0067] 사용자가 활동 팝업 구성요소를 정상적으로 사용중지시키는 응용 시나리오에서, onPause 메소드 또는 onStop 메소드의 호출이, 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는 이벤트가 아니라, 사용자의 정상적인 사용중지 동작에 의해 트리거된다는 것을 알 수 있다. 따라서, 시스템이 활동 팝업 구성요소의 onPause 메소드 또는 onStop 메소드에서 로딩된 재실행 로직을 실행함으로써 활동 팝업 구성요소를 계속해서 재실행한다면, 사용자의 오동작(misoperation)이 야기된다(구체적으로, 사용자는 활동 팝업 구성요소를 사용중지하기를 기대하지만, 대신 재실행 로직을 사용함으로써 활동 팝업 구성요소를 재실행함).
- [0068] 따라서, 오동작을 피하기 위해, 시스템이 활동 팝업 구성요소의 onPause 메소드 또는 onStop 메소드가 호출된다고 결정할 때, 시스템이 onPause 메소드 또는 onStop 메소드에서 로딩된 재실행 로직에 기초하여 활동 팝업 구성요소를 재실행하기 전에, 시스템은 onPause 메소드 또는 onStop 메소드의 호출이 활동 팝업 구성요소의 사용중지 이벤트를 위해 사용자에게 의해 트리거되는 것인지 여부를 더 결정할 수 있다.
- [0069] onPause 메소드 또는 onStop 메소드의 호출이 활동 팝업 구성요소의 사용중지 이벤트를 위해 사용자에게 의해 트리거되는 것인 경우, 이는 활동 팝업 구성요소에 대한 정상적인 사용중지 프로시저이고, 시스템은 활동 팝업 구성요소가 정상적으로 사용중지되게 하고, 활동 팝업 구성요소에 대한 정상적인 사용중지 프로시저에 대응하며, 활동 팝업 구성요소를 사용중지시키도록 활동 팝업 구성요소의 onDestroy 메소드를 자동으로 호출할 수 있다.
- [0070] 물론, onPause 메소드 또는 onStop 메소드의 호출이 활동 팝업 구성요소의 사용중지 이벤트를 위해 사용자에게 의해 트리거되는 것이 아닌 경우, 시스템은 활동 팝업 구성요소를 재실행하도록 onPause 메소드 또는 onStop 메소드에서 로딩된 재실행 로직을 정상적으로 실행시키고 활동 팝업 구성요소를 활동 스택의 맨 위로 이동시킬 수 있다.
- [0071] onPause 메소드 또는 onStop 메소드의 호출이 활동 팝업 구성요소의 사용중지 이벤트를 위해 사용자에게 의해 트리거되는 것인지 여부를 결정하기 위해, 시스템이 활동 팝업 구성요소의 클래스 파일에 마크를 추가하고 그 다음 클래스 파일에 대응하는 마크가 존재하는지 여부를 체크할 수 있다는 것을 주목할 만하다.
- [0072] 예를 들어, 구현 동안, 시스템이 onPause 메소드 및 onStop 메소드를 호출하도록 활동 팝업 구성요소에 대한 사

용자의 사용중지 이벤트를 검출한 후에, 시스템은 활동 팝업 구성요소의 클래스 파일에 마크를 추가할 수 있다. 마크는 onPause 메소드 및 onStop 메소드의 호출이 활동 팝업 구성요소의 사용중지 이벤트에 대하여 사용자에게 의해 트리거되는 것임을 나타내는 데에 사용된다.

- [0073] 시스템이 onPause 메소드 또는 onStop 메소드가 호출된다고 결정할 때, 시스템은 활동 팝업 구성요소의 클래스 파일에 마크가 존재하는지 여부를 더 체크할 수 있다. 클래스 파일에 마크가 존재하는 경우, 이는 onPause 메소드 및 onStop 메소드의 호출이 활동 팝업 구성요소의 사용중지 이벤트를 위해 사용자에게 의해 트리거되는 것임을 나타낸다. 이 경우, 시스템은 활동 팝업 구성요소에 대한 정상적인 사용중지 프로시저에 대응할 수 있다.
- [0074] 메소드를 사용함으로써, 사용자가 활동 팝업 구성요소를 정상적으로 사용중지하는 응용 시나리오에서, 활동 팝업 구성요소가 재실행될 때, 활동 팝업 구성요소의 onPause 메소드 또는 onStop 메소드에서 로딩된 재실행 로직을 실행함으로써 야기되는 오동작의 문제를 피할 수 있다는 것을 알 수 있다.
- [0075] 앞서 기재된 구현에서의 기술적 해결책은 특정 응용 시나리오를 참조하여 아래에 상세하게 기재된다.
- [0076] 본 구현에서 보여주는 응용 시나리오는 단일 디바이스 로그인 응용 시나리오 및 착신 호 응답 응용 시나리오를 포함할 수 있다.
- [0077] 단일 디바이스 로그인 응용 시나리오에서, 팝업에 의해 프롬프트되는 트리거 이벤트는 단일 디바이스 로그인 이벤트를 포함할 수 있다. 응용 시나리오에서, 사용자는 로그인 계정을 사용함으로써 단말기에 로그인할 수 있고, 단말기에 대응하는 로그인 서버는 로그인 계정을 검증할(verify) 수 있다. 로그인 서버가 로그인 계정을 성공적으로 검증한 후에, 사용자는 단말기에 성공적으로 로그인한다.
- [0078] 사용자가 동일 로그인 계정을 사용함으로써 또다른 단말기에 로그인할 때, 로그인 서버는 로그인 상태를 종료하도록 현재 로그인 단말기를 트리거하기 위해 현재 로그인 단말기에 통지 메시지를 보낸다. 서버에 의해 보내진 통지 메시지를 수신한 후에, 현재 로그인 단말기는 단일 디바이스 로그인 이벤트가 검출된 것으로 간주할 수 있다. 이 경우에, 현재 로그인 단말기는 로그인 상태를 종료할 수 있고, 사용자에게 단일 디바이스 로그인 팝업을 출력할 수 있다. 현재 로그인 상태를 종료하는 이유와 함께 사용자에게 프롬프트하도록 사용되는 프롬프트 텍스트가 팝업에 채워질 수 있다.
- [0079] 시스템이 단일 디바이스 로그인 팝업을 출력하도록 활동 팝업 구성요소를 실행할 때, 출력된 팝업이 또다른 팝업 또는 페이지에 의해 가려지는 것을 막기 위해, 시스템은 기본적으로 활동 팝업 구성요소의 실행 상태를 singleTask 모드로 설정할 수 있다.
- [0080] 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 시스템은 onPause 메소드 또는 onStop 메소드를 호출한다. 재실행 로직이 onPause 메소드 또는 onStop 메소드에서 미리 로딩된다. onPause 메소드 또는 onStop 메소드에서 재실행 로직을 실행한 후에, 시스템은 활동 팝업 구성요소를 재실행할 수 있다.
- [0081] 활동 팝업 구성요소가 singleTask 모드로서 사전구성되어 있기 때문에, 활동 팝업 구성요소가 재실행된 후에, 시스템은 활동 스택으로부터 활동 팝업 구성요소를 가릴 또다른 활동 구성요소를 제거하고, 활동 팝업 구성요소를 다시 활동 스택의 맨 위로 이동시킨다. 이 경우, 단일 디바이스 로그인 팝업이 시스템 프론트 엔드 상에 다시 디스플레이되며 또다른 팝업 또는 페이지에 의해 가려지지 않는다. 활동 팝업 구성요소를 가릴 다른 활동 구성요소가 활동 스택으로부터 직접 제거되기 때문에, 그 다른 활동 구성요소에 대응하는 상호작용 인터페이스는 디스플레이될 수 없고 사용자는 페이지 변화를 인지할 수 없다.
- [0082] 단일 디바이스 로그인 응용 시나리오에서, 메소드를 사용함으로써 단일 디바이스 로그인 팝업이 가려지기 때문에 사용자에게 프롬프트 정보가 제시간에 프롬프트될 수 없었던 문제를 피할 수 있다는 것을 알 수 있다.
- [0083] 착신 호 응답 응용 시나리오에서, 팝업에 의해 프롬프트되는 트리거 이벤트는 착신 호 이벤트를 포함할 수 있다. 응용 시나리오에서, 착신 호를 검출할 때, 단말기 시스템은 착신 호가 검출됨을 사용자에게 프롬프트하도록 사용자에게 착신 호 팝업을 출력할 수 있다. 착신 호의 발신자와 함께 사용자에게 프롬프트하도록 사용되는 프롬프트 텍스트가 팝업에 채워질 수 있다. 사용자는 착신 호에 응답하거나 또는 착신 호를 거절하도록 팝업에 대한 대응하는 동작, 예를 들어 위로 슬라이딩하거나 아래로 슬라이딩하는 동작을 수행할 수 있다.
- [0084] 시나리오에서, 착신 호를 검출할 때 시스템에 의해 출력되는 착신 호 팝업이 또다른 팝업 또는 페이지에 의해 가려지는 경우, 사용자가 착신 호에 응답하기 위한 착신 호 팝업에 대한 동작을 수행할 수 없거나, 또는 착신 호 팝업이 가려질 때 단말기의 (홈 키와 같은)키가 기능할 수 없다는 호환성 문제가 어찌된 야기된다.

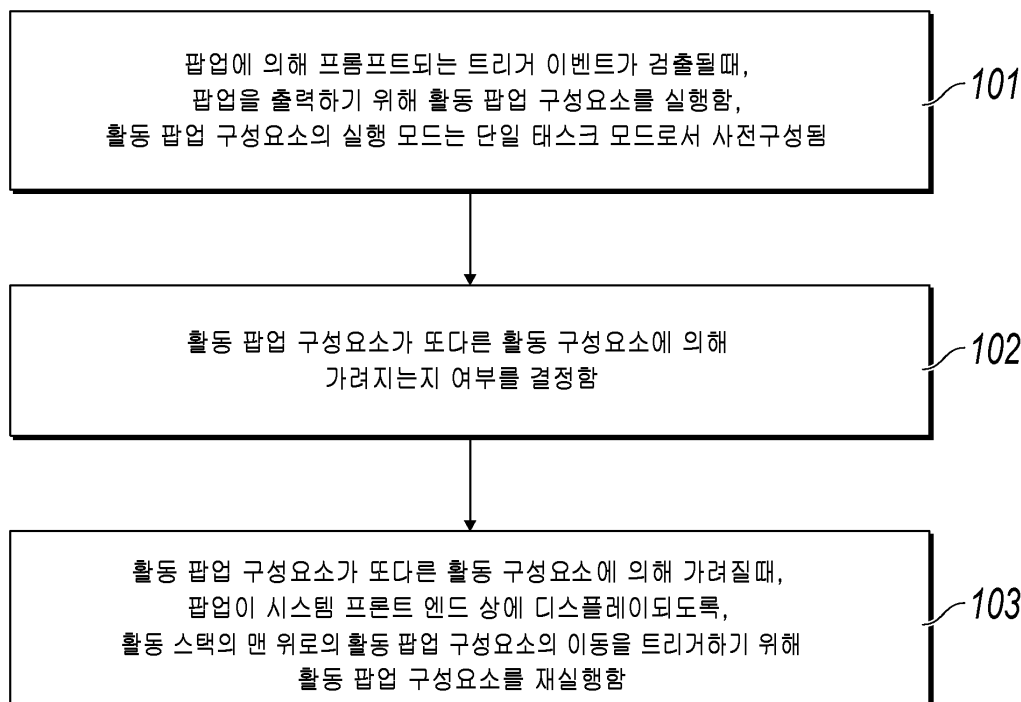
- [0085] 시스템이 착신 호 팝업을 출력하도록 활동 팝업 구성요소를 실행할 때, 출력된 팝업이 또다른 팝업 또는 페이지에 의해 가려지는 것을 막기 위해, 시스템은 기본적으로 활동 팝업 구성요소의 실행 상태를 singleTask 모드로 설정할 수 있다.
- [0086] 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 시스템은 onPause 메소드 또는 onStop 메소드를 호출한다. 재실행 로직이 onPause 메소드 또는 onStop 메소드에서 미리 로딩된다. onPause 메소드 또는 onStop 메소드에서 재실행 로직을 실행한 후에, 시스템은 활동 팝업 구성요소를 재실행할 수 있다.
- [0087] 활동 팝업 구성요소가 singleTask 모드로서 사전구성되어 있기 때문에, 활동 팝업 구성요소가 재실행된 후에, 시스템은 활동 스택으로부터 활동 팝업 구성요소를 가릴 또다른 활동 구성요소를 제거하고, 활동 팝업 구성요소를 다시 활동 스택의 맨 위로 이동시킨다. 이 경우, 착신 호 팝업이 다시 시스템 프론트 엔드 상에 디스플레이되며 또다른 팝업 또는 페이지에 의해 가려지지 않는다. 활동 팝업 구성요소를 가릴 다른 활동 구성요소가 활동 스택으로부터 직접 제거되기 때문에, 그 다른 활동 구성요소에 대응하는 상호작용 인터페이스는 디스플레이될 수 없고 사용자는 페이지 변화를 인지할 수 없다.
- [0088] 착신 호 응답 응용 시나리오에서, 착신 호가 팝업에 의해 가려지기 때문에 사용자가 착신 호에 응답할 수 없는 것, 단말기 키와의 호환성 문제를, 메소드를 사용함으로써 피할 수 있다는 것을 알 수 있다.
- [0089] 물론, 상기 기재된 응용 시나리오 외에도, 기재된 구현에서의 기술적 해결책은 또다른 유사 시나리오에 더 적용될 수 있다. 구체적으로, 실제 응용에서, 상기 기재된 응용 시나리오에 추가적으로, 기재된 구현에서의 기술적 해결책에 적용가능한 응용 시나리오는, 팝업이 사용자에게 프롬프트하도록 출력되어야 하고 출력된 팝업이 또다른 팝업 또는 페이지에 의해 가려질 때 시스템에서의 기본 기능 또는 APP의 사용에 어떠한 영향을 미치는 모든 응용 시나리오를 더 포함할 수 있다. 열거는 본 출원에서 생략된다.
- [0090] 앞서 기재된 구현에서, 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 활동 팝업 구성요소가 팝업을 출력하도록 실행되고, 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성되어 있다. 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부가 결정된다. 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 활동 스택의 맨 위로의 활동 팝업 구성요소의 이동을 트리거하도록 활동 팝업 구성요소가 재실행되며, 그리하여 팝업이 시스템 프론트 엔드 상에 디스플레이된다. 본 출원에서 활동 팝업 구성요소가 단일 태스크 모드로 있기 때문에, 활동 팝업 구성요소는 재실행된 후에 활동 스택의 맨 위로 즉시 이동될 수 있다. 따라서 활동 팝업 구성요소가 사용자에게 팝업을 출력하도록 실행된 후에, 팝업이 가려지지 않으며 항상 시스템 프론트 엔드 상에 디스플레이되는 것이 보장될 수 있다.
- [0091] 또한, 본 출원에서는, 팝업이 더 이상 안드로이드 시스템에서의 윈도우 구성요소에 기초하여 출력되지 않고, 사용자는 더 이상 팝업을 구성하는 것이 허용되지 않는다. 이는 사용자가 시스템에서의 팝업을 수동으로 금지하기 때문에 팝업이 출력될 수 없었던 문제를 피한다.
- [0092] 또한, 본 출원에서는 팝업이 활동 팝업 구성요소를 사용함으로써 출력되기 때문에, 시스템에서의 임의의 활동 구성요소가 임의의 페이지 상에 팝업을 출력하도록 메인 바디로서 사용될 수 있으며, 높은 융통성을 달성한다.
- [0093] 기재된 방법 구현에 대응하여, 본 출원은 디바이스 구현을 더 제공한다.
- [0094] 도 2를 참조하면, 본 출원은 단말기에 적용되는 안드로이드 기반의 팝업 프롬프트 디바이스(20)를 제공한다. 도 3을 참조하면, 안드로이드 기반의 팝업 프롬프트 디바이스(20)를 포함하는 단말기의 하드웨어 아키텍처는 보통 CPU, 메모리, 비휘발성 메모리, 네트워크 인터페이스, 내부 버스 등을 포함한다. 예로서 소프트웨어 구현을 사용함으로써, 안드로이드 기반의 팝업 프롬프트 디바이스(20)는 보통 메모리에 로딩된 컴퓨터 프로그램으로서 이해될 수 있다. CPU가 컴퓨터 프로그램을 실행한 후에, 하드웨어와 소프트웨어를 조합한 로직 디바이스가 형성된다. 디바이스(20)는, 팝업에 의해 프롬프트되는 트리거 이벤트가 검출될 때, 팝업을 출력하기 위해 활동 팝업 구성요소를 실행하도록 구성된 실행 모듈(201)로서, 활동 팝업 구성요소의 실행 모드는 단일 태스크 모드로서 사전구성되어 있는 것인, 실행 모듈(201), 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려지는지 여부를 결정하도록 구성된 결정 모듈(202), 및 활동 팝업 구성요소가 또다른 활동 구성요소에 의해 가려질 때, 팝업이 시스템 프론트 엔드 상에 디스플레이되도록, 활동 스택의 맨 위로 활동 팝업 구성요소의 이동을 트리거하기 위해 활동 팝업 구성요소를 재실행하도록 구성된 재실행 모듈(203)을 포함한다.
- [0095] 본 구현에서, 결정 모듈(202)은, 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출되는지 여부를 결정하고, 활동 팝업 구성요소의 onPause 또는 onStop 메소드가 호출될 때, 활동 팝업 구성요소가 또다른 활동 구성

요소에 의해 가려진다고 결정하도록 구성된다.

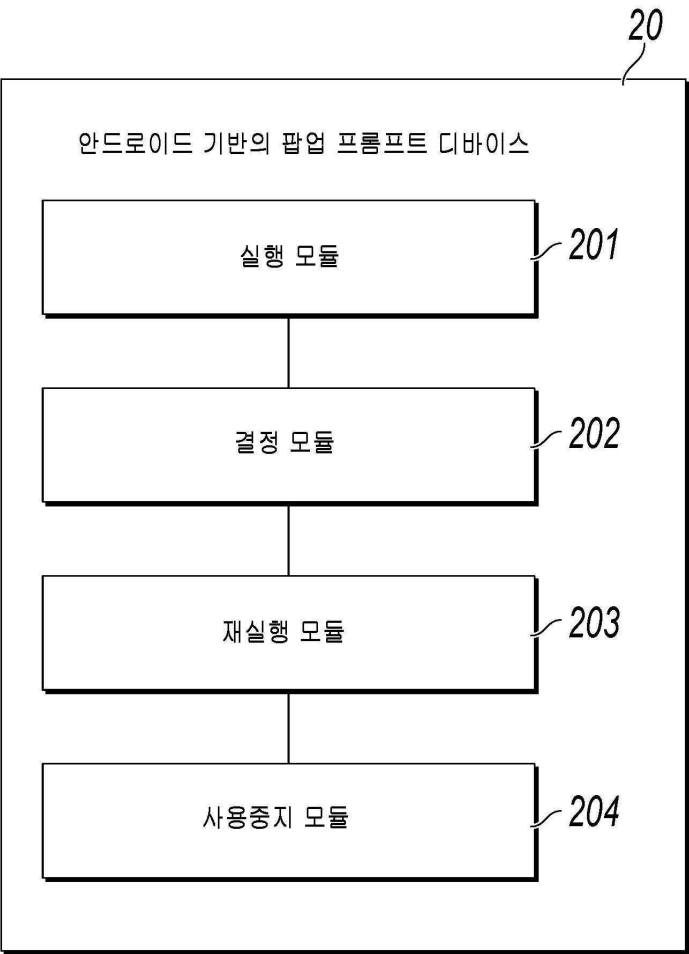
- [0096] 본 구현에서, 재실행 모듈(203)은, 활동 팝업 구성요소의 onPause 메소드가 호출될 때, onPause 메소드에 기초하여 활동 팝업 구성요소를 재실행하고, 활동 팝업 구성요소의 onStop 메소드가 호출될 때, 활동 팝업 구성요소의 onPause 메소드가 이미 호출되어 있는지 여부를 결정하고, onPause 메소드가 호출되지 않은 경우 onStop 메소드에 기초하여 활동 팝업 구성요소를 재실행하도록 구성된다.
- [0097] 본 구현에서, 재실행 모듈(203)은 또한, onPause 메소드 또는 onStop 메소드의 호출이 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거되는지 여부를 결정하고, onPause 메소드 또는 onStop 메소드의 호출이 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거된 것이 아닌 경우, onPause 메소드 또는 onStop 메소드에 기초하여 활동 팝업 구성요소를 재실행하도록 구성된다.
- [0098] 본 구현에서, 디바이스(20)는, onPause 메소드 또는 onStop 메소드의 호출이 활동 팝업 구성요소에 대한 사용중지 이벤트에 의해 트리거된 경우, 활동 팝업 구성요소를 정상적으로 사용중지시키도록 구성된 사용중지 모듈(204)을 더 포함한다.
- [0099] 당해 기술 분야에서의 숙련자는 본 명세서를 고려하고 여기에 개시된 본 개시를 실시한 후에 본 출원의 또다른 구현 솔루션을 쉽게 생각해낼 수 있다. 본 출원은 본 출원의 임의의 변형, 기능, 또는 적응적 변경을 커버하도록 의도된다. 이들 변형, 기능, 또는 적응적 변경은 본 출원의 일반적 원리에 따르며, 일반 지식 또는 본 출원에 개시되지 않은 기술 분야에서 일반적으로 사용되는 기술적 의미를 포함한다. 본 명세서 및 구현은 단지 예로서 간주되며, 본 출원의 실제 범위 및 사상은 다음 청구항에 의해 지적된다.
- [0100] 본 출원은, 위에 기재되며 첨부 도면에 도시된 정확한 구조에 한정되지 않고, 본 출원의 범위에서 벗어나지 않고서 수정 및 변경이 행해질 수 있다는 것을 이해하여야 한다. 본 출원의 범위는 첨부된 청구항에 의해서만 한정된다.
- [0101] 상기의 기재는 단지 본 출원의 구현의 예일 뿐이고, 본 출원을 한정하도록 의도되지 않는다. 본 출원의 사상 및 원리에서 벗어나지 않고서 행해지는 임의의 수정, 등가의 교체, 또는 개선은 본 출원의 보호 범위 내에 속할 것이다.

도면

도면1



도면2



도면3

