

# Flutter 프레임워크에서 UI/UX 연속성에 의한 사용성 증대 사례 연구

송영범, 이환용  
아주대학교

yeongbeom@ajou.ac.kr, hwan@ajou.ac.kr

## Case study on usability increase through UI/UX continuity in Flutter Framework

Yeongbeom Song, Hwanyong Lee  
Ajou University

### 요약

기존의 네이티브 애플리케이션은 플랫폼마다 마다 상이한 UI/UX 를 가지며, 별도의 소스코드로 빌드되기 때문에 동일한 UI/UX 를 제공하기 어렵다는 단점을 가진다. 하지만 Flutter 프레임워크 기반 애플리케이션은 하나의 소스코드로 여러 플랫폼의 애플리케이션을 빌드하기 때문에 모든 플랫폼에서 동일한 UI/UX 를 제공한다. 이를 검증하기 위해 Flutter 프레임워크 기반 애플리케이션 'ConnectIt'을 개발하여 네이티브 애플리케이션 'Instagram'을 비교하였으며, Flutter 프레임워크 기반 애플리케이션이 네이티브 애플리케이션 대비 UI/UX 연속성 유지가 용이하고 사용성 개선 효과가 존재함을 확인 하였다.

### I. 서론

최근 스마트기기의 확산으로 동일한 서비스를 여러 플랫폼에서 사용하는 경우가 빈번하다. 이때 동일한 서비스는 각 플랫폼에서 동일한 UI/UX 를 제공하여 UI/UX 연속성을 유지하는 것이 사용성 향상에 중요한 요소이다.[1] 하지만 네이티브 애플리케이션은 플랫폼 각각의 SDK 를 사용하므로 동일한 UI/UX 를 제공하는 것이 매우 어렵다. 따라서 본 논문에서는 크로스플랫폼 애플리케이션 개발 프레임워크 Flutter 를 통해 간단한 크로스플랫폼 애플리케이션을 개발하고, 이를 네이티브 애플리케이션과 비교하여 Flutter 프레임워크에서 UI/UX 연속성과 사용성을 개선하는 방법을 연구하였다.

### II. 본론

#### 2.1. 연구 방법 및 범위

본 연구는 Android/iOS 네이티브 애플리케이션 'Instagram'을 연구를 위해 Flutter 프레임워크로 개발된 크로스플랫폼 애플리케이션 'ConnectIt'과 비교하여 UI/UX 연속성을 관찰하고 사용성을 분석하였다.

이때 ConnectIt 개발에서 Flutter 프레임워크의 기본 디자인 스타일을 사용하여 추가 설정 없이 UI/UX 연속성을 유지하는지에 대해 연구하였다. 또한 플랫폼에 종속된 요소를 제외하기 위해 분석 범위를 각 플랫폼의 상단바와 하단바를 제외한 애플리케이션 화면으로 한정하였으며, 각 플랫폼에 대응된 두 애플리케이션의 UI 를 컴포넌트와 제스처를 중심으로 대조하였다.

이러한 플랫폼 별 UI/UX 비교를 통해 UI/UX 연속성이 유지되지 않는 사례와 UI/UX 연속성이 유지되는 사례를 설명하고 사용성을 분석하였다.

#### 2.2. 연구 진행 및 실험

Instagram 은 전반적으로 Android/iOS 각각의 UI/UX 가 유사하게 구성되어 있지만, 세부적인 컴포넌트와 제스처에서 차이를 보여 UI/UX 연속성이 유지되지 않고 있다.

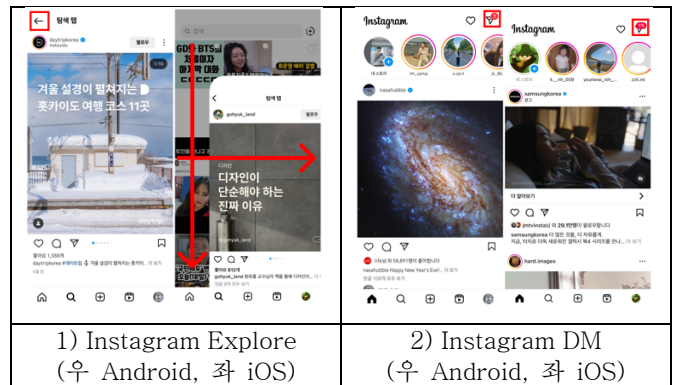


그림 1 플랫폼 별 Instagram UI/UX 비교

[그림 1]의 1 번은 Instagram 에서 사진 및 동영상 탐색을 위한 Explore 화면이다. Android 는 한 요소를 열람하고 본래 화면으로 돌아오기 위해서는 '뒤로가기' 버튼의 사용이 필수적이다. 하지만 iOS 는 현재 화면에서 '왼쪽 → 오른쪽', '위쪽 → 아래쪽' 방향으로의 스와이프 제스처를 통해 본래 화면으로 돌아올 수 있다.

[그림 1] 2 번은 Instagram 의 Home 화면으로 우측 상단의 DM 버튼을 통해 받은 메시지를 확인하고 DM 화면으로 이동할 수 있다. Android 의 경우 DM 버튼에 받은 메시지의 수가 단순히 숫자로 표시되었지만, iOS 의 경우 DM 버튼에 받은 메시지 수와 단위가 함께 표시된 것을 볼 수 있다. 이 외에도 플랫폼 별 세부적인 UI/UX 차이가 존재하며, 전체 내용은 [표 1]과 같다.

표 1 플랫폼 별 Instagram UI/UX 차이

	Android	iOS
Explore 이동	Back button	Swipe gesture
DM 알림	숫자만 표기	숫자와 단위 표기
상단바 정렬	주로 좌측	주로 중앙
기본 구성요소	Material	Cupertino

반면 ConnectIt 은 기본적으로 플랫폼마다 동일한 UI/UX 를 보이며, 세부 컴포넌트와 제스처에도 동일한 요소를 사용하여 UI/UX 연속성이 유지되고 있다.



그림 2 플랫폼 별 ConnectIt UI/UX 비교

[그림 2] 1 번은 ConnectIt 의 로그인 실패 알림 화면으로 로그인 실패 이유를 확인할 수 있다. 이때 Android/iOS 가 동일한 UI/UX 를 가지며, Android/iOS 모두가 Material 디자인 스타일의 아이콘과 컴포넌트(Dialog, TextButton, Checkbok 등)를 사용하는 것을 확인할 수 있다.

[그림 2] 2 번은 ConnectIt 의 보드 화면으로 타인의 포스트잇을 열람할 수 있다. 해당 화면에서도 Android/iOS 가 완벽히 동일한 UI/UX 를 제공하고 있으며, 동일하게 Android/iOS 모두가 Material 디자인 스타일의 아이콘과 컴포넌트(BottomTabNavigator, FloatingActionButton, Icon Button 등)를 사용하는 것을 확인할 수 있다.

### 2.3. 연구 결과 및 의의

앞서 2.2.절을 통해 네이티브 애플리케이션 대비 Flutter 기반 애플리케이션은 모든 플랫폼에서 동일한 UI/UX 를 가지는 것을 확인하였다. 이는 Flutter 기반 애플리케이션이 모든 플랫폼에서 동일한 소스코드를 사용하기 때문이며, 이러한 특성으로 Flutter 기반 애플리케이션은 기본적으로 UI/UX 연속성을 유지하고 있다.[2] 또한 Flutter 프레임워크는 [그림 3]과 같이 Material Widgets 와 Cupertino Widgets 를 동시에 지원하여 각 디자인 스타일을 적재적소에 활용할 수 있어 더 높은 UI/UX 연속성을 보유할 수 있도록 한다.[3]

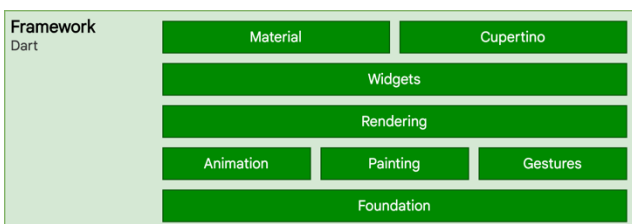


그림 3 Flutter Framework Architecture Diagram

이러한 높은 UI/UX 연속성은 사용자가 멘탈 모델을 활용하여 새로운 플랫폼에서 쉽게 동일 서비스에 적응하는 것을 돕는 효과를 가진다. 또한 이해 가능성, 학습성을 강화 시켜 의미성을 향상하고 예측 가능성, 친숙성, 일반화 가능성을 강화 시켜 일관성을 향상함을 통해, 사용성을 증대하는 효과를 보유하고 있다[4]. 이러한 세부 요소의 향상으로 사용자는 목적을 달성하기 위한 과정을 더욱 효율적으로 수행할 수 있다.

### III. 결론

본 연구에서는 Flutter 프레임워크 기반 크로스플랫폼 애플리케이션 'ConnectIt'을 개발하여 네이티브 애플리케이션 'Instagram'과 비교함을 통해 Flutter 기반 애플리케이션은 기본적으로 UI/UX 연속성을 유지하고 있으며, UI/UX 연속성 유지를 통해 사용성을 개선할 수 있다는 결과를 도출하였다.

Flutter 프레임워크는 하나의 소스코드로 여러 플랫폼의 애플리케이션을 빌드하기 때문에 하나의 서비스가 모든 플랫폼에서 동일한 UI/UX 를 보유하고 있다. 또한 Material/Cupertino 디자인 스타일을 동시에 지원하여 플랫폼에 종속되지 않은 UI/UX 설계를 할 수 있다. 이러한 장점으로 Flutter 프레임워크는 UI/UX 연속성을 강화 수 있으며, 이는 사용성 개선으로 이어진다.

결론적으로 Flutter 프레임워크를 사용한 서비스 개발은 크로스플랫폼 프레임워크의 장점인 개발 기간 단축, 개발 비용 절감과 더불어 UI/UX 연속성 유지에 의한 사용성 개선이라는 장점을 가지고 있음을 알 수 있다.

따라서 본 연구의 결과는 크로스플랫폼 프레임워크를 조직이나 기업에서 선택할 때 사용성 개선의 관점이라는 새로운 시각 또는 측면을 제공할 것으로 기대 된다.

### ACKNOWLEDGMENT

"본 연구는 2024 년 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업의 연구결과로 수행되었음"(2022-0-01077)

### 참 고 문 헌

[1] 신영미, 전수진, "크로스 플랫폼 환경에서 일관적인 사용자 경험을 위한 컨텍스트별 큐레이션 서비스 연구", 한국 HCI 학회 학술대회, 2017, pp. 796-799.

[2] Flutter, Multi-Platform, 2023 년 11 월 11 일 접속, <https://flutter.dev/multi-platform>.

[3] Flutter architectural overview, 2023 년 12 월 21 일 접속, <https://docs.flutter.dev/resources/architectural-overview>

[4] 김진우, Human Computer Interaction 개론, 안그래픽스, 2012.