



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0115467  
(43) 공개일자 2014년10월01일

(51) 국제특허분류(Int. Cl.)

G06F 9/44 (2006.01) G06F 3/048 (2006.01)  
G06F 3/14 (2006.01) G06F 15/16 (2006.01)

(21) 출원번호 10-2013-0029467

(22) 출원일자 2013년03월20일  
심사청구일자 없음

(71) 출원인

주식회사 넥스다임

경기도 안양시 동안구 학의로 268, 메가밸리 제 3층 제310호 (관양동)

(72) 발명자

김용호

경기도 안양시 동안구 관악대로404번길 10-24 301호

박승만

경기도 과천시 과천동 555-64 203호

전체 청구항 수 : 총 12 항

(54) 발명의 명칭 애플리케이션 환경 제공 시스템

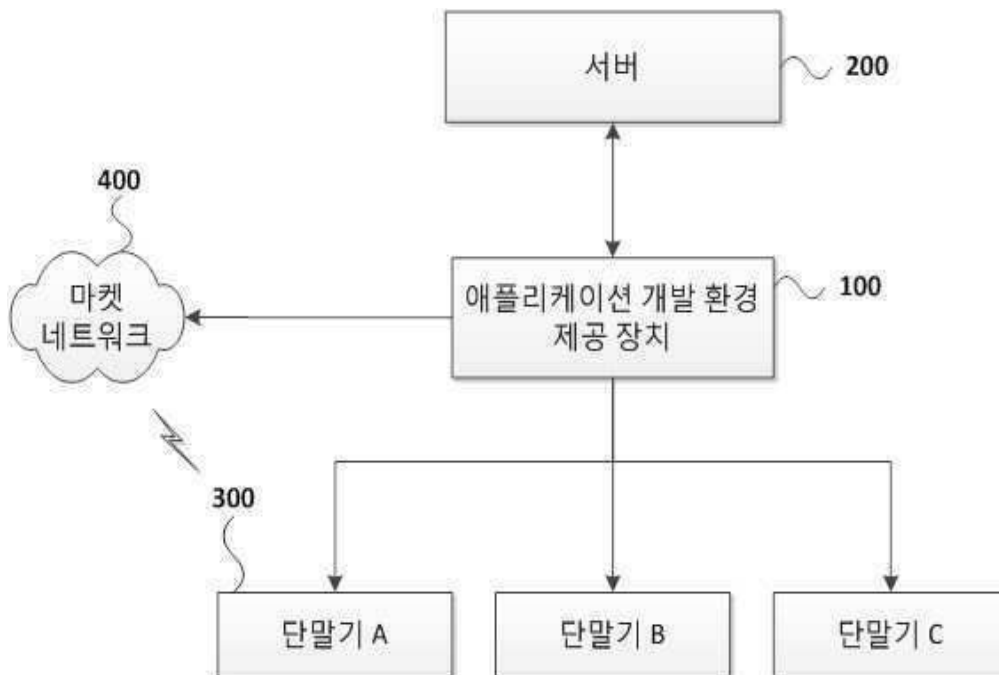
(57) 요약

본 발명의 실시예에 따른 애플리케이션 개발 환경 제공 시스템은

개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보를 하나 이상 입력하기 위한 애플리케이션 프로젝트 설정 화면을

(뒷면에 계속)

대표도 - 도2



표시하고, 애플리케이션을 구성하는 복수의 페이지 이미지와 상기 복수의 페이지 이미지를 연결하는 복수개의 화살표 이미지를 통해 상기 애플리케이션을 구성하는 복수의 페이지 이미지 사이의 연결관계를 나타내는 페이지 간 연결 영역과 상기 페이지 간 연결 영역 상에 페이지를 추가하고, 페이지들 간 연결 관계를 설정하기 위한 폼 탭 영역을 포함하는 제1화면을 표시하고, 상기 복수의 페이지들 중 어느 하나에 대하여 각 페이지 내에서 각 화면 또는 기능을 담당하는 복수의 컴포넌트들의 사이의 연결관계를 나타내는 제2 화면을 표시하는 표시부와, 상기 애플리케이션 프로젝트 설정 화면 상에 개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보가 하나 이상 입력되면, 입력된 정보에 부합되는 애플리케이션 프로젝트가 생성되도록 제어하며, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 연결되도록 제어하는 제어부와, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 화살표로 연결되면, 상기 제1컴포넌트와 제2컴포넌트간 연결에 대응하는 코드를 자동으로 생성하는 생성부를 구비하는 애플리케이션 개발 환경 제공 장치;

상기 애플리케이션 개발 환경 장치로 하나 이상의 컴포넌트를 제공하는 서버;를 포함하는 애플리케이션 개발 환경 제공 시스템이다.

---

## 특허청구의 범위

### 청구항 1

개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보를 하나 이상 입력하기 위한 애플리케이션 프로젝트 설정 화면을 표시하고, 애플리케이션을 구성하는 복수의 페이지 이미지와 상기 복수의 페이지 이미지를 연결하는 복수개의 화살표 이미지를 통해 상기 애플리케이션을 구성하는 복수의 페이지 이미지 사이의 연결관계를 나타내는 페이지 간 연결 영역과 상기 페이지 간 연결 영역 상에 페이지를 추가하고, 페이지들 간 연결 관계를 설정하기 위한 폼 탭 영역을 포함하는 제1화면을 표시하고, 상기 복수의 페이지들 중 어느 하나에 대하여 각 페이지 내에서 각 화면 또는 기능을 담당하는 복수의 컴포넌트들의 사이의 연결관계를 나타내는 제2 화면을 표시하는 표시부와, 상기 애플리케이션 프로젝트 설정 화면 상에 개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보가 하나 이상 입력되면, 입력된 정보에 부합되는 애플리케이션 프로젝트가 생성되도록 제어하며, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 연결되도록 제어하는 제어부와, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 화살표로 연결되면, 상기 제1컴포넌트와 제2컴포넌트간 연결에 대응하는 코드를 자동으로 생성하는 생성부를 구비하는 애플리케이션 개발 환경 제공 장치;

상기 애플리케이션 개발 환경 장치로 하나 이상의 컴포넌트를 제공하는 서버;를 포함하는 애플리케이션 개발 환경 제공 시스템.

### 청구항 2

제1항에 있어서,

상기 제어부는 상기 이벤트 정보와 상기 액션 정보 사이의 연결 관계를 표시하도록 상기 표시부를 제어하는 애플리케이션 개발 환경 제공 시스템.

### 청구항 3

제1항에 있어서,

상기 이벤트 정보는 복수의 이벤트 아이템을 포함하고, 상기 복수의 이벤트 아이템 중 적어도 하나는 상기 제2 컴포넌트에 연결되는 애플리케이션 개발 환경 제공 시스템.

### 청구항 4

제1항에 있어서,

상기 액션 정보는 복수의 액션 아이템을 포함하고,

상기 복수의 이벤트 아이템 중 적어도 하나는 상기 복수의 액션 아이템 중 어느 하나와 연결되는 애플리케이션 개발 환경 제공 시스템.

### 청구항 5

제1항에 있어서,

상기 표시된 제1 화면은 상기 복수의 페이지들 사이의 연결 관계를 방향성을 가지는 이미지 객체로 나타내는 애플리케이션 개발 환경 제공 시스템.

### 청구항 6

제5항에 있어서,

상기 이미지 객체가 변형 또는 삭제됨에 따라 상기 복수의 페이지들 사이의 연결 관계가 변경되는 애플리케이션 개발 환경 제공 시스템.

**청구항 7**

제1항에 있어서, 상기 표시부는,

상기 복수의 페이지들 중 특정 페이지의 세부 정보를 더 표시하고,

상기 세부 정보는 상기 특정 페이지에 포함된 컴포넌트들에 대한 식별 정보 및 상기 특정 페이지에서 선택된 컴포넌트에 대한 속성 정보를 포함하는 애플리케이션 개발 환경 제공 시스템.

**청구항 8**

제7항에 있어서,

상기 속성 정보는 상기 선택된 컴포넌트에 대한 기본 정보, 데이터 연결 정보, 스타일 정보 및 이벤트 정보 중 적어도 하나를 포함하는 애플리케이션 개발 환경 제공 시스템.

**청구항 9**

제1항에 있어서,

상기 표시된 제2 화면은 상기 복수의 컴포넌트들 사이의 연결 관계를 방향성을 가지는 이미지 객체로 나타내며, 상기 이미지 객체가 변형 또는 삭제됨에 따라 상기 복수의 컴포넌트들 사이의 연결 관계가 변경되는 애플리케이션 개발 환경 제공 시스템.

**청구항 10**

제1항에 있어서, 상기 표시부는,

상기 제2 컴포넌트 및 상기 액션 정보간의 관계를 나타내는 구조 화면을 표시하는 애플리케이션 개발 환경 제공 시스템.

**청구항 11**

제1항에 있어서, 상기 자동으로 생성되는 코드는,

중간언어로 구성되며,

상기 중간언어는,

복수의 컴포넌트 사이에 지정되는 이벤트 또는 액션간 연결관계 화면에 기초하여 생성되며,

상기 컴포넌트 및 상기 컴포넌트에 지정된 이벤트 또는 액션을 나타내는 복수의 엘리먼트 코드들을 포함하는 애플리케이션 개발 환경 제공 시스템.

**청구항 12**

제1항에 있어서, 상기 자동으로 생성되는 코드는,

중간언어로 구성되며,

상기 중간언어는,

상기 컴포넌트간 상대적인 배치 정보를 포함하며,

상기 배치 정보는 화면에 표시된 적어도 하나의 컴포넌트의 배치 상태에 따라 결정되는

애플리케이션 개발 환경 제공 시스템.

**명세서**

**기술분야**

본 발명은 애플리케이션 개발 환경 제공 방법에 관한 것이며, 더욱 상세하게는 애플리케이션 개발자의 개발

효율을 향상시킬 수 있는 애플리케이션 개발 환경 제공 방법에 관한 것이다.

### 배경 기술

- [0002] 최근, 다양한 종류의 스마트 폰이나 태블릿 PC와 같은 단말 장치들이 등장함에 따라 이러한 단말 장치들에서 사용할 수 있는 애플리케이션에 대한 관심이 증폭되고 있다.
- [0003] 특히, 최근에는 개발자들이 애플리케이션을 개발하여 소정의 애플리케이션 스토어에 업로드하면, 사용자들이 단말 장치를 이용하여 상기 애플리케이션 스토어에 접속한 후 자신이 원하는 애플리케이션을 구매하여 다운로드하는 방식이 널리 이용되고 있다. 이에 따라, 다양한 분야에서 애플리케이션 개발이 활발하게 이루어지고 있다.
- [0004] 이와 같이 다양하고, 활발한 애플리케이션 개발을 위해서, 일반적으로 C 언어, 자바(JAVA), OBDC, SQL 등을 다룰 줄 아는 전문가들은 직접적인 코드 작성 및 개발을 수행하고 있다.
- [0005] 도 1은 일반적인 애플리케이션 개발 환경을 나타내는 도면이다.
- [0006] 도 1에 도시된 바와 같이, 일반적인 애플리케이션 개발은 각각의 이벤트 또는 윈도우 메시지에 의한 단계별 매칭을 위해 Activity, Dialog, Frame 등과 같은 위젯들을 선언하고, 이를 상속받는 이벤트 프로시저(procedure)를 소정의 코드로 구현하는 방식에 의해 이루어지고 있다.
- [0007] 그러나, 이와 같은 개발 방식은 숙련된 개발자만이 접근할 수 있으며, 앞서 설명한 언어들에 대한 전문 지식들을 필요로 한다. 그리고, 개발자는 많은 양의 코드를 이벤트 프로시저를 통해 직접 구현해야 하므로 많은 지식과 상당한 노력을 들여 애플리케이션 개발을 수행하여야 하는 어려움이 있다.
- [0008] 또한, 개발된 코드에 의한 서비스 시나리오에 변경이 있는 경우, 개발자는 다시 많은 코드들을 변경해야 하는 문제점이 있다.
- [0009] 뿐만 아니라, 개발자마다 전문 지식의 정도 및 개인적인 코드 스타일 차이가 상이하여 단체협업 또는 개발자 교체시 모듈간 호환성 확보에 어려움이 있으며, 모듈화에 대한 이해도에 따라 코드가 복잡하게 생산될 수 있어 치명적인 버그가 다수 발생할 수 있다.
- [0010] 한편, 개발자가 동일한 기능을 갖는 애플리케이션을 개발하더라도 그 애플리케이션이 적용될 OS의 플랫폼에 따라 서로 다른 애플리케이션 개발 툴을 이용하여 개발을 수행해야 하는 문제가 있으며, 애플리케이션의 중복 개발에 따른 노력과 시간이 많이 소요될 수 있다.

### 발명의 내용

#### 해결하려는 과제

- [0011] 본 발명의 목적은, 개발자의 개발 효율을 향상시킬 수 있는 애플리케이션 개발 환경 제공 방법을 제공함에 있다.
- [0012] 또한, 복잡하고 전문적인 지식 없이도 시나리오 관리 및 애플리케이션 개발을 용이하게 할 수 있는 유저 인터페이스를 제공하는 애플리케이션 개발 환경 제공 방법을 제공함에 있다.
- [0013] 또한, 본 발명의 또 다른 목적은 모듈간 호환성을 유지하면서도 다양한 기기에 적응적으로 변환 적용할 수 있어 중복 개발을 방지할 수 있는 애플리케이션 개발 환경 제공 방법을 제공함에 있다.

#### 과제의 해결 수단

- [0014] 상기 목적을 달성하기 위한 본 발명의 실시예에 따른 애플리케이션 개발 환경 제공 시스템은
- [0015] 개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보를 하나 이상 입력하기 위한 애플리케이션 프로젝트 설정 화면을 표시하고, 애플리케이션을 구성하는 복수의 페이지 이미지와 상기 복수의 페이지 이미지를 연결하는 복수

개의 화살표 이미지를 통해 상기 애플리케이션을 구성하는 복수의 페이지 이미지 사이의 연결관계를 나타내는 페이지 간 연결 영역과 상기 페이지 간 연결 영역 상에 페이지를 추가하고, 페이지들 간 연결 관계를 설정하기 위한 폼 탭 영역을 포함하는 제1화면을 표시하고, 상기 복수의 페이지들 중 어느 하나에 대하여 각 페이지 내에서 각 화면 또는 기능을 담당하는 복수의 컴포넌트들의 사이의 연결관계를 나타내는 제2 화면을 표시하는 표시부와, 상기 애플리케이션 프로젝트 설정 화면 상에 개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보가 하나 이상 입력되면, 입력된 정보에 부합되는 애플리케이션 프로젝트가 생성되도록 제어하며, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 연결되도록 제어하는 제어부와, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 화살표로 연결되면, 상기 제1컴포넌트와 제2컴포넌트간 연결에 대응하는 코드를 자동으로 생성하는 생성부를 구비하는 애플리케이션 개발 환경 제공 장치;

[0016] 상기 애플리케이션 개발 환경 장치로 하나 이상의 컴포넌트를 제공하는 서버;를 포함하는 애플리케이션 개발 환경 제공 시스템이다.

### 발명의 효과

[0017] 본 발명의 실시예에 따르면, 애플리케이션 개발 환경을 제공할 때 복수의 페이지들간 연결 관계를 나타내는 제1 화면을 표시하고, 어느 하나의 페이지에 대하여 컴포넌트간 연결 관계를 나타내는 제2 화면을 표시하게 된다.

[0018] 특히, 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트를 연결하여 컴포넌트간 연결 관계를 직관적으로 표시할 수 있게 된다.

[0019] 한편, 컴포넌트간 연결 관계를 나타내는 이미지 객체에 대한 변형에 따라 연결 관계가 변경될 수 있어 전문적인 지식이 없는 일반 개발자라 하더라도, 컴포넌트간 연결관계 및 시나리오 구현을 용이하게 수행할 수 있게 된다.

[0020] 또한, 이와 같은 애플리케이션 개발을 위해 본 발명의 실시 예에 따른 중간 언어를 이용함으로써, 동일한 애플리케이션을 서로 다른 OS의 플랫폼으로 용이하게 변환 적용할 수 있게 된다. 따라서, 한 번의 애플리케이션 개발을 통해 다양한 OS에서 이용 가능한 애플리케이션을 개발할 수 있게 되므로 개발에 필요한 시간과 노력을 크게 저감시킬 수 있다.

[0021] 또한, 애플리케이션 개발을 용이하게 하기 위한 다양한 유저 인터페이스 및 그 동작을 제공할 수 있어, 개발자의 이용 편의성이 증대될 수 있다.

### 도면의 간단한 설명

[0022] 도 1은 일반적인 애플리케이션 개발 화면을 나타낸다.

도 2는 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치를 포함하는 전체 개발 시스템의 일 예를 개략적으로 나타낸 도면이다.

도 3은 본 발명의 일 실시 예에 따른 애플리케이션 개발환경 제공장치의 세부 구성을 나타낸다.

도 4 내지 도 7은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 방법을 설명하기 위한 도면들이다.

도 8은 본 발명의 실시 예에 따른 제2 화면과 실제 코드와의 관계를 나타내는 도면이다.

도 9은 본 발명의 일 실시 예에 따른 제2 화면(120)의 액션 리스트를 설명하기 위한 도면이다.

도 10는 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 컴포넌트 블록 표시 및 연결관계 표시 방법을 나타낸다.

도 11은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 컴포넌트 추가 방법을 설명하기 위한 도면이다.

도 12은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 데이터 바인딩(binding) 방

법을 나타낸다.

도 13는 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 관계형 데이터베이스 관리 시스템(RDBMS) 설정 방법을 나타낸다.

도 14은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 오픈 API형 데이터 설정 방법을 나타낸다.

도 15는 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 전문 Restful/SOAP 설정 방법을 나타낸다.

도 16 내지 도 25는 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 방법을 설명하기 위한 도면들이다.

도 26는 애플리케이션 개발 환경 제공 장치(100)가 타겟 단말기(300)에 구동 가능한 제2 언어로 변환하여 출력한 애플리케이션이 타겟 단말기(300)에서 실행된 화면을 나타낸다.

도 27 및 도 28은 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 제2 화면의 컴포넌트 배치 방법을 나타낸다.

도 29은 본 발명의 또 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 레이아웃 방법을 나타낸다.

도 29 내지 도 31은 블록 레이아웃의 실시 예들을 나타낸다.

도 32은 애플리케이션 개발 환경 제공 장치(100)의 플랫폼을 도식화한 블록도이며, 도 33는 애플리케이션을 실행할 수 있는 단말기(300)의 플랫폼을 도식화한 블록도이고, 도 34은 서버(300)의 아키텍처를 나타내는 블록도이다.

도 35는 본 발명의 실시 예에 따른 컴포넌트 구조도를 나타낸다.

도 36 내지 도 53는 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 방법에 따라 제공되는 화면들을 이용하여 애플리케이션이 생성되는 과정을 설명하기 위하여 참조되는 도면들이다.

### 발명을 실시하기 위한 구체적인 내용

[0023] 이하의 내용은 단지 본 발명의 원리를 예시한다. 그러므로 당업자는 비록 본 명세서에 명확히 설명되거나 도시되지 않았지만 본 발명의 원리를 구현하고 본 발명의 개념과 범위에 포함된 다양한 장치를 발명할 수 있는 것이다. 또한, 본 명세서에 열거된 모든 조건부 용어 및 실시예들은 원칙적으로, 본 발명의 개념이 이해되도록 하기 위한 목적으로만 명백히 의도되고, 이와 같이 특별히 열거된 실시예들 및 상태들에 제한적이지 않는 것으로 이해되어야 한다.

[0024] 또한, 본 발명의 원리, 관점 및 실시예들 뿐만 아니라 특정 실시예를 열거하는 모든 상세한 설명은 이러한 사항의 구조적 및 기능적 균등물을 포함하도록 의도되는 것으로 이해되어야 한다. 또한 이러한 균등물들은 현재 공지된 균등물뿐만 아니라 장래에 개발될 균등물 즉 구조와 무관하게 동일한 기능을 수행하도록 발명된 모든 소자를 포함하는 것으로 이해되어야 한다.

[0025] 따라서, 예를 들어, 본 명세서의 블록도는 본 발명의 원리를 구체화하는 예시적인 회로의 개념적인 관점을 나타내는 것으로 이해되어야 한다. 이와 유사하게, 모든 흐름도, 상태 변환도, 의사 코드 등은 컴퓨터가 판독 가능한 매체에 실질적으로 나타낼 수 있고 컴퓨터 또는 프로세서가 명백히 도시되었는지 여부를 불문하고 컴퓨터 또는 프로세서에 의해 수행되는 다양한 프로세스를 나타내는 것으로 이해되어야 한다.

[0026] 프로세서 또는 이와 유사한 개념으로 표시된 기능 블록을 포함하는 도면에 도시된 다양한 소자의 기능은 전용 하드웨어뿐만 아니라 적절한 소프트웨어와 관련하여 소프트웨어를 실행할 능력을 가진 하드웨어의 사용으로 제공될 수 있다. 프로세서에 의해 제공될 때, 상기 기능은 단일 전용 프로세서, 단일 공유 프로세서 또는 복수의 개별적 프로세서에 의해 제공될 수 있고, 이들 중 일부는 공유될 수 있다.

[0027] 또한 프로세서, 제어 또는 이와 유사한 개념으로 제시되는 용어의 명확한 사용은 소프트웨어를 실행할 능력을 가진 하드웨어를 배타적으로 인용하여 해석되어서는 아니되고, 제한 없이 디지털 신호 프로세서(DSP) 하드웨어, 소프트웨어를 저장하기 위한 롬(ROM), 램(RAM) 및 비 휘발성 메모리를 암시적으로 포함하는 것으로 이해

되어야 한다. 주지관용의 다른 하드웨어도 포함될 수 있다.

- [0028] 본 명세서의 청구범위에서, 상세한 설명에 기재된 기능을 수행하기 위한 수단으로 표현된 구성요소는 예를 들어 상기 기능을 수행하는 회로 소자의 조합 또는 펌웨어/마이크로 코드 등을 포함하는 모든 형식의 소프트웨어를 포함하는 기능을 수행하는 모든 방법을 포함하는 것으로 의도되었으며, 상기 기능을 수행하도록 상기 소프트웨어를 실행하기 위한 적절한 회로와 결합된다. 이러한 청구범위에 의해 정의되는 본 발명은 다양하게 열거된 수단에 의해 제공되는 기능들이 결합되고 청구항이 요구하는 방식과 결합되기 때문에 상기 기능을 제공할 수 있는 어떠한 수단도 본 명세서로부터 파악되는 것과 균등한 것으로 이해되어야 한다.
- [0029] 상술한 목적, 특징 및 장점은 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해질 것이며, 그에 따라 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 또한, 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에 그 상세한 설명을 생략하기로 한다.
- [0030] 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일 실시예를 상세히 설명하기로 한다.
- [0031] 도 2는 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)를 포함하는 애플리케이션 개발 시스템의 일 예를 개략적으로 나타낸 도면이다.
- [0032] 도 2를 참조하면, 본 발명의 일 실시 예에 따른 애플리케이션 개발 시스템은 애플리케이션 개발 환경 제공 장치(100), 서버(200), 단말기(300) 및 마켓 네트워크(400)를 포함한다.
- [0033] 애플리케이션 개발 환경 제공 장치(100)는 본 발명의 실시 예에 따라 애플리케이션을 구성하는 복수의 페이지들 사이의 연결관계 및 상기 복수의 페이지들 중 어느 하나에 대하여 복수의 컴포넌트들의 사이의 연결관계를 직관적으로 나타내는 화면들을 제공할 수 있다.
- [0034] 이를 위해, 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션을 구성하는 복수의 페이지들 사이의 연결관계를 나타내는 제1 화면을 표시하고, 상기 복수의 페이지들 중 어느 하나에 대하여 복수의 컴포넌트들의 사이의 연결관계를 나타내는 제2 화면을 표시하는 표시부 및 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 연결되도록 제어하는 제어부를 포함할 수 있다.
- [0035] 여기서, 애플리케이션은 애플리케이션 개발 환경 제공 장치(100) 또는 단말기(300)에서 실행하기 위한 애플리케이션 데이터를 의미할 수 있다. 그리고, 애플리케이션 데이터는 애플리케이션 개발 환경 제공 장치(100)에 의해 생성될 수 있으며, 단말기(300)로 직접 전송될 수 있다.
- [0036] 또한, 생성된 애플리케이션 데이터는 마켓 네트워크(400)로 업로드 될 수 있고, 업로드 된 애플리케이션 데이터는 구매 인증된 단말기(300)로 전송될 수도 있다.
- [0037] 그리고, 애플리케이션은 복수의 페이지들을 포함할 수 있으며, 각 페이지들은 복수의 컴포넌트를 포함할 수 있다.
- [0038] 애플리케이션 페이지는 단말기(300)에 의해 해석되어 실행되는 객체로서의 전자 문서 또는 그래픽 유저 인터페이스를 의미할 수 있다. 또한, 각 페이지에 포함된 복수의 컴포넌트들은 구조적으로 다른 페이지에 포함되는 컴포넌트들과 연결 관계를 형성할 수 있다.
- [0039] 이와 같은 연결 관계는, 특정 컴포넌트에 포함된 이벤트 정보가 다른 컴포넌트의 액션 정보와 연결되는 경우 형성될 수 있다.
- [0040] 보다 구체적으로, 애플리케이션이 실행되고, 상기 이벤트 정보에 기초하여 특정 컴포넌트의 이벤트가 발생하면, 이와 연결된 타 컴포넌트의 액션 정보에 기초하여 상기 타 컴포넌트에 대한 액션이 수행될 수 있다.
- [0041] 따라서, 단말기(300)는 애플리케이션이 실행되면, 복수의 페이지간 연결 관계 및 각 컴포넌트들간 연결 관계에 의해 다양한 동작을 수행할 수 있다. 그리고, 개발자는 이를 이용하여 다양한 애플리케이션 서비스를 제공할 수 있게 된다. 특히, 본 발명의 실시 예에 따르면, 이와 같은 페이지 및 컴포넌트간 연결 관계 설정을 위한 직관적이고 단순한 인터페이스를 제공함으로써 개발을 위한 시간 및 노력을 저감시킴은 물론이며 프로그래밍에 대한 전문 지식 없이도 애플리케이션 개발을 매우 용이하게 되는 효과가 있다.
- [0042] 한편, 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션 개발 환경을 통해 생성되는 애플리케이션 프로젝트를 중간 언어로 구성할 수 있으며, 중간 언어로 구성된 애플리케이션 프로젝트 또는 애플리케이션 코드를 타겟 단말기의 타겟 플랫폼 또는 OS에 따라 적응적으로 변환할 수 있다. 그리고, 애플리케이션 개발 환경 제

공 장치(100)는 타겟 단말기에서 실행 가능한 애플리케이션을 출력할 수 있다.

- [0043] 특히, 애플리케이션 개발 환경 제공 장치(100)는 기존의 정형화된 개발 패턴의 문제점과 코드 재사용성이 낮은 문제점들을 해결하기 위해, 컴포넌트 기반의 중간 언어로 구성되는 애플리케이션을 생성할 수 있다. 이에 따라 애플리케이션 개발자는 시나리오 기획, 서버 개발 및 클라이언트 개발을 동시적으로 수행할 수 있게 된다.
- [0044] 그리고, 애플리케이션 개발 환경 제공 장치(100)는 어플리케이션 개발을 위한 통합 플랫폼을 제공할 수 있으며, UX 컴포넌트 및 서버 컴포넌트 개발 및 템플릿 개발을 추가 할 수 있는 인터페이스를 제공할 수 있고, 누구나 쉽게 컴포넌트를 제작하고 쉽게 추가 할 수 있는 오픈 플랫폼을 제공할 수 있다. 따라서, 본 발명의 실시 예에 따르면 개발자 중심이 아닌 사용자 및 제품 중심으로서의 애플리케이션 개발 환경을 제공할 수 있게 된다.
- [0045] 한편, 서버(200)는 애플리케이션 개발 환경 제공 장치(100)와 연결될 수 있으며, 애플리케이션 개발 환경 제공 장치(100)에서 필요한 데이터 및 인증 정보 등을 제공할 수 있다.
- [0046] 또한, 서버(200)는 애플리케이션 개발 환경 제공 장치(100)의 버전 정보를 체크할 수 있으며, 업데이트를 제공할 수도 있다. 그리고, 서버(200)는 애플리케이션 개발 환경 제공 장치(100)에서 필요로 하는 데이터베이스 또는 WAS(Web Application Server)를 포함할 수 있다.
- [0047] 특히, 서버(200)는 애플리케이션 개발 환경 제공 장치(100)에서 사용될 수 있는 컴포넌트 데이터베이스를 포함할 수 있다. 그리고, 서버(200)는 특정 컴포넌트의 구매가 인증된 애플리케이션 개발 환경 제공 장치(100)로 해당 컴포넌트를 전송할 수 있다. 그리고 서버(200)는 컴포넌트의 해석을 위한 웹 컨테이너를 포함할 수도 있다.
- [0048] 한편, 각 단말기(300)는 애플리케이션 개발 환경 제공 장치(100)에서 생성된 애플리케이션이 실행 가능한 단말기일 수 있으며, 스마트 폰이나 태블릿 PC등이 예시될 수 있다.
- [0049] 특히, 각 단말기(300)의 플랫폼 또는 OS는 서로 다를 수 있다. 따라서, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 타겟 단말기를 식별하고, 중간 언어로 구성된 애플리케이션을 타겟 단말기에 해당하는 코드로 변환하여 제공할 수 있다. 그리고, 타겟 단말기는 변환된 애플리케이션을 수신하여 실행할 수 있게 된다. 이를 통해, 개발자는 동일 어플리케이션을 다른 플랫폼으로 변환하기 위한 시간과 노력을 크게 줄일 수 있으며, 타 OS에 대한 전문 지식 없이도 용이하게 변환 및 개발을 수행할 수 있게 된다.
- [0050] 도 3은 본 발명의 일 실시 예에 따른 애플리케이션 개발환경 제공장치의 세부 구성을 나타낸다.
- [0051] 애플리케이션 개발환경 제공장치의 세부 구성은 개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보를 하나 이상 입력하기 위한 애플리케이션 프로젝트 설정 화면을 표시하고, 애플리케이션을 구성하는 복수의 페이지 이미지와 상기 복수의 페이지 이미지를 연결하는 복수개의 화살표 이미지를 통해 상기 애플리케이션을 구성하는 복수의 페이지 이미지 사이의 연결 관계를 나타내는 페이지 간 연결 영역과 상기 페이지 간 연결 영역 상에 페이지를 추가하고, 페이지들 간 연결 관계를 설정하기 위한 폼 탭 영역을 포함하는 제1화면을 표시하고, 상기 복수의 페이지들 중 어느 하나에 대하여 각 페이지 내에서 각 화면 또는 기능을 담당하는 복수의 컴포넌트들의 사이의 관계를 화살표로 연결-이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 화살표로 연결-하여 나타내는 제2 화면을 표시하고, 상기 복수의 페이지 중 하나에 추가할 수 있는 미리 정의된 컴포넌트 리스트-복수의 컴포넌트 아이템들을 포함하며, 각각의 컴포넌트 아이템들은 미리 정의된 기능 또는 화면을 가지는 각 컴포넌트 코드에 대응하는 컴포넌트 리스트-를 표시하는 표시부(191)와, 상기 애플리케이션 프로젝트 설정 화면 상에 개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스정보가 하나 이상 입력되면, 입력된 정보에 부합되는 애플리케이션 프로젝트가 생성되도록 제어하며, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 연결되도록 제어하며, 상기 컴포넌트 아이템들 중 어느 하나가 선택되어 드래그 앤 드롭 방식으로 페이지 화면에 이동되면, 상기 이동된 컴포넌트 아이템에 대응되는 코드를 애플리케이션 코드에 삽입되도록 제어하는 제어부(196)와, 상기 제2 화면에서 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트가 화살표로 연결되면, 상기 제1컴포넌트와 제2 컴포넌트간 연결에 대응하는 코드를 자동으로 생성하는 생성부(193)로 구성되어 있다.
- [0052] 또한, 애플리케이션 개발환경 제공장치의 세부 구성 요소로써 애플리케이션 프로젝트 생성 화면에 대한 입력

을 통해 애플리케이션 프로젝트를 생성하고 애플리케이션 식별 정보(Application ID)를 입력하며 인증 정보 및 프로파일과 어플리케이션 이미지를 입력하는 입력부(192)가 있으며 상기 표시부를 통해 선택된 타겟 플랫폼이 구동되는 단말기의 해상도 정보를 식별하고, 상기 해상도 정보 및 상기 제2화면에 표시된 적어도 하나의 컴포넌트의 배치 상태에 대응하는 배치 정보에 기초하여 상기 컴포넌트들의 레이아웃을 결정한 후, 상기 결정된 레이아웃에 따라 상기 제1 코드를 상기 제2 코드로 변환하는 변환부(194) 서버와 애플리케이션 개발환경 제공장치간의 데이터 통신을 담당하는 통신부(195)로 구성된다. 상기에서 데이터 통신을 하는 데이터로는 생성된 제1 또는 제 2의 컴포넌트들도 포함된다.

- [0053] 이하에서는 도 4 내지 도 7을 참조하여 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 방법을 설명한다.
- [0054] 도 4은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 방법을 설명하기 위한 흐름도이다.
- [0055] 도 4을 참조하면, 먼저 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션 프로젝트를 생성한다(S101).
- [0056] 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션의 개발을 위한 애플리케이션 프로젝트를 생성할 수 있다. 애플리케이션 프로젝트는 개발 대상 애플리케이션을 구성하는 다양한 정보를 포함할 수 있다. 또한, 애플리케이션 프로젝트는 개발 대상 애플리케이션의 식별 정보, 버전 정보, 개발자 정보, 인증 정보, 이미지 정보, 데이터베이스 연결 정보, 서버 연결 정보, 모듈 정보 및 소스 정보를 포함할 수 있으며, 개발 대상 플랫폼에 대응하여 최적화된 정보를 포함할 수 있다.
- [0057] 이를 위해, 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션 프로젝트 생성 화면을 제공할 수 있다.
- [0058] 도 5는 애플리케이션 개발 환경 제공 장치(100)에 의해 제공되는 애플리케이션 프로젝트 생성 화면을 나타낸다.
- [0059] 도 5에 도시된 바와 같이, 개발자는 애플리케이션 프로젝트 생성 화면에 대한 입력을 통해 애플리케이션 프로젝트를 생성할 수 있다. 애플리케이션 프로젝트 생성 화면은 도 5와 같이 애플리케이션 식별 정보(Application ID) 입력부, 인증 정보 및 프로파일 입력부 및 어플리케이션 이미지 입력부를 포함할 수 있다. 이에 따라 개발자가 입력해야 하는 입력정보의 최소화 및 필요이미지의 최소화가 구현될 수 있다.
- [0060] 특히, 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션 프로젝트가 생성되고, 개발 대상 플랫폼이 설정되면, 이에 따라 애플리케이션 페이지의 자동 리사이징(Resizing) 및 자동 버전 관리를 수행할 수 있다.
- [0061] 이후, 애플리케이션 개발 환경 제공 장치(100)는 복수의 페이지들간 연결관계를 나타내는 제1 화면을 표시한다(S103).
- [0062] 애플리케이션 개발 환경 제공 장치(100)는 생성된 애플리케이션 프로젝트에 의해 개발 중인 애플리케이션의 페이지 정보를 제1 화면을 통해 표시할 수 있다. 그리고, 제1 화면은 각 페이지간의 연결 관계를 나타낼 수 있다.
- [0063] 도 6는 애플리케이션 개발 환경 제공 장치(100)에 의해 표시되는 제1 화면(110)을 나타낸다.
- [0064] 도 6에 도시된 바와 같이, 제1 화면(110)은 애플리케이션 프로젝트에서 개발 중인 애플리케이션에 포함되는 복수의 페이지들에 대한 연결 관계를 나타낼 수 있다. 각 페이지들은 단말기(300)에서 애플리케이션이 동작하는 경우 표시될 수 있는 각 전자 문서를 나타낼 수 있다. 그리고 애플리케이션 동작에 의해 단말기(300)에서 표시되는 전자 문서는 각각의 페이지들간의 연결 관계에 따라 전환될 수 있다. 이와 같은 전환 또는 이동은 앞서 설명한 컴포넌트의 이벤트 발생 및 이에 따른 액션 동작에 의해 수행될 수 있다.
- [0065] 예를 들어, 도 6와 같이 애플리케이션은 Home 페이지, Card 페이지, Find 페이지, Donut 페이지, SNS 페이지 및 View 페이지를 포함할 수 있다. 그리고, 제1 화면(110)은 각 페이지간 연결 관계를 나타내는 복수의 화살표 이미지(111)를 포함할 수 있다. 각 화살표 이미지(111)는 각 페이지들간의 이동 또는 전환관계를 나타낼 수 있다.
- [0066] 특히, 애플리케이션 개발자는 제1 화면(110)에 표시되는 각각의 화살표 이미지(111)를 변형 또는 삭제함으로써 페이지간 연결 관계를 원하는 대로 변경 할 수 있다. 예를 들어, 개발자는 새로운 화살표를 연결하여 페이지 사이를 연결시킬 수 있다. 또한, 기존의 화살표를 삭제하여 페이지 사이를 끊을 수 있다.
- [0067] 그리고, 애플리케이션 개발 환경 제공 장치(100)는 이와 같은 화살표 이미지의 변경 또는 삭제가 있는 경우,

이를 중간 언어로 구성된 애플리케이션 코드에 반영시킬 수 있다. 따라서, 애플리케이션 개발자는 코드 수정에 대한 전문적인 지식 없이도 각 페이지간의 연결 관계를 용이하게 파악하고, 수정할 수 있게 된다.

- [0068] 한편, 애플리케이션 개발 환경 제공 장치(100)는 제1 화면(110) 주변에 페이지간 연결 관계 설정 또는 페이지 추가를 위한 폼 탭(112)을 표시할 수 있다. 폼 탭(112)은 예를 들어, 연결 관계 설정 및 페이지 추가를 위해 미리 정의된 아이콘들을 포함할 수 있다.
- [0069] 그리고, 각 아이콘들은 드래그 앤 드랍 방식에 의해 이동될 수 있으며, 제1 화면(110)으로 이동되는 경우 해당 아이콘에 대응하여 애플리케이션 개발 환경 제공 장치(100)는 연결 관계 설정 또는 미리 정의된 페이지 추가를 수행할 수 있다.
- [0070] 다시 도 4를 설명한다.
- [0071] 이후, 애플리케이션 개발 환경 제공 장치(100)는 어느 하나의 페이지에 대하여 컴포넌트간 연결 관계를 나타내는 제2 화면을 표시한다(S105). 그리고, 애플리케이션 개발 환경 제공 장치(100)는 이벤트 정보를 포함하는 제1 컴포넌트와 액션 정보를 포함하는 제2 컴포넌트를 연결한다(S107).
- [0072] 앞서 설명한 바와 같이, 페이지는 복수의 컴포넌트들을 포함할 수 있다. 각 컴포넌트들은 페이지 내에서 각 화면 또는 기능을 담당하는 하나의 단위 객체를 의미할 수 있다. 그리고, 컴포넌트들은 이와 같은 화면 또는 기능을 담당하기 위한 이벤트 정보 또는 액션 정보를 포함할 수 있다.
- [0073] 이벤트 정보는 단말기(300)에 대해 미리 설정된 사용자 입력 또는 단말기(300) 동작에 의해 발생하는 이벤트들을 포함할 수 있다. 예를 들어, 이벤트들은 onClose, onInit, onScrollMouse, onClick, onError, onSuccess 또는 onEnd 이벤트 등을 포함할 수 있다.
- [0074] 그리고, 이벤트 각각은 다른 컴포넌트에 포함된 액션 정보에 대응될 수 있다. 액션 정보는 단말기(300)에서 발생된 이벤트에 따라 수행되는 애플리케이션의 동작에 대한 정보를 포함할 수 있다. 예를 들어 이벤트에 따라 수행되는 애플리케이션의 동작은 특정 컴포넌트의 실행, 숨김, 보임, 토글, 페이지 이동, clear명령, goHome명령, goBack명령, bind명령 또는 next명령 등을 포함할 수 있다.
- [0075] 이와 같이 각 컴포넌트에 포함되는 이벤트 정보 또는 액션 정보는 각각 대응될 수 있으며 이에 따른 연결 관계는 제2 화면에 표시될 수 있다.
- [0076] 도 7은 본 발명의 실시 예에 따른 제2 화면(120)을 나타낸다.
- [0077] 도 7에 도시된 바와 같이, 제2 화면(120)은 각 컴포넌트를 나타낼 수 있으며, 각 컴포넌트에 포함되는 이벤트 정보와 액션 정보간 연결 관계를 나타낼 수 있다.
- [0078] 특히, 제2 화면(120)에서 각 컴포넌트들은 각각의 블록 형태로 표시될 수 있다. 그리고, 각 컴포넌트에 포함되는 액션 정보는 블록의 일측에 텍스트 박스 형태로 표시될 수 있으며, 이벤트 정보는 블록의 타측에 텍스트 박스 형태로 표시될 수 있다.
- [0079] 그리고, 각 박스들간의 연결 관계가 화살표 이미지로 표시될 수 있다. 각각의 화살표 이미지는 특정 컴포넌트의 특정 이벤트로부터 시작될 수 있으며, 다른 컴포넌트의 특정 액션을 향할 수 있다. 이 경우, 화살표 이미지는 특정 이벤트가 발생하면 그 다음으로 실행될 컴포넌트의 액션임을 나타낼 수 있다.
- [0080] 앞서 제1 화면(110)에서 설명한 바와 동일하게, 애플리케이션 개발자는 제2 화면(120)에서 각 화살표 이미지를 변경 또는 삭제할 수 있고, 각 컴포넌트들은 드래그 앤 드랍 방식에 의해 추가 또는 삭제될 수도 있다. 이에 따라, 각 컴포넌트들의 이벤트와 액션간 연결 관계는 변경될 수 있으며, 애플리케이션 개발 환경 제공 장치(100)는 제2 화면(120)에 표시되는 컴포넌트와 화살표 이미지의 변경에 따라 중간 언어로 구성되는 애플리케이션의 코드를 생성 또는 삭제할 수 있다.
- [0081] 예를 들어, button1 컴포넌트 블록(121)은 onClick 이벤트 박스(123)를 포함할 수 있으며, onClick 이벤트 박스(123)는 화살표 이미지에 의해 link1 컴포넌트 블록(122)에 포함된 goBack 액션 박스(124)와 연결될 수 있다. 이 경우, 애플리케이션 개발 환경 제공 장치(100)는 button1 컴포넌트에 대한 onClick이벤트 발생시 link1 컴포넌트의 goBack 액션이 수행되도록 중간 언어로 코드를 생성하여 개발 중인 애플리케이션 코드에 삽입할 수 있다. 따라서, 개발자는 코드에 대한 이해 없이 유저 인터페이스 조작만으로도 컴포넌트간 연결 관계를 설정할 수 있게 된다.
- [0082] 도 8은 본 발명의 실시 예에 따른 제2 화면과 실제 코드와의 관계를 나타내는 도면이다.

- [0083] 도 8에 도시된 바와 같이, 본 발명의 실시 예에 따르면 애플리케이션 개발자는 제2 화면(120)에서 Button1 컴포넌트의 onClick 이벤트 발생시 그 다음 실행할 컴포넌트 messagebox1과, 그 컴포넌트가 가지고 있는 액션인 show를 화살표 이미지로 연결함으로써 지정할 수 있게 된다. 이 경우 도 8 하단 코드의 박스부분과 같은 코드를 애플리케이션 개발 환경 제공 장치(100)는 자동적으로 생성하여 애플리케이션 코드에 삽입할 수 있다. 따라서, 개발자는 JAVA 코드 작성으로 이벤트와 액션간의 연결 관계를 구현할 필요 없이 유저 인터페이스에 대한 조작만으로도 원하는 컴포넌트의 이벤트와 액션간 연결을 설정 할 수 있다.
- [0084] 도 9는 본 발명의 일 실시 예에 따른 제2 화면(120)의 액션 리스트를 설명하기 위한 도면이다.
- [0085] 도 9를 참조하면, 애플리케이션 개발 환경 제공 장치(100)는 이벤트와 액션간의 일대일 관계를 제2 화면(120)에서 화살표 이미지로 나타낼 수 있으나, 이는 한 번의 이벤트에 대응하여 많은 컴포넌트들의 액션을 실행하게 하는 데에는 제약이 따를 수 있다.
- [0086] 따라서, 본 발명의 실시 예에 따르면, 이를 해결하기 위해 제2 화면(120)은 액션 리스트 컴포넌트 블록(125)을 더 포함할 수 있다. 액션 리스트 컴포넌트 블록(125)은 하나의 이벤트에 대응하여 다수의 액션을 수행하기 위해 컴포넌트 사이에 삽입되는 중간 컴포넌트를 의미할 수 있다.
- [0087] 예를 들어, 개발자는 액션 리스트 컴포넌트 블록(125)을 생성하고, 한 번의 이벤트(onClink)에 대응하여 액션 리스트 컴포넌트 블록(125)의 execute 액션이 실행되도록 화살표 이미지를 연결함으로써 복수의 이벤트들(event1~event9)이 발생되도록 연결 관계를 설정할 수 있다. 그리고, 개발자는 각 복수의 이벤트들(event1~event9)에 의해 실행될 컴포넌트 및 액션을 다시 화살표 이미지를 이용하여 지정할 수 있다. 도 9에서는 event1에 대응하여 panel1 컴포넌트의 hide 액션이 수행되며, event2에 대응하여 messagebox1의 show 액션이 수행되도록 설정된 것을 나타내고 있다.
- [0088] 그리고, 애플리케이션 개발 환경 제공 장치(100)는 도 9 하단과 같이 제2 화면(120)에서 설정된 연결 관계가 애플리케이션에서 실행될 수 있도록 중간 언어로 구성된 코드를 생성할 수 있다.
- [0089] 이에 따라, 단말기(300)에서 어플리케이션이 실행되면, 제2 화면(120)에서 설정된 연결 관계에 의해, 사용자가 Button1을 Click하였을 때, panel1을 hide하고 messagebox1을 show하는 동작이 수행되게 된다.
- [0090] 한편, 도 10은 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 컴포넌트 블록 표시 및 연결관계 표시 방법을 나타낸다.
- [0091] 도 10을 참조하면, 애플리케이션 개발 환경 제공 장치(100)는 페이지 화면(130)과 함께 상기 페이지에 포함된 컴포넌트 블록들을 나타내는 컴포넌트 블록 화면(140)을 표시할 수 있다.
- [0092] 페이지 화면(130)은 각각의 컴포넌트들에 의해 구현된 페이지 이미지를 나타낼 수 있다.
- [0093] 그리고, 컴포넌트 블록 화면(140)은 컴포넌트와 이벤트 및 액션간의 연결 관계를 트리 구조로 나타낼 수 있다. 각 컴포넌트들은 1 depth로 표시될 수 있으며, 각각의 컴포넌트의 하위 depth에는 그 컴포넌트가 포함하는 액션들이 위치할 수 있다. 이에 따라, 개발자는 원하는 컴포넌트와 액션을 용이하게 선택할 수 있다.
- [0094] 그리고, 선택된 컴포넌트 또는 액션의 속성 정보가 컴포넌트 블록 화면(140)의 상단에 표시될 수 있다. 속성 정보는 각 컴포넌트 또는 액션의 식별 정보, 데이터 연결 정보 및 각 액션과 연결된 이벤트 정보를 포함할 수 있다.
- [0095] 그리고, 이와 같은 컴포넌트 블록 화면(140)과 페이지 화면(130)은 앞서 설명한 제2 화면(120)에 대응될 수 있다. 따라서, 애플리케이션 개발 환경 제공 장치(100)는 동일한 어플리케이션을 제2 화면(120) 또는 컴포넌트 블록 화면(140)과 페이지 화면(130) 중에서 선택하여 표시할 수 있다. 이를 위해, 애플리케이션 개발 환경 제공 장치(100)는 컴포넌트의 연결 관계 표시 방법에 대한 설정 메뉴를 제공할 수도 있다.
- [0096] 이와 같은 이벤트 및 액션간 연결 설정 방식은 기존 개발환경에서의 더블클릭 및 프로시저 생성을 통한 이벤트 처리 방법과 달리, 각 컴포넌트에 포함되는 이벤트 및 액션을 선택하는 방식일 수 있다. 따라서, 개발자는 특정 버튼을 눌렀을 경우 실행하고자 하는 컴포넌트 객체 및 그 객체에 포함된 액션을 선택하기만 하면 이벤트 및 액션간 연결 설정이 완료될 수 있다.
- [0097] 도 11은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 컴포넌트 추가 방법을 설명하기 위한 도면이다.
- [0098] 도 11을 참조하면, 애플리케이션 개발 환경 제공 장치(100)는 페이지 화면(151)에 추가할 수 있는 미리 정의

된 컴포넌트 리스트(150)를 표시할 수 있다. 그리고, 컴포넌트 리스트(150)는 복수의 컴포넌트 아이템들을 포함할 수 있다. 각각의 컴포넌트 아이템들은 미리 정의된 기능 또는 화면을 가지는 각 컴포넌트 코드에 대응될 수 있다.

- [0099] 그리고, 개발자는 이와 같은 컴포넌트 아이템들 중 어느 하나를 선택하여 드래그 앤 드랍 방식으로 페이지 화면(151)에 이동시킬 수 있다. 이에 대응하여, 애플리케이션 개발 환경 제공 장치(100)는 이동된 컴포넌트 아이템에 대응되는 코드를 애플리케이션 코드에 삽입함으로써 컴포넌트 추가를 용이하게 할 수 있다. 따라서, 개발자는 애플리케이션 개발시 컴포넌트에 대한 코드 작성 없이도 간단한 드래그 앤 드랍만으로 컴포넌트를 페이지에 추가할 수 있게 된다.
- [0100] 또한, 미리 정의된 컴포넌트들은 서버(200)로부터 수신 및 업데이트될 수 있다. 애플리케이션 개발 환경 제공 장치(100)는 서버(200)와 통신하여 새로운 컴포넌트들을 수신하고, 컴포넌트 리스트를 업데이트할 수 있다. 개발자는 업데이트된 컴포넌트 리스트로부터 드래그 앤 드랍으로 애플리케이션 페이지에 새로운 컴포넌트를 추가할 수 있다.
- [0101] 도 12은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 데이터 바인딩(binding) 방법을 나타낸다.
- [0102] 도 12에 도시된 바와 같이, 컴포넌트는 데이터 컴포넌트(dbtable1, data\_store)를 포함할 수 있다. 데이터 컴포넌트는 SQL 또는 RESTful/SOAP을 통한 웹 상의 DB와 연결할 수 있으며, 연결된 DB로부터 데이터를 수신하여 표시할 수 있는 객체를 나타낼 수 있다. 애플리케이션 개발 환경 제공 장치(100)는 데이터 컴포넌트에 대하여 DB 연결을 위한 바인딩을 제공할 수 있으며, 이에 따라 개발자는 SQL 또는 RESTful/SOAP에 대한 지식이 없이도 애플리케이션과 DB간의 데이터 바인딩을 수행할 수 있게 된다.
- [0103] 도 13는 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 관계형 데이터베이스 관리 시스템(RDBMS) 설정 방법을 나타낸다.
- [0104] 도 12에 도시된 바와 같이, 애플리케이션은 RDBMS와 연결될 수 있다. 따라서, 애플리케이션 개발 환경 제공 장치(100)는 RDBMS를 위한 설정 화면을 표시할 수 있다. 이를 위해, 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션에 대응되는 RDBMS 설정을 위한 데이터베이스 타입 정보, 호스트 정보, 카탈로그 정보, 유저 ID 정보, 패스워드 정보, 테스트 버튼, 프록시 URL 정보, 프록시 연결 테스트 버튼 및 SecretKeySpec 정보 설정 인터페이스를 포함할 수 있다.
- [0105] 도 14은 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 오픈 API형 데이터 설정 방법을 나타낸다.
- [0106] 도 14에 도시된 바와 같이, 컴포넌트는 오픈 API형 데이터와 연결될 수 있다. 예를 들어, 도14에 도시된 Twitter1 컴포넌트는 SNS의 일종인 트위터의 오픈 API와 연결될 수 있다. 이에 대한 설정을 위해, 애플리케이션 개발 환경 제공 장치(100)는 트위터 아이디를 설정하기 위한 인터페이스를 더 표시할 수 있다.
- [0107] 도 15는 본 발명의 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 전문 Restful/SOAP 설정 방법을 나타낸다.
- [0108] 도 15에 도시된 바와 같이, 애플리케이션 개발 환경 제공 장치(100)는 전문적인 Restful/SOAP 설정을 위한 Restful/SOAP 설정 화면을 제공할 수 있다. 이와 같은 애플리케이션 개발 환경 제공 장치(100)를 통해 전문적인 지식이 있는 사용자의 경우에는 세부적인 클래스 모듈간 설정 및 서버와의 연결 설정을 수행할 수 있게 된다.
- [0109] 이하에서는 도 16 내지 도 25를 참조하여 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 방법을 설명한다.
- [0110] 도 16는 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 방법을 나타낸다.
- [0111] 도 16를 참조하면, 먼저 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 중간 언어로 정의되는 적어도 하나의 컴포넌트를 화면에 표시하고(S201), 컴포넌트에 지정되는 이벤트 또는 액션에 따라 상기 중간 언어로 구성된 제1 코드를 생성한다(S203).
- [0112] 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 중간 언어로 정의되는 적어도 하나의 컴포넌트를 화면에 표시할 수 있다. 중간 언어는 애플리케이션을 타겟 단말기에 따라 적용하기 위한 XML방식의

언어를 포함할 수 있다. 또한, 중간 언어는 상술한 바와 같은 컴포넌트간 연결 관계 화면에 기초하여 생성될 수 있다. 이와 같이 본 발명의 실시 예에 따른 개발 환경 제공 장치(100)는 중간 언어를 활용함으로써 동일 어플리케이션을 다른 플랫폼에 적용하기 위한 중복 개발을 방지할 수 있다.

- [0113] 도 17은 본 발명의 실시 예에 따른 중간 언어와 일반적인 JAVA 코드와의 비교도이다.
- [0114] 도 17에 도시된 바와 같이, 코드가 많아지고 복잡해 질수록 일반적인 JAVA 코드와는 달리 중간 언어의 코드는 짧고 간결하게 작성될 수 있음을 알 수 있다. 또한, 이와 같은 중간 언어 코드는 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 컴포넌트간의 연결 관계 기반 애플리케이션 개발 환경에 의해 자동적으로 작성될 수 있어 개발자의 수고를 저감시킬 수 있다. 또한, 개발 중 서비스 시나리오 변경 및 유지 보수에도 상당한 이점을 얻을 수 있다.
- [0115] 도 18은 중간 언어로 정의되는 컴포넌트의 XML 포맷을 나타낸다.
- [0116] 도 18에 도시된 바와 같이, 컴포넌트는 중간 언어로 정의된 XML 코드로 구현될 수 있다. 특히, 중간 언어는 플랫폼 중립적이어야 하며, 최종단계에서는 각 플랫폼에 최적화된 코드 생성을 필요로 한다.
- [0117] 따라서, 본 발명의 실시 예에 따르면 도 18에 도시된 바와 같이 중간 언어에는 XML 엘리먼트들이 정의될 수 있으며, 애플리케이션 개발 환경 제공 장치(100)는 이 XML 엘리먼트들을 이용하여 최종 빌드 단계에서 각 플랫폼에 최적화된 코드를 생성할 수 있게 된다.
- [0118] 도 19 내지 24은 본 발명의 실시 예에 따른 컴포넌트별 XML 코드 유형 및 이에 따른 제1 화면 또는 제2 화면의 구성을 설명하기 위한 도면이다.
- [0119] 도 19은 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 기본 형태의 컴포넌트에 대한 XML코드를 나타낸다. 도 19에 도시된 바와 같이, 기본 형태의 컴포넌트는 빈 여백을 나타낼 수 있다.
- [0120] 도 20은 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 헤더 컴포넌트 및 풋터(footer) 컴포넌트를 나타낸다. 도 20에 도시된 바와 같이, 헤더 컴포넌트 및 풋터 컴포넌트는 각각 페이지의 상단에 표시되는 상단 바 또는 하단에 표시되는 하단 바에 대응될 수 있다.
- [0121] 한편, 도 21은 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 레이블, 텍스트박스 및 버튼 컴포넌트를 나타낸다. 도 21에 도시된 바와 같이, 각 레이블, 텍스트박스, 및 버튼 컴포넌트들은 각각 애플리케이션 페이지의 특정 위치와 XML코드에 대응될 수 있다.
- [0122] 그리고, 도 22은 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 컨테이너형 컴포넌트들을 나타낸다. 도 22을 참조하면, 컴포넌트가 다른 컴포넌트를 포함하는 경우, 그 컴포넌트는 컨테이너형 컴포넌트 일 수 있다. 예를 들어, Panel1 컴포넌트는 textbox1 컴포넌트 및 button1 컴포넌트를 포함할 수 있으므로, Panel1 컴포넌트는 컨테이너형 컴포넌트일 수 있다.
- [0123] 한편, 도 23는 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 리스트형 컴포넌트들을 나타낸다. 도 23를 참조하면, 예를 들어, 리스트형 컴포넌트는 복수의 컴포넌트 아이템들을 리스트로 나열하는 것을 나타낼 수 있다. 그리고, 도 23에 도시된 바와 같이, 애플리케이션 개발 환경 제공 장치(100)는 리스트 나열을 위해 리스트형 컴포넌트에 대응되는 XML코드를 생성할 수 있다.
- [0124] 그리고, 도 24은 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 논 비주얼(non-visual)컴포넌트들을 나타낸다. 도 24를 참조하면, 예를 들어 컴포넌트는 존재하나, 페이지에는 표시되지 않는 컴포넌트의 경우, 그 컴포넌트는 논 비주얼 컴포넌트 일 수 있다.
- [0125] 한편 컴포넌트는 도 25와 같은 구조를 가질 수 있다.
- [0126] 도 25를 참조하면, 예를 들어, 컴포넌트는 비주얼 컴포넌트, 컨테이너 컴포넌트, 콜렉션 컴포넌트 및 논비주얼 컴포넌트를 포함할 수 있다.
- [0127] 비주얼 컴포넌트는 일반적인 유저 인터페이스 컴포넌트(버튼, 라벨, 텍스트박스 등)를 의미할 수 있으며, 컨테이너 컴포넌트는 비주얼 컴포넌트를 포함하는 유저 인터페이스 컴포넌트(폼, 헤더, 풋터, 패널)를 의미할 수 있다. 그리고, 콜렉션 컴포넌트는 리스트를 표현하는 유저 인터페이스 컴포넌트(리스트뷰, 커버플로우, 네비게이션 바)를 의미할 수 있으며, 논비주얼 컴포넌트는 화면에 바로 보이지는 않지만 특정행위 또는 데이터 처리를 담당하는 컴포넌트(타이머, 링크, 메시지박스, db테이블)일 수 있다.

- [0128] 이와 같은 각 컴포넌트들은 이에 대응되는 XML 형식의 중간 언어로 구성될 수 있다. 그리고, 컴포넌트에 지정되는 이벤트 또는 액션에 따라 상기 중간 언어로 구성된 제1 코드는 타겟 플랫폼에 대응되는 제2 코드로 변환된다(S205).
- [0129] 이와 같이 중간 언어로 구성된 컴포넌트는 XML의 형태를 가질 수 있다. 예를 들어, 각 컴포넌트는 XML의 노드 단위를 나타낼 수 있다. 그리고, 각 컴포넌트는 qname 및 id attribute를 통해 선언될 수 있다. qname의 prefix는 namespace를 의미할 수 있으며, name은 컴포넌트 명을 가질 수 있다. 그리고, id attribute는 하나의 페이지에서 유일한 값을 가질 수 있으며, 영문자로 시작하되, 영문자와 숫자간 조합을 가질 수도 있다.
- [0130] 한편, 중간 언어의 최 상위 노드는 페이지를 나타낼 수 있다. 이에 따라, qname의 prefix는 페이지 엘리먼트의 xmlns: attribute에서 선언될 수 있다.
- [0131] 그리고, 한 화면을 구성하는 페이지는 한 개의 파일 단위로 존재할 수 있고, 각각의 페이지의 id 엘리먼트는 모든 페이지들에서 유일한 값을 가질 수 있다.
- [0132] 이와 같은 컴포넌트들의 액션 및 이벤트들은 각 선언된 컴포넌트 엘리먼트의 child 엘리먼트로 포함될 수 있다. 예를 들어, 컨테이너 컴포넌트는 비주얼 컴포넌트를 child 엘리먼트로 포함시킬 수 있다. 또한, 컬렉션 컴포넌트는 자신이 포함 가능한 아이텀 컴포넌트를 아이텀 엘리먼트의 형태로 포함할 수 있다. 그리고, 모든 컴포넌트들은 이벤트 및 액션 중 적어도 하나를 포함할 수 있다.
- [0133] 도 26는 애플리케이션 개발 환경 제공 장치(100)가 타겟 단말기(300)에 구동 가능한 제2 언어로 변환하여 출력한 애플리케이션이 타겟 단말기(300)에서 실행된 화면을 나타낸다.
- [0134] 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 중간 언어로 작성된 코드를 타겟 디바이스에 적합한 제2 언어로 변환할 수 있다. 예를 들어, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 iOS시리즈, Android시리즈, 타이젠 OS 등으로 중간 언어에서 제2 코드로의 직접 변환을 수행할 수 있다. 따라서, 개발 효율을 극대화 할 수 있게 된다.
- [0135] 또한, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 타겟 플랫폼에 따른 시뮬레이션 화면을 애플리케이션 개발 환경과 함께 제공할 수 있다. 이에 따라 개발자는 선택된 디바이스에서 구동되는지 여부를 미리 판단할 수 있게 된다.
- [0136] 한편, 도 27 및 도 28은 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)에서 제공되는 제2 화면의 컴포넌트 배치 방법을 나타낸다.
- [0137] 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 컴포넌트의 표시 및 각 이벤트 및 액션간의 연결 관계를 효과적으로 표시하기 위해, 컴포넌트 객체의 상단 영역에는 컴포넌트 객체의 인스턴스명, 객체명 및 아이콘을 배치할 수 있다. 이에 따라 컴포넌트의 구분이 용이하게 될 수 있다. 이는 도 27에 나타나 있다.
- [0138] 여기서, 컴포넌트 객체 내 일측에는 박스 형태의 액션을 배치할 수 있으며, 타측에는 박스 형태의 이벤트들을 배치할 수 있다. 그리고, 각 이벤트에서 액션으로 연결되는 연결 관계는 화살표 이미지의 방향으로 표시할 수 있다. 이에 따라 개발자는 일측에서부터 타측으로 이어지는 자연스러운 애플리케이션의 동작 흐름을 파악할 수 있게 된다.
- [0139] 한편, 본 발명의 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 컴포넌트의 표시 및 각 이벤트 및 액션간의 연결 관계를 효과적으로 표시하기 위해, 각 컴포넌트들을 정렬하여 표시할 필요성이 있다.
- [0140] 도 28에 도시된 바와 같이, 이와 같은 컴포넌트의 정렬을 위해 애플리케이션 개발 환경 제공 장치(100)는 최상위 폼 페이지를 왼쪽 상단 1순위로 배치하고, 버튼 및 메뉴 항목들을 2순위로 배치하며, 상기 버튼 및 메뉴 항목에 대응되는 링크(화살표 이미지)들을 3순위로 배치하고, 헤더와 풋터와 같은 이벤트에 대해 자주 사용되지 않는 항목들은 최하순위로 배치할 수 있다.
- [0141] 이와 같은 배치를 통해 도 28에 도시된 바와 같이, 왼쪽에서 오른쪽으로 이어지는 각 컴포넌트의 배치 흐름에 용이하게 될 수 있으며, 개발자가 일일이 컴포넌트를 재 배치해야 하는 수고를 저감시킬 수 있게 된다.
- [0142] 한편, 도 29는 본 발명의 또 다른 일 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 레이아웃 방법을 나타낸다.

- [0143] 기존의 하이브리드 앱 개발 도구들은 다양한 해상도를 지원하지 못하는 문제점이 있었다. 특히, 기존의 WYSIWYG 지원 앱 개발 도구들은 개발자의 드래그 앤 드롭을 통해 레이아웃 좌표를 적용하기 위해 X, Y 좌표만을 이용하여 UI컴포넌트를 배치 설계하고 있다. 그러나, 이는 단말의 해상도가 조금이라도 바뀌거나 가로 모드로 변경되면 이에 맞는 별도 화면을 각각 개발해야 하는 문제점이 있었다.
- [0144] 따라서, 도 29를 참조하면, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 블록단위를 이용하여 어떤 단말기든 어떤 해상도이든 동일하게 표현 될 수 있는 블록레이아웃 방식을 사용할 수 있다.
- [0145] 블록레이아웃 방식은 화면을 각 UI컴포넌트를 이용하여 배치하는 경우, X, Y의 좌표정보를 이용하는 것이 아니라, 컨테이너 컴포넌트를 기준으로 상하좌우에 정렬하는 방식을 의미할 수 있다. 이 경우, 해상도가 변경되어도 각 컴포넌트들은 항상 상하좌우 위치를 고수하게 되어 타겟 단말기마다 동일한 레이아웃을 표현할 수 있게 된다.
- [0146] 또한, 연속된 컴포넌트가 배열된 경우, 예를 들어, 버튼1, 버튼2 및 버튼 3을 배치하는 경우에, 각 버튼들의 사이즈만 정의하더라도 화면이 로테이션 되거나 해상도가 다른 디바이스에서도 동일한 레이아웃이 표현될 수 있다.
- [0147] 도 29 내지 도 31은 이와 같은 블록 레이아웃의 실시 예들을 나타낸다.
- [0148] 예를 들어, 도 29과 같이, 각 컴포넌트들 중 최상단에 헤더, 중간단에 페널1 및 하단의 풋터가 순차적으로 배치될 수 있다. 폼, 헤더, 풋터 및 페널들은 컨테이너 컴포넌트로서의 역할을 수행할 수 있고, 각 컨테이너들은 상하 배치를 담당할 수 있게 된다.
- [0149] 도 30는 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)가 텍스트박스 컴포넌트 및 버튼 컴포넌트에 대한 X,Y 좌표정보 없이 애플리케이션을 생성하는 방법을 설명하기 위한 도면이다.
- [0150] 도 30를 참조하면, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 X, Y 좌표 정보 없이 텍스트박스 컴포넌트 및 버튼 컴포넌트에 대한 width 및 height를 설정할 수 있다. 이 경우, 실제 화면에서 표시되는 경우에는 텍스트박스를 먼저 배치하고 바로 오른 쪽에 버튼을 두게 되어 있어 애플리케이션 개발 환경에 의해 제작된 애플리케이션 들은 X, Y 좌표변환 없이도 각 단말기에 적합한 해상도로 변환될 수 있다.
- [0151] 도 31은 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)의 컴포넌트 배치 방법을 나타낸다.
- [0152] 도 31에 도시된 바와 같이, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 페이지상의 하단 풋터 영역에 네비게이션 컴포넌트를 배치할 수 있다. 이 경우, 컨테이너 형식의 네비게이션 컴포넌트에 대해, 애플리케이션 개발 환경 제공 장치(100)는 그 Child 컴포넌트들을 타겟 단말기(300)의 해상도에 따라 크기 변경을 수행할 수 있다. 그리고, 크기 변경된 Child 컴포넌트들은 화면에 딱 차도록 표시될 수 있다.
- [0153] 한편, 도 32은 애플리케이션 개발 환경 제공 장치(100)의 플랫폼을 도식화한 블록도이며, 도 33는 애플리케이션을 실행할 수 있는 단말기(300)의 플랫폼을 도식화한 블록도이고, 도 34은 서버(300)의 아키텍처를 나타내는 블록도이다.
- [0154] 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 각각의 개발자 기호 및 목적에 따라 다양한 deploment 방식을 지원할 수 있으며, 이를 위해 다양한 시스템 프레임워크를 탑재할 수 있다. 본 발명의 실시 예에 따르면, iOS의 경우, xarchive, ipa, Xcode project 3가지 타입 지원 및 인증서 signing까지 자동화를 지원할 수 있다. 그리고, Android의 경우 apk, ADT project 2가지 타입 지원 및 인증서 생성 및 signing 까지 자동화를 지원할 수 있고, Data Server를 통해 WAR 패키징 및 ftp를 통한 서버 업로드를 지원할 수 있다.
- [0155] 한편, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 진정한 의미의 Once Source, Multi Platform을 구현하기 위해, Webkit 엔진 내장 및 OSX, Linux, Windows에도 모두 동일하게 개발될 수 있도록 설계될 수 있다. 한편, 애플리케이션 개발 환경 제공 장치(100)는 eclipse와는 차별화 하여 개발자로 하여금 별도의 설정 및 WAS와 같은 별도의 개발자적 설치사항이 필요하지 않도록 애플리케이션 개발 환경을 제공할 수 있다. 또한, JVM 및 Webkit의 내장으로 별도의 연관된 설치 및 설정이 전혀 필요 없게 된다.
- [0156] 도 35는 본 발명의 실시 예에 따른 컴포넌트 구조도를 나타낸다.
- [0157] 도 35를 참조하면, 컴포넌트는 가상 컴포넌트를 포함할 수 있으며, 하위 단계에 비주얼 컴포넌트 및 논비주얼

컴포넌트를 포함할 수 있다. 그리고, 비주얼 컴포넌트는 라벨, 버튼, 텍스트박스, 텍스트뷰, 콤보박스, 텍스트 영역, 이미지뷰, 차트, 구글맵, 미디어 또는 임베디드 컴포넌트 중 적어도 하나를 포함할 수 있으며, 그 하위 단계에 컨테이너 컴포넌트 및 콜렉션 컴포넌트를 포함할 수 있다.

- [0158] 컨테이너 컴포넌트는 애플리케이션, 폼, 헤더, 풋터, 패널, 레이아웃 패널, 수직 패널, Accordion, 프레임 스플리터, 프레임 컨테이너 및 프레임 중 적어도 하나를 포함할 수 있고, 컬렉션 컴포넌트는 이미지슬라이드 뷰, 커버 플로우, 아이콘리스트 뷰, 썸네일리스트 뷰, 네비게이션 바 리스트 뷰, 리스트 뷰, 멀티 리스트 뷰 및 캘린더 리스트 뷰 컴포넌트 중 적어도 하나를 포함할 수 있다.
- [0159] 한편, 논비주얼 컴포넌트는 로딩 바, 타이머, 날짜시간, DBinsert/Update/Delete, 링크, 웹링크, 메시지박스, 액션리스트, 변수, 로그인 및 doc뷰어 컴포넌트중 적어도 하나를 포함할 수 있으며, 하위레벨에 데이터스토어 컴포넌트를 포함할 수 있다. 데이터스토어 컴포넌트는 DB관련하여, DB테이블, 웹 앱 모듈, Rss, Assets, 파일 리스트, 폼데이터스토어, 업데이트스토어, GPS, 컴퍼스 및 배터리 컴포넌트 중 적어도 하나를 포함할 수 있다.
- [0160] 이하에서는 도 36 내지 도 53을 참조하여 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 방법에 따라 제공되는 화면들을 이용하여 애플리케이션이 생성되는 과정을 설명한다.
- [0161] 도 36는 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공을 위한 프로젝트 생성 화면을 나타낸다. 도 36에서 개발자는 파일 메뉴의 새로 만들기를 클릭함으로써, 제작하고자 하는 프로젝트에 대한 선택을 수행할 수 있다.
- [0162] 이후, 도 37과 같이 개발자가 새로운 프로젝트를 선택하고자 하는 경우 복수의 템플릿을 제공하는 화면이 표시될 수 있다. 특히, 템플릿 외에도 function 및 tablet이 제공될 수 있으며, 개발자가 Default를 선택하는 경우 템플릿 없이 임의로 원하는 프로젝트 제작에 착수할 수 있는 인터페이스를 제공할 수 있다. 그리고, 개발자가 복수의 템플릿 중 특정 템플릿을 선택하는 경우, 기본적인 구성이 미리 설정된 템플릿에 기초하여 새로운 프로젝트를 생성할 수 있다.
- [0163] 그리고, 도 38에서는 프로젝트 생성을 위한 라이선스 유효성 검증 화면을 나타내고 있다. 서버(200) 또는 애플리케이션 개발 환경 제공 장치(100)는 사용자, 비밀번호 및 호스트 URL입력에 기초하여 프로젝트 생성을 위한 라이선스가 있는지를 확인할 수 있다. 그리고, 애플리케이션 개발 환경 제공 장치(100)는 구매 인증되거나 또는 ASP 서비스로 이용함이 확인되는 경우에만 유효하게 다음 단계로 이동하도록 제한할 수 있다.
- [0164] 도 39은 프로젝트 정보 설정 화면을 나타낸다. 프로젝트 정보 설정 화면은 애플리케이션 개발을 위한 프로젝트의 명칭, 어플명, 아이디 및 프로젝트 버전을 설정할 수 있다.
- [0165] 도 40는 본 발명의 실시 예에 따른 제1 화면을 포함하는 어플리케이션 개발 환경을 나타낸다. 특히, 제1 화면은 도 39의 sitemap 에디터 창에 대응될 수 있다. 도 40에서는 앞서 선택된 RSS 템플릿에 의해 미리 생성된 기본 Home 페이지가 생성되어 있으며, 활성화 되어 있음을 나타내고 있다.
- [0166] 도 41은 제1 화면에 새로운 페이지가 추가된 것을 나타낸다. 앞서 설명한 바와 같이, 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 드래그 앤 드랍을 이용하여 페이지를 추가할 수 있다. 예를 들어, 도 41과 같이 sitemap 에디터 창 우측 palette에 포함된 Form 페이지 아이콘을 제1 화면으로 드래그 앤 드랍함으로써 Form 페이지가 추가될 수 있다.
- [0167] 도 42은 Form 페이지의 속성값 변경을 통해 페이지의 ID를 RSSpage로 변경하는 것을 나타낸다. 개발자는 속성 창에 표시된 입력 인터페이스를 통하여 각 페이지별 속성값 변경을 수행할 수 있다.
- [0168] 그리고, 도 43는 Home 페이지와 RSSpage 사이의 연결을 수행하는 것을 나타낸다. 도 43에 도시된 바와 같이, 개발자는 link를 클릭 후, 먼저 연결 시작 페이지인 Home 페이지 아이콘을 클릭하고, 이후 연결 대상 페이지인 RSSpage 페이지 아이콘을 클릭함으로써 페이지간 연결이 수행될 수 있다. 연결이 수행되면 제1 화면에는 각 페이지간의 연결 관계가 앞서 설명한 화살표 이미지로 표시될 수 있다.
- [0169] 도 44은 Home 페이지에 컴포넌트를 추가하기 위한 화면을 나타낸다. 예를 들어, 애플리케이션 개발 환경 제공 장치(100)는 도 43의 에디터 상에서 select 클릭 및 이에 따른 Home 페이지 더블 클릭에 따라, 컴포넌트 추가 화면을 표시할 수 있다. 도 44과 같이, 추가 가능한 컴포넌트의 종류는 예를들어 Basic, Time, ListView, Layout, Database, DataStore, Forms, Extention, Platform, Tablet, Charts, Rules 및 Unofficials 중 적어도 하나를 포함할 수 있다.

- [0170] 한편, 도 45는 Home 페이지에 RSS\_CNN 버튼 컴포넌트가 생성된 것을 나타낸다. 예를 들어, Basic 종류의 컴포넌트 중 Button 컴포넌트가 드래그 앤 드랍되어 Home 페이지에 추가될 수 있으며, 속성 창에서 ID값이 RSS\_CNN값으로 변경됨으로써 도 45와 같은 RSS\_CNN 버튼 컴포넌트가 생성될 수 있다. 또한, 속성의 Style 탭에서 size값 변경을 수행하여 RSS\_CNN 버튼 컴포넌트를 적당한 크기로 변경할 수 있다.
- [0171] 그리고, 도 46는 Home 페이지에 포함된 각 컴포넌트간 연결 관계를 나타내는 제2 화면을 표시하고, RSS\_CNN 버튼 컴포넌트를 link\_RSSpage 컴포넌트와 연결시키는 것을 나타낸다. 제2 화면으로의 전환은 하단의 Workflow 탭 선택으로 이루어질 수 있다. 그리고, 우측 flow창의 아이콘들을 드래그 앤 드랍함으로써 RSS\_CNN 버튼의 onClick 이벤트와 link\_RSSpage의 move 액션을 연결시킬 수 있다. 이에 따라 컴포넌트간 연결 관계가 화살표 이미지로 표시될 수 있으며, 이와 같은 링크 연결을 통해 Home 페이지에서의 RSS\_CNN버튼이 클릭되면 RSSpage 페이지로 이동되도록 설정될 수 있다.
- [0172] 도 47은 RSSpage 페이지의 컴포넌트 추가 화면을 나타낸다. 도 47과 같이, RSS page 페이지로 ThumbListView를 드래그 앤 드랍 하여 RSSpage에 이동하면, 미리 설정된 썸네일 리스트 뷰 컴포넌트가 RSS page의 페이지 화면에 표시될 수 있다.
- [0173] 도 48은 RSSPage 페이지에 대한 컴포넌트간 연결 관계 화면을 나타낸다.
- [0174] 도 48에 도시된 바와 같이, RSSpage 가 초기화 될 때에 발생하는 onInit이벤트와 RSS Data를 가져오는 Bind 액션이 연결될 수 있다. 이는 앞서 설명한 바와 마찬가지로 flow를 클릭하여 RSS page form 컴포넌트의 onInit 이벤트와 rss1 컴포넌트의 bind를 연결 함으로서 즉시 수행될 수 있다.
- [0175] 도 49은 RSSpage 페이지의 RSS컴포넌트에 대한 데이터 바인딩을 위한 화면을 나타낸다.
- [0176] 도 49과 같이, 애플리케이션 개발 환경 제공 장치(100)는 RSS 데이터 값들을 페이지 상에서 보여주는 화면을 표시할 수 있다. 또한, 애플리케이션 개발 환경 제공 장치(100)는 RSS data 값들이 ThumbListView 컴포넌트의 어느 위치에 대응될 것인지를 결정할 수 있으며, rss1에 자동으로 형성된 값들을 ThumbListView의 대응 리스트 값에 할당할 수 있다.
- [0177] 그리고, 도 50는 완성된 애플리케이션을 다양한 플랫폼에 맞게 변환하기 위한 내보내기 화면을 나타낸다. 앞서 설명한 바와 같이, 완성된 애플리케이션은 중간 언어로 구성될 수 있으며, 타겟 OS에 따라 제2 언어로 변환될 수 있다. 따라서, 내보내기 화면은 이와 같은 제2 언어를 선택할 수 있도록 하는 선택 화면을 제공할 수 있다. 예를 들어, 내보내기 선택 대상은 안드로이드 프로젝트, Xcode 프로젝트, 윈도우폰, 타이젠 중 어느 하나일 수 있다.
- [0178] 도 51은 안드로이드 프로젝트를 선택한 경우 패키지 생성화면을 나타낸다. 이 경우, 애플리케이션 개발 환경 제공 장치(100)는 안드로이드 마켓 어플리케이션(apk) 자체를 생성하거나 안드로이드 개발을 위한 프로젝트 자체를 생성할 수 있는 선택 메뉴를 표시할 수 있다. 프로젝트 자체를 생성하는 경우에는 안드로이드 앱 개발을 위한 다른 틀에서도 애플리케이션 수정이 가능하게 되는 장점이 있다.
- [0179] 도 52는 해당 어플리케이션에서 실제로 동작하는지를 확인하기 위한 시뮬레이터 동작 화면을 나타내며, 도 53은 최종 애플리케이션이 변환 출력되고, 스마트 디바이스로 전송되어 동작하는 스마트 디바이스의 화면을 나타낸다.
- [0180] 도 53에 도시된 바와 같이, 스마트 디바이스에서는 앞서 생성한 RSS\_CNN 버튼을 home 페이지에서 클릭하면 Form1 페이지(RSSpage)로 이동되며, Form1 페이지 기동에 따라 데이터 바인딩된 웹상의 DB로부터 RSS데이터를 수신하여 썸네일 리스트 뷰 컴포넌트에 표시할 수 있는 애플리케이션이 동작하게 된다.
- [0181] 이와 같이 본 발명의 실시 예에 따른 애플리케이션 개발 환경 제공 장치(100)는 애플리케이션 개발에 있어서, 각종 다양한 코드 언어에 대한 전문적인 지식 없이도, 단순 클릭 및 드래그 앤 드랍만으로 다양한 플랫폼에 대한 용이한 어플리케이션 개발이 가능하게 된다.
- [0182] 상술한 본 발명에 따른 애플리케이션 개발 환경 제공 방법은 컴퓨터에서 실행되기 위한 프로그램으로 제작되어 컴퓨터가 읽을 수 있는 기록 매체에 저장될 수 있으며, 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다.
- [0183] 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고, 상기 방법을 구현하기 위한 기능적인(function) 프로그램

램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있다.

[0184]

또한, 이상에서는 본 발명의 바람직한 실시예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특정의 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 다양한 변형 실시가 가능한 것은 물론이고, 이러한 변형 실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해 되어서는 안될 것이다.

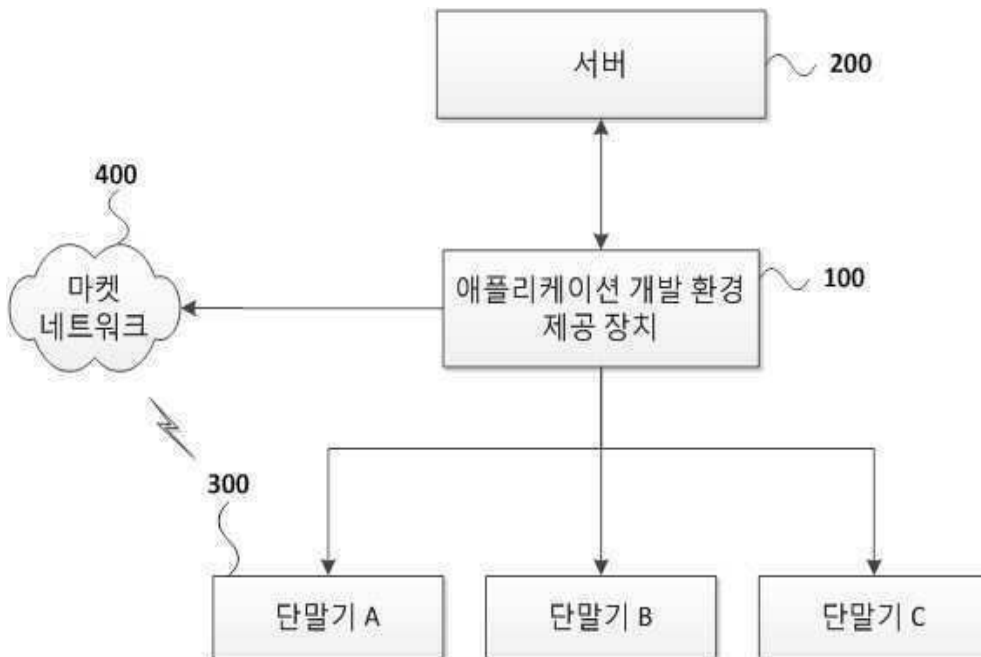
도면

도면1

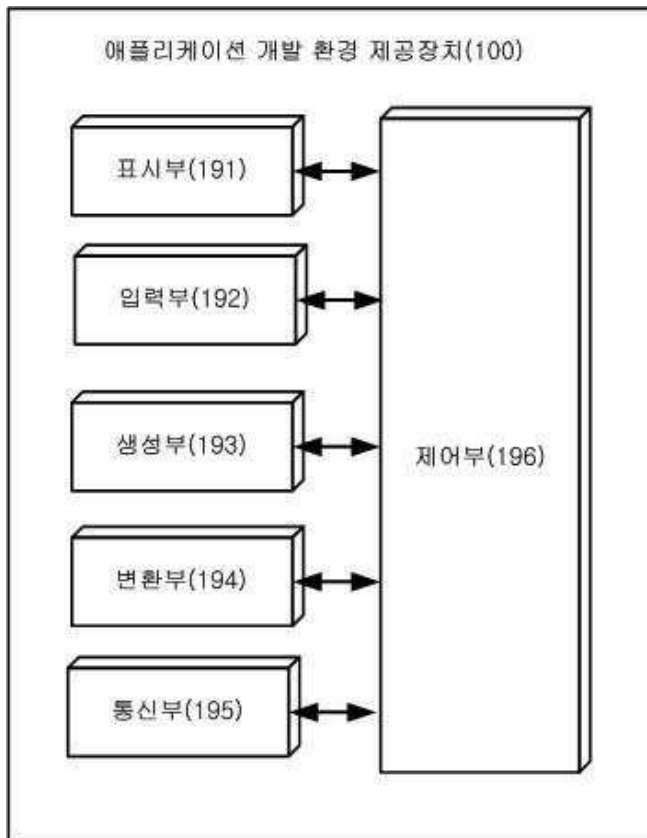
```

ExamActivity.java
1 package com.tempuri;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7
8 public class ExamActivity extends Activity {
9     @Override
10    public void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.main);
13        final Button requestInfo = (Button) findViewById(R.id.requestInfo);
14        requestInfo.setOnClickListener(new Button.OnClickListener() {
15            public void onClick(View view) {
16
17            }
18        });
19    }
20
21 }
22
    
```

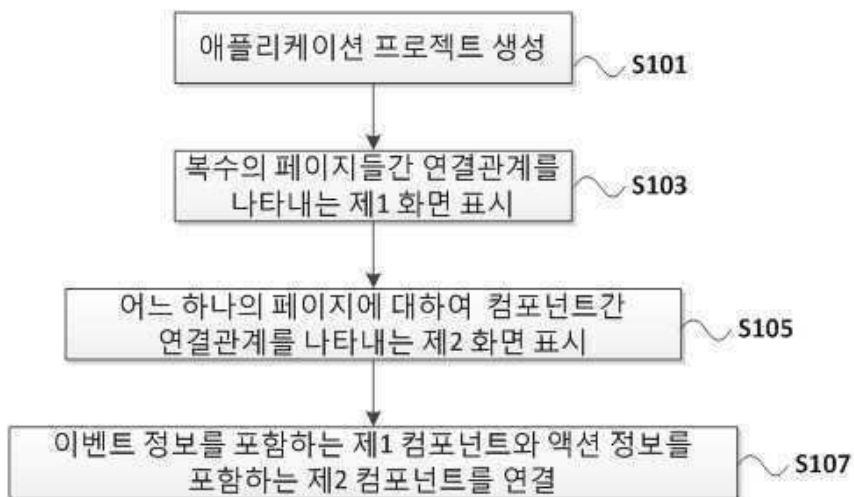
도면2



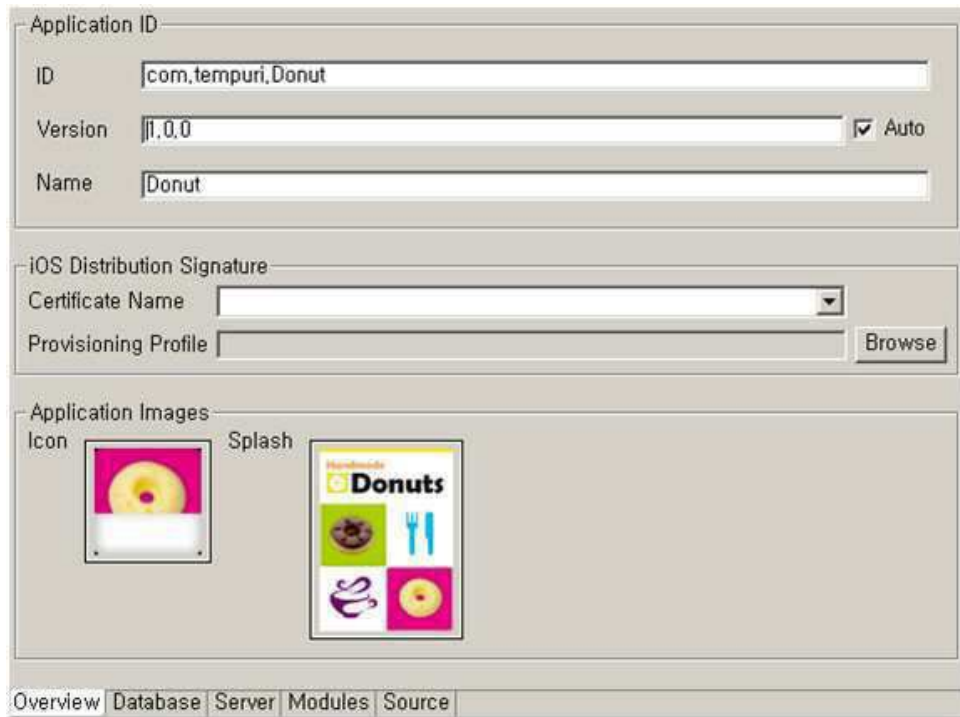
도면3



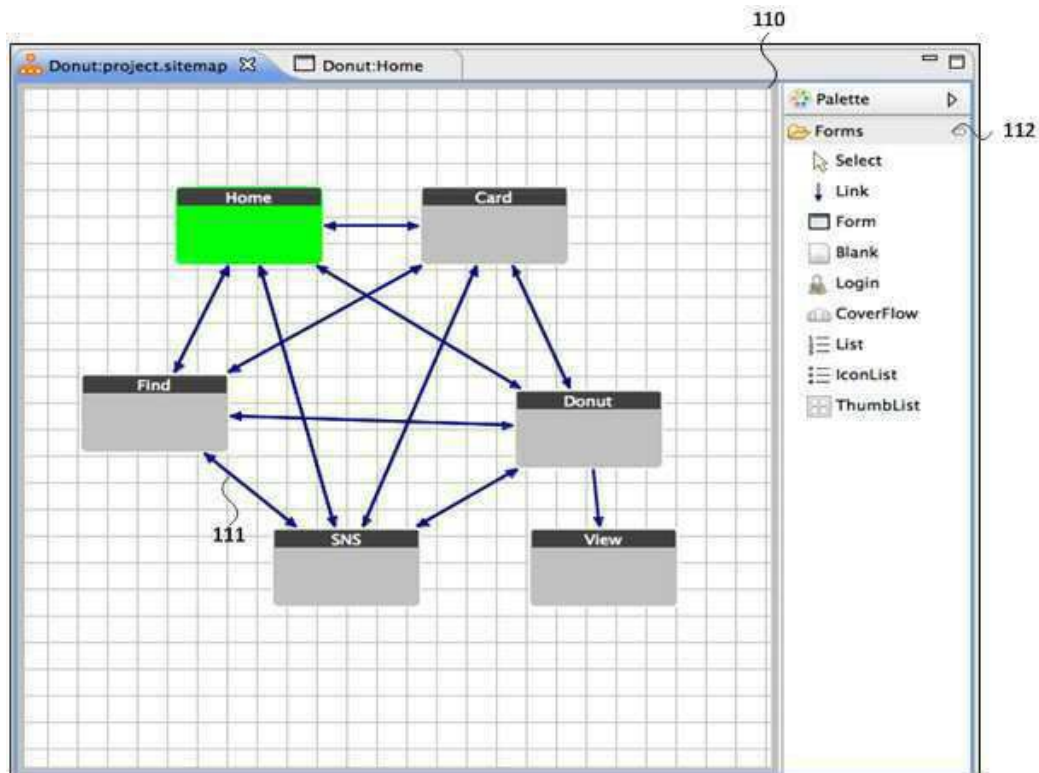
도면4



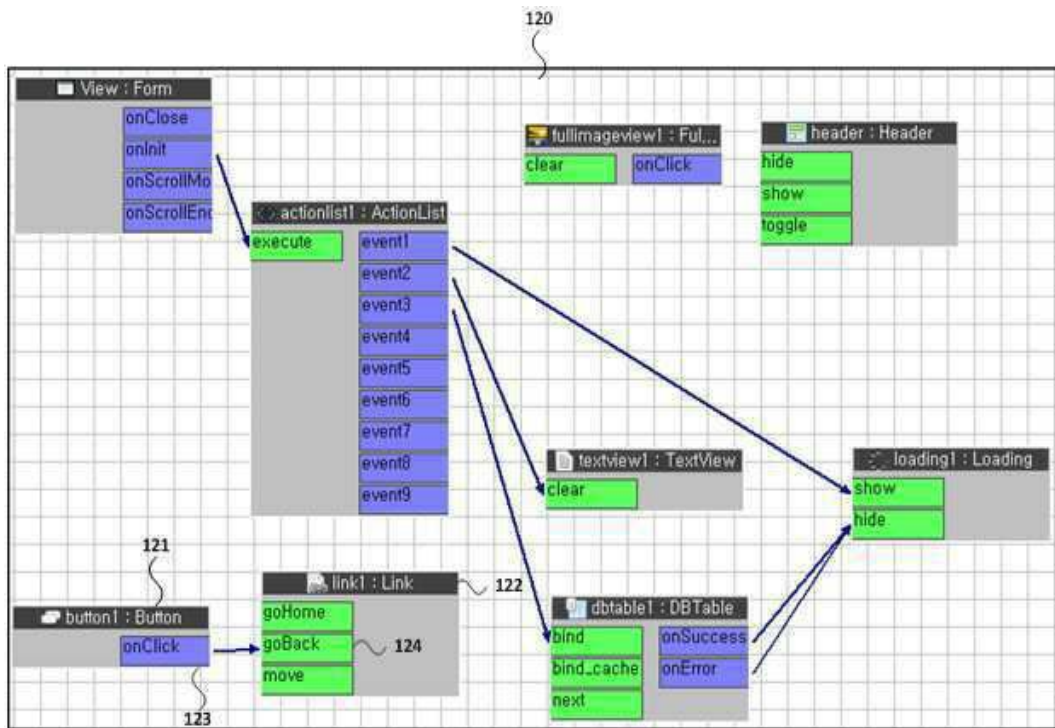
도면5



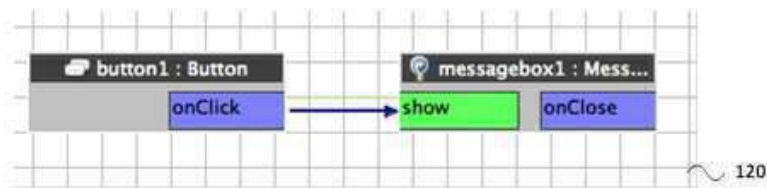
도면6



도면7



도면8



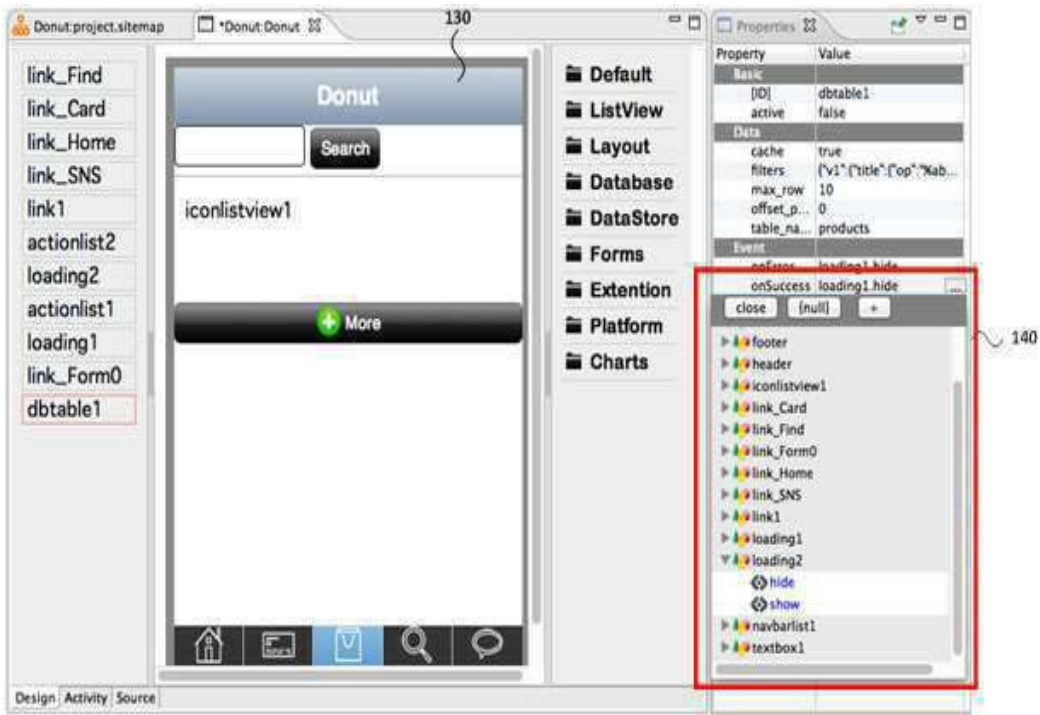
```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <x:Form id="Home" xmlns:x1="http://forms.component.nest.nexdigm.co.kr" xmlns:x="ht
3   <mainForm>true</mainForm>
4   <x:Header id="header">
5     <title>Home</title>
6   </x:Header>
7   <x:Footer id="footer"/>
8   <x:Button id="button1">
9     <onClick>
10      <target ref="messagebox1" method="show"/>
11    </onClick>
12  </x:Button>
13  <x1:MessageBox id="messagebox1"/>
14 </x:Form>
15 |
    
```

도면9



도면10



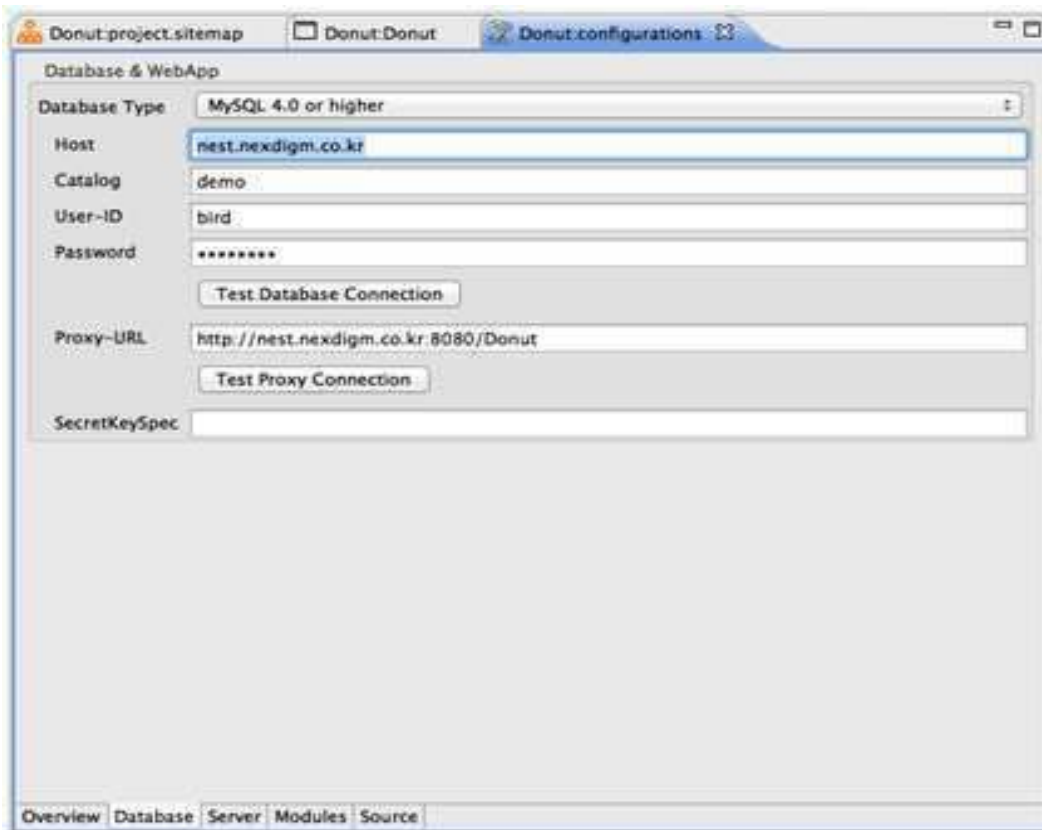
도면11



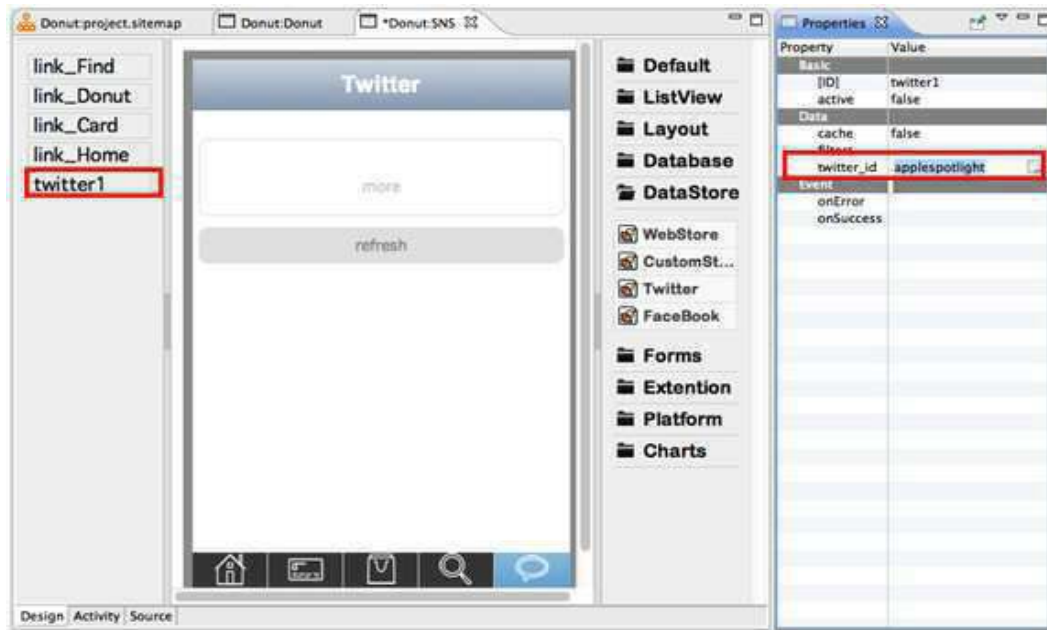
도면12



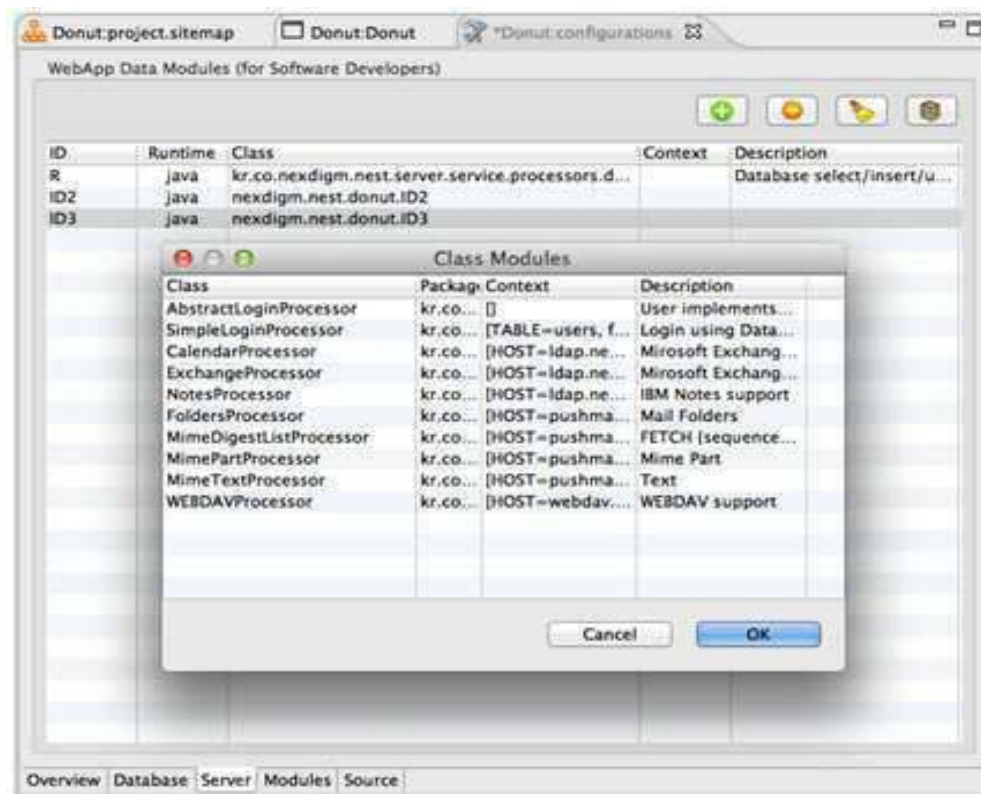
도면13



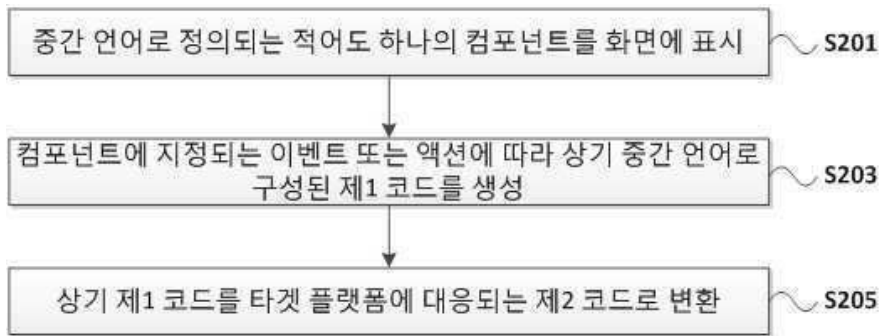
도면14



도면15



도면16



도면17

<pre> 28 29 final Composite panel = new Composite(container, SWT.NONE); 30 panel.setBounds(0, 0, 500, 100); 31 32 Button button = new Button(container, SWT.NONE); 33 button.setBounds(0, 120, 30, 20); 34 35 button.addSelectionListener(new SelectionListener() { 36     @Override 37     public void widgetSelected(SelectionEvent event) { 38         panel.setVisible(false); 39 40         MessageBox msgBox = new MessageBox(getShell()); 41         msgBox.setMessage("Panel is disappeared"); 42         msgBox.setText("Alert"); 43         if (IDialogConstants.OK_ID == msgBox.open()) { 44             // TODO: Next implement stub 45         } 46     } 47 } 48           </pre> <p style="text-align: center;">&lt;일반적인 JAVA 코드&gt;</p>	<pre> 3 &lt;title&gt;name&lt;/title&gt; 4 &lt;/x:Header&gt; 5 &lt;x:Footer id="footer"/&gt; 6 &lt;x:Button id="button1"&gt; 7     &lt;onClick&gt; 8         &lt;target ref="actionlist1" method="execute"/&gt; 9     &lt;/onClick&gt; 10 &lt;/x:Button&gt; 11 &lt;x1:Panel id="panel1"/&gt; 12 &lt;x2:ActionList id="actionlist1"&gt; 13     &lt;event1&gt; 14         &lt;target ref="panel1" method="hide"/&gt; 15     &lt;/event1&gt; 16     &lt;event2&gt; 17         &lt;target ref="messagebox1" method="show"/&gt; 18     &lt;/event2&gt; 19 &lt;/x2:ActionList&gt; 20 &lt;x2:MessageBox id="messagebox1"/&gt; 21 &lt;/x:Form&gt; 22 23 24           </pre> <p style="text-align: center;">&lt;본 발명&gt;</p>
---	--

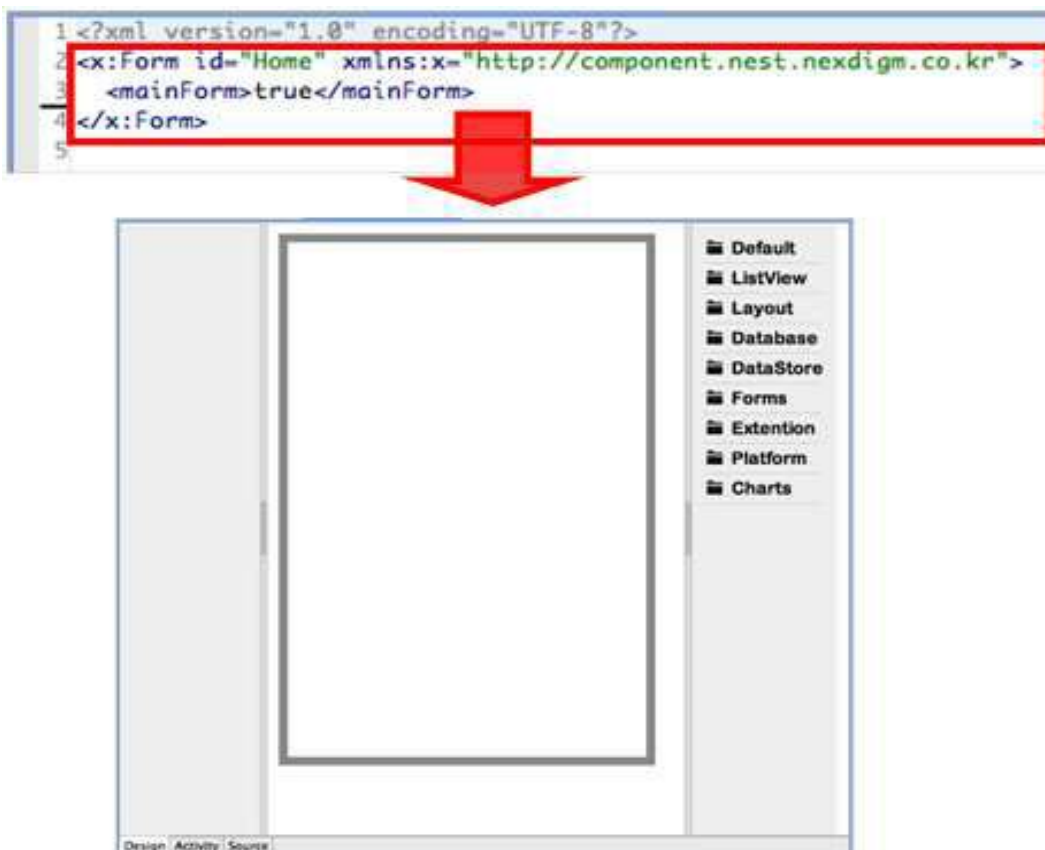
도면18

```

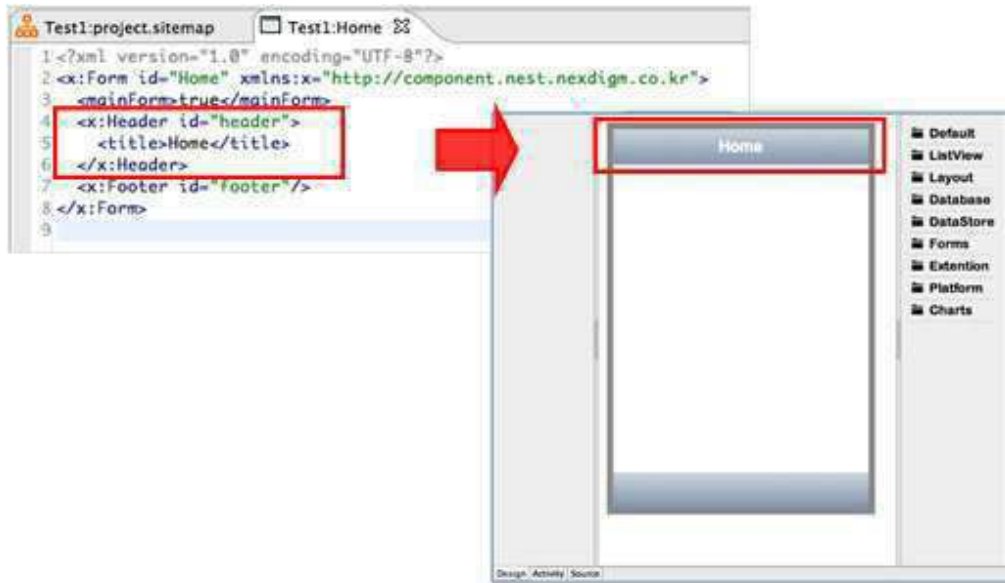
<?xml version="1.0" encoding="UTF-8"?>
<x:Form id="View" xmlns:x1="http://forms.component.nest.nexdigm.co.kr"
xmlns:x2="http://database.component.nest.nexdigm.co.kr" xmlns:x="http://component.nest.nexdigm.co.kr">
  <onInit>
    <target ref="actionlist1" method="execute"/>
  </onInit>
  <x1:ActionList id="actionlist1">
    <event1>
      <target ref="loading1" method="show"/>
    </event1>
    <event2>
      <target ref="dbtable1" method="bind"/>
    </event2>
  </x1:ActionList>
  <x:Loading id="loading1"/>
  <x2:DBTable id="dbtable1">
    <table_name>products</table_name>
    <onSuccess>
      <target ref="loading1" method="hide"/>
    </onSuccess>
  </x2:DBTable>
</x:Form>

```

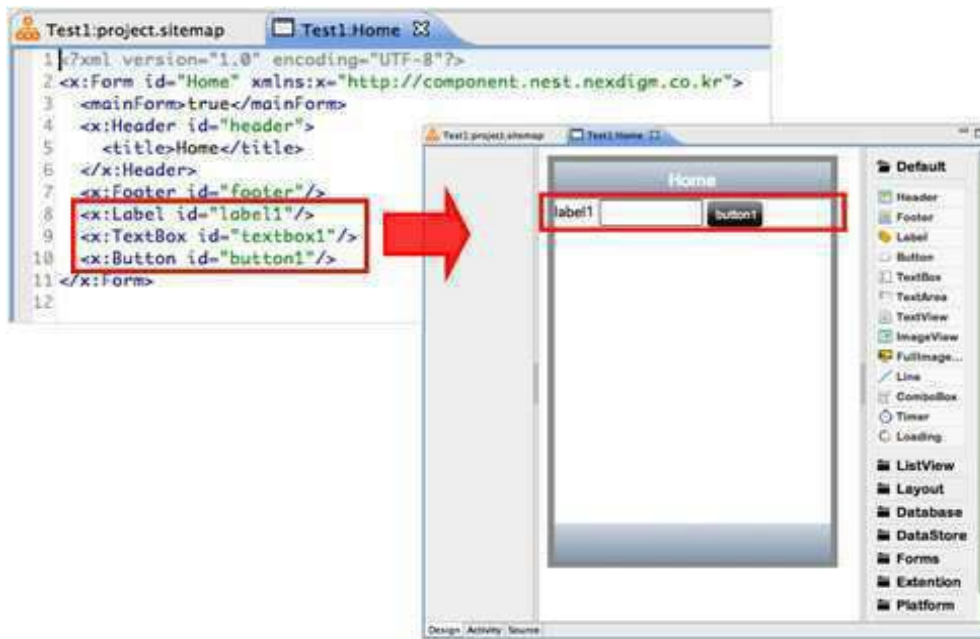
도면19



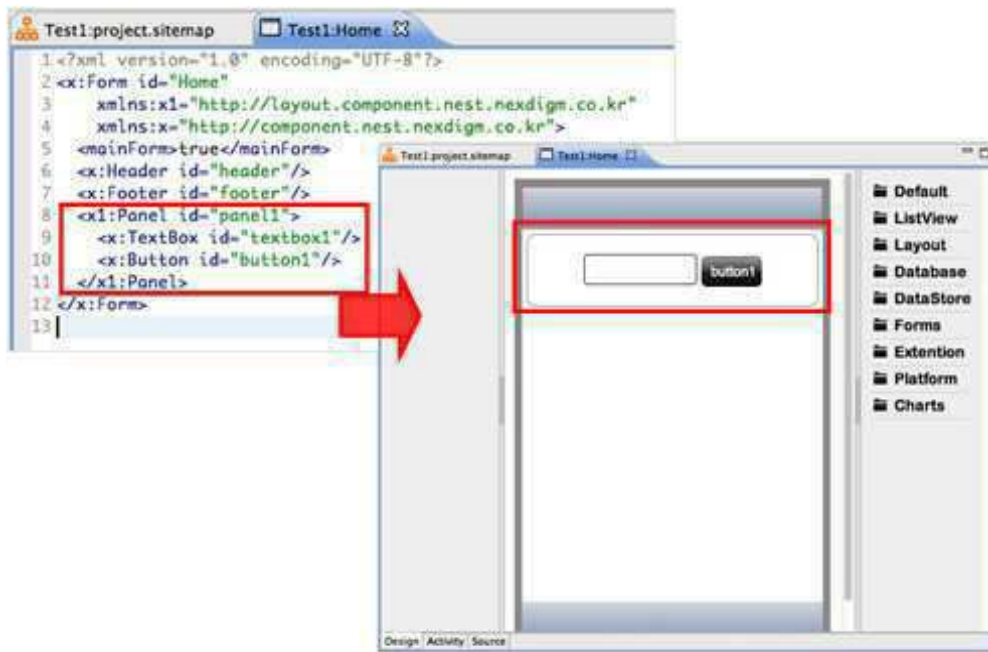
도면20



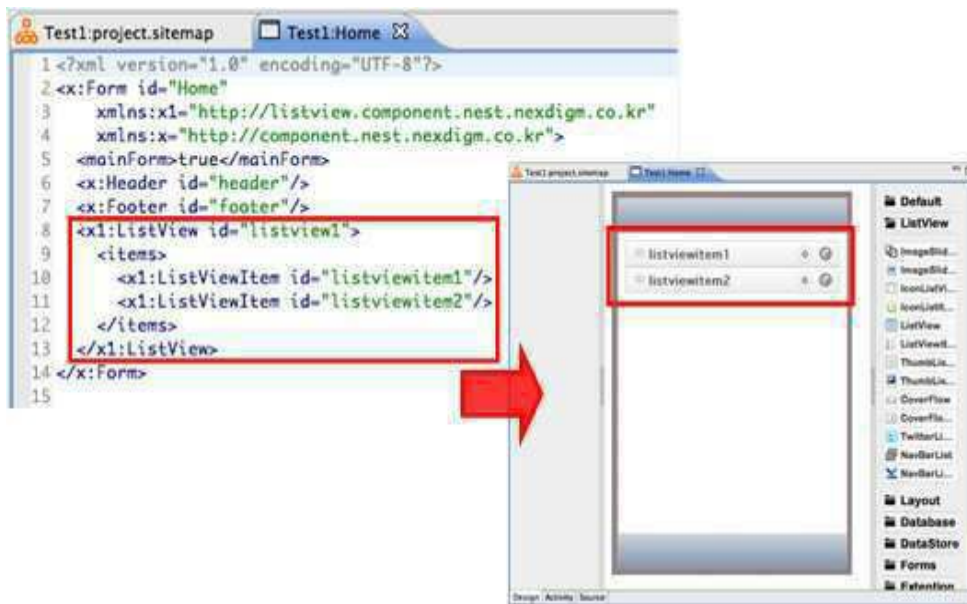
도면21



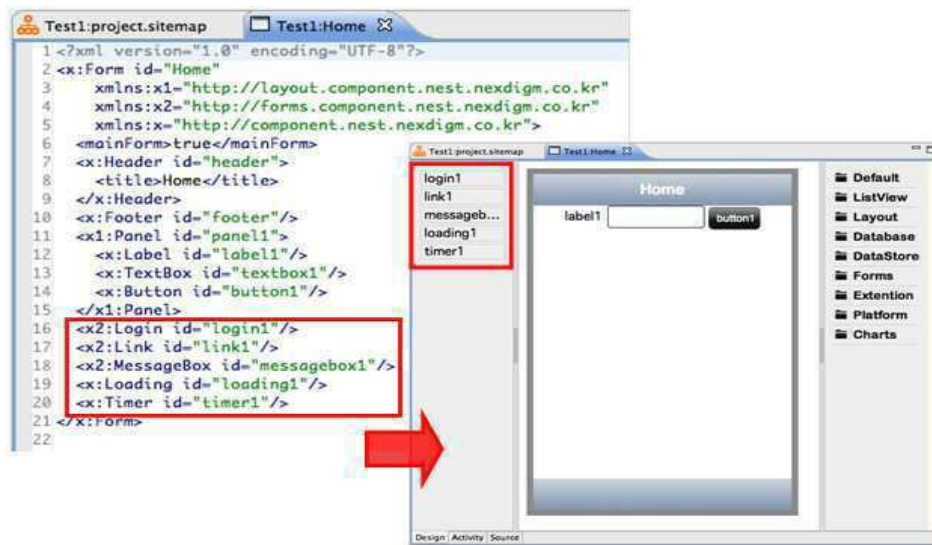
도면22



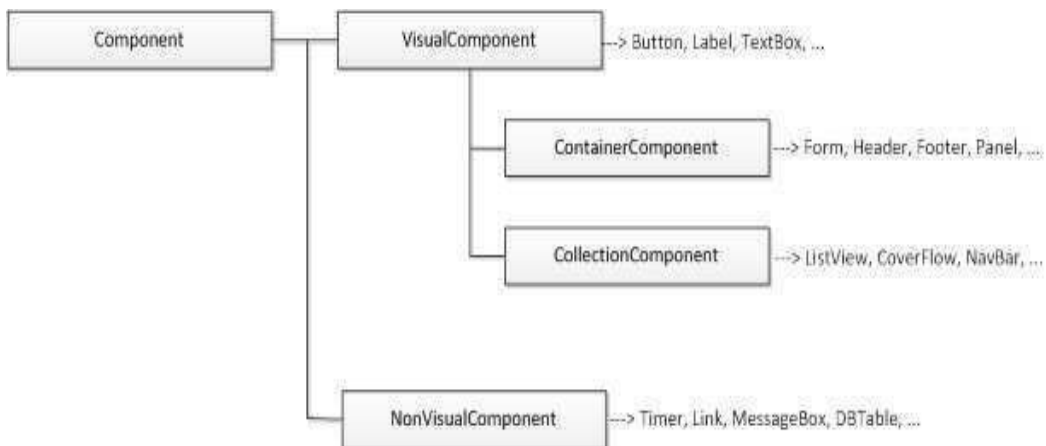
도면23



도면24



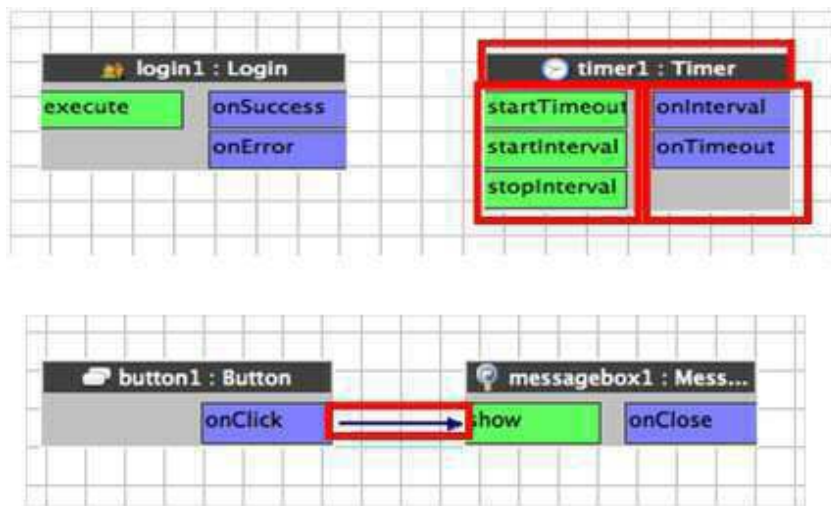
도면25



도면26



도면27



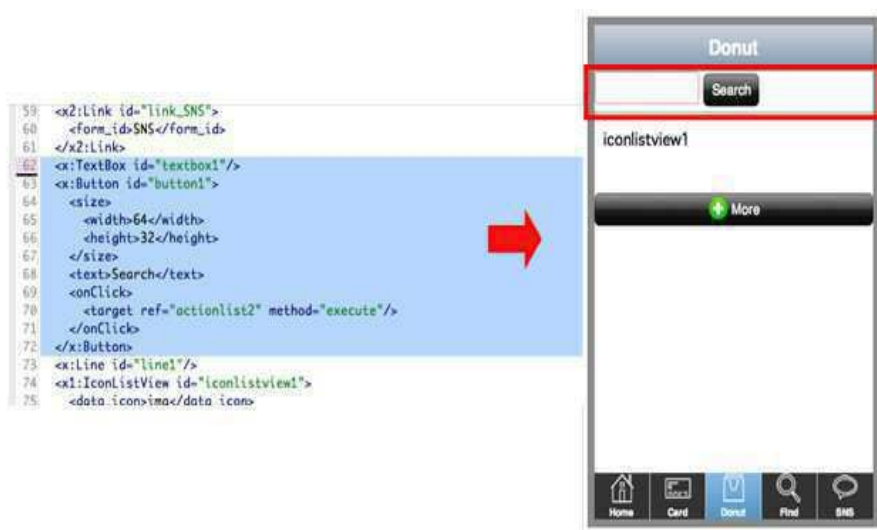
도면28



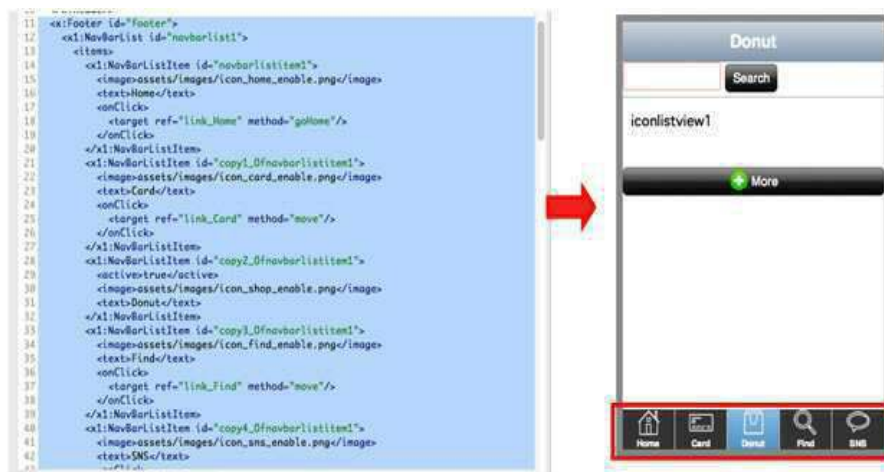
도면29



도면30



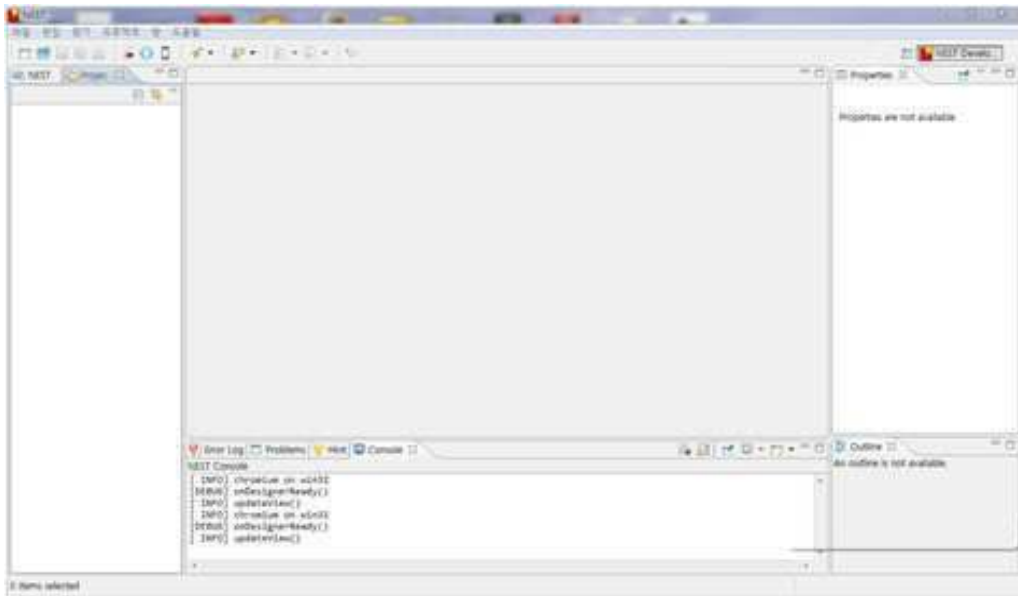
도면31







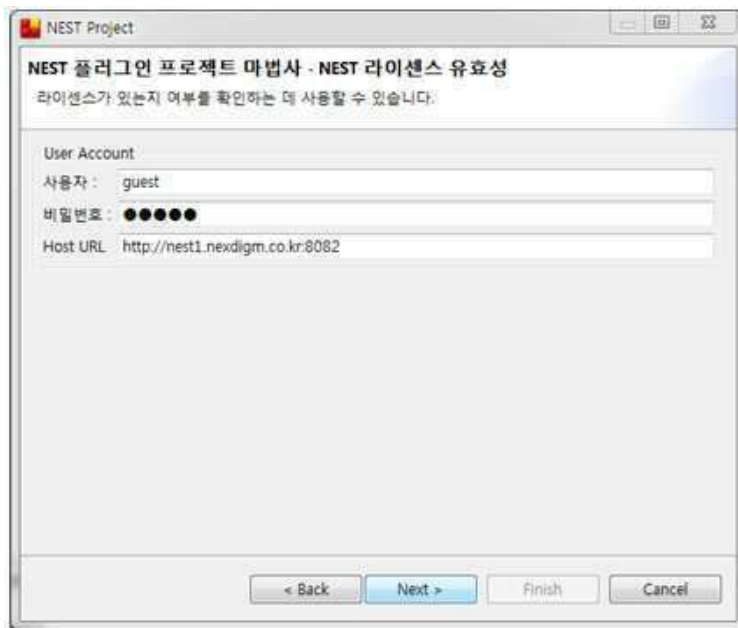
도면36



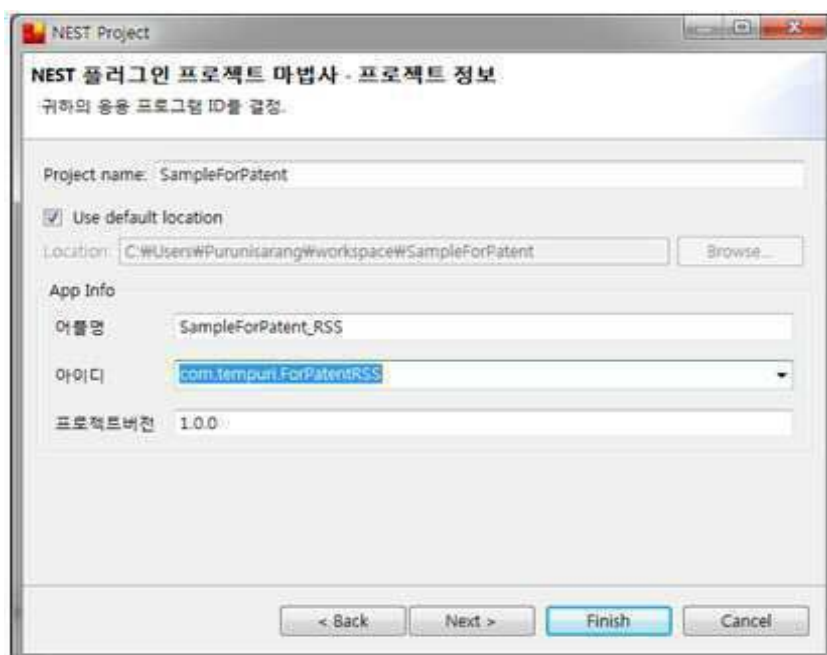
도면37



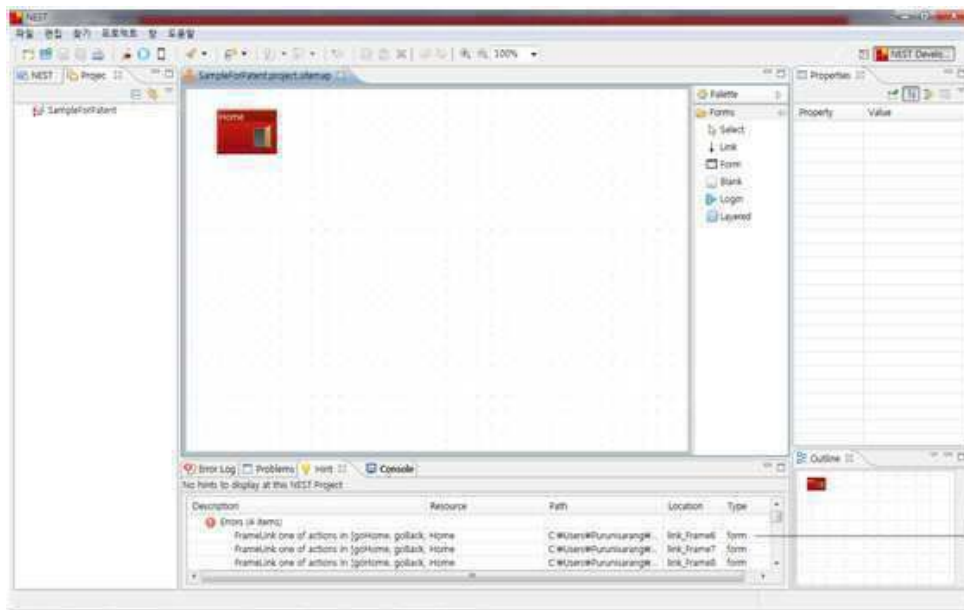
도면38



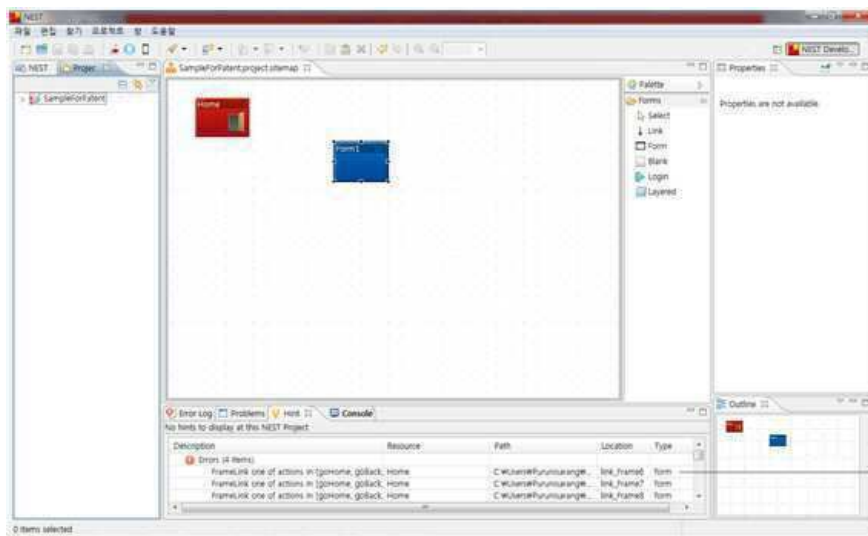
도면39



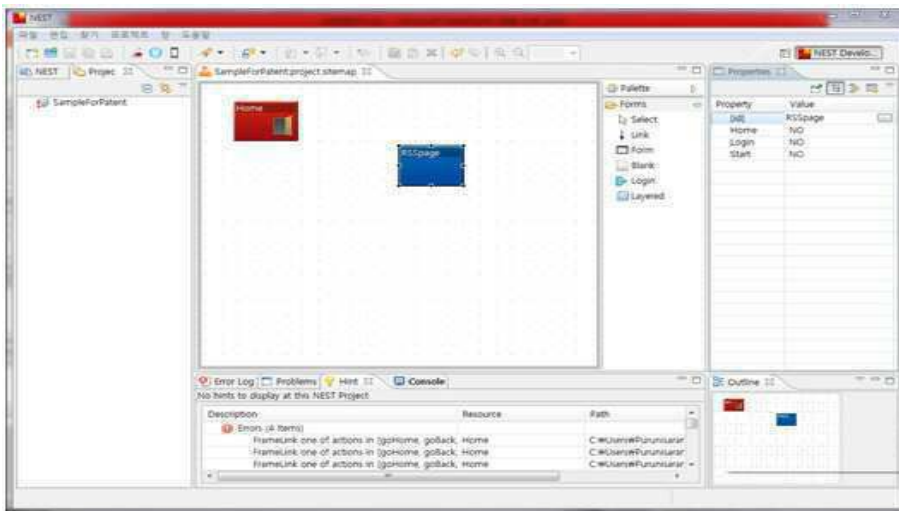
도면40



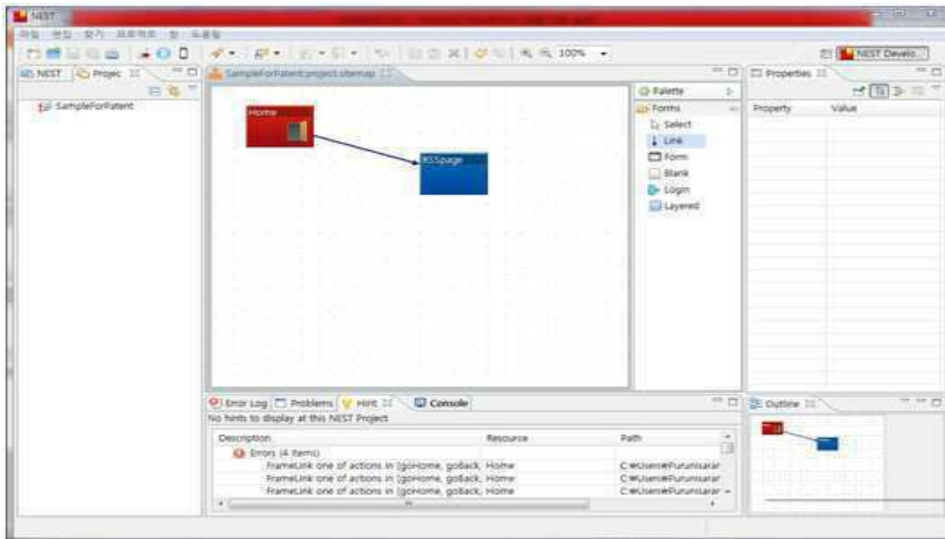
도면41



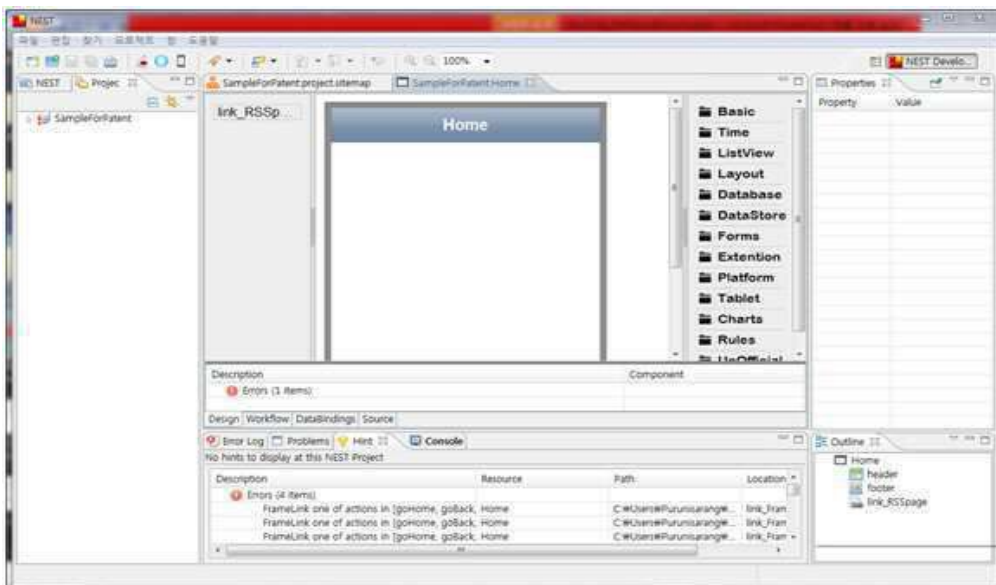
도면42



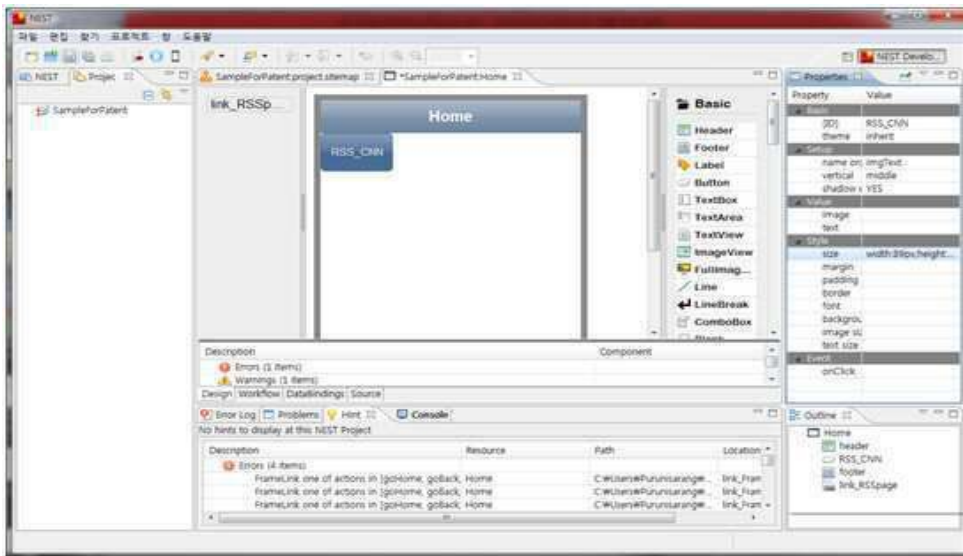
도면43



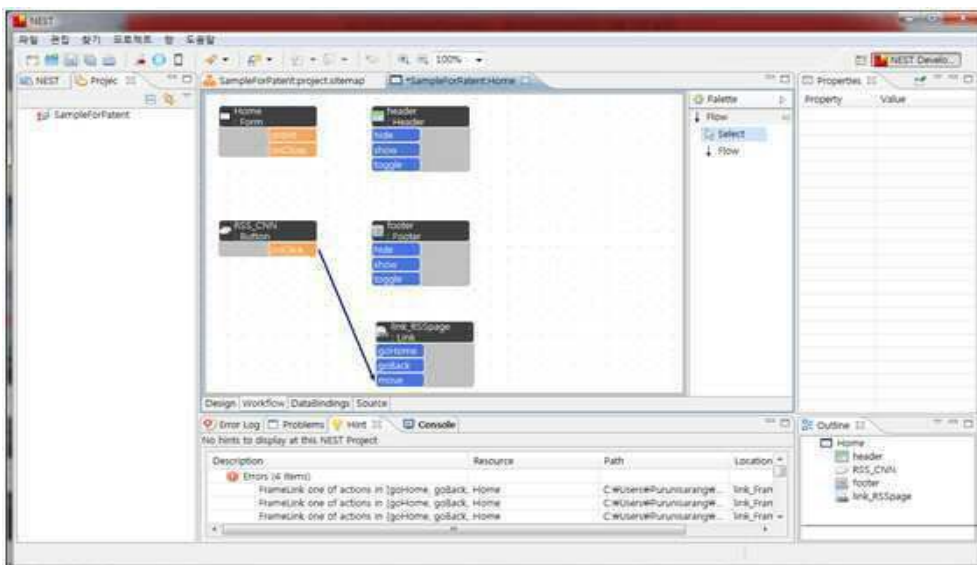
도면44



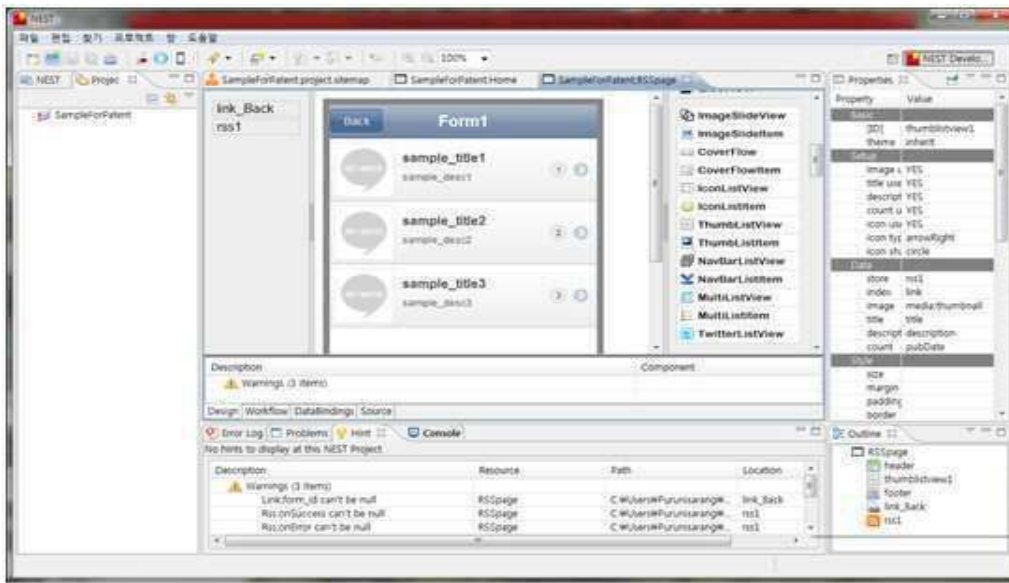
도면45



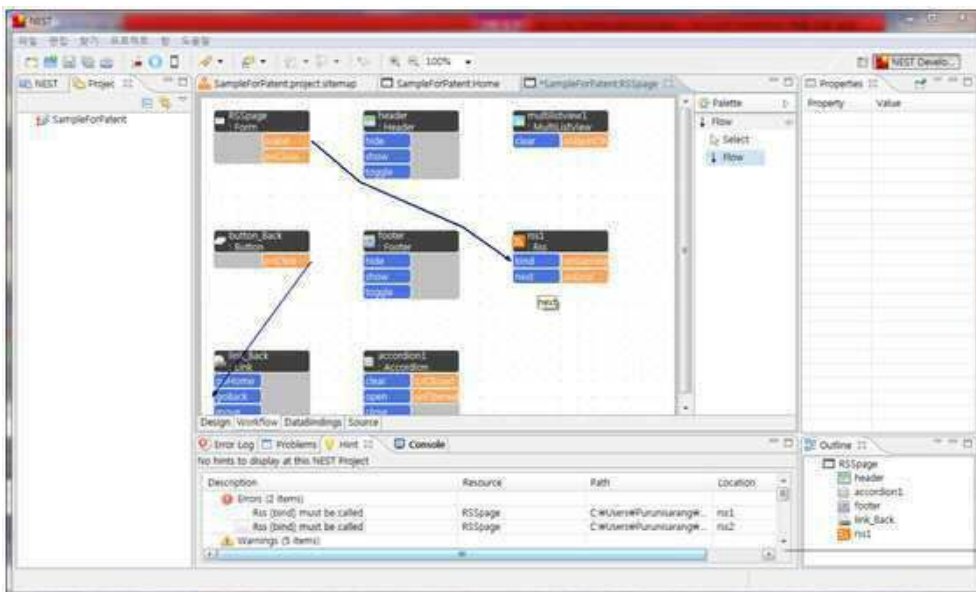
도면46



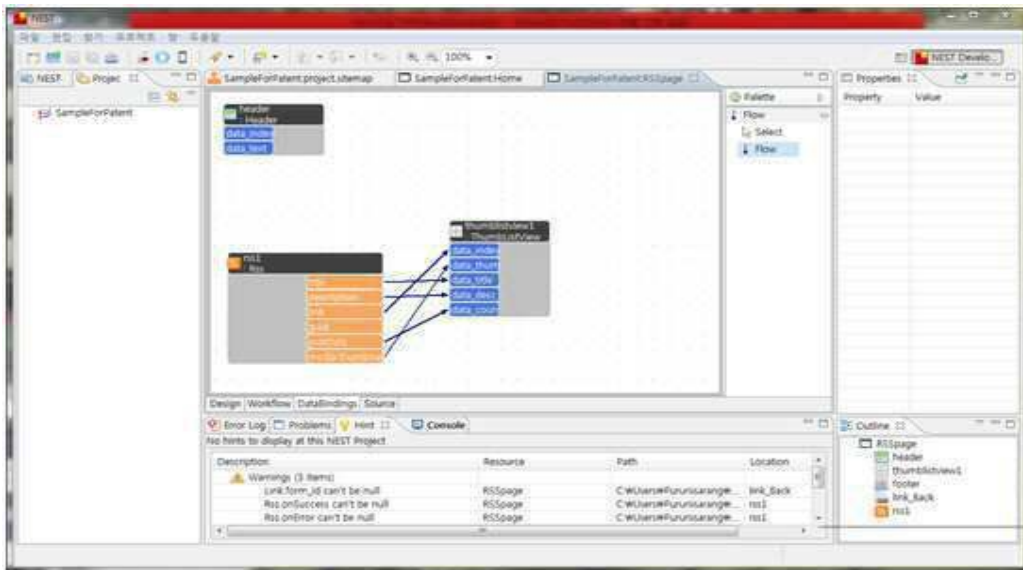
도면47



도면48



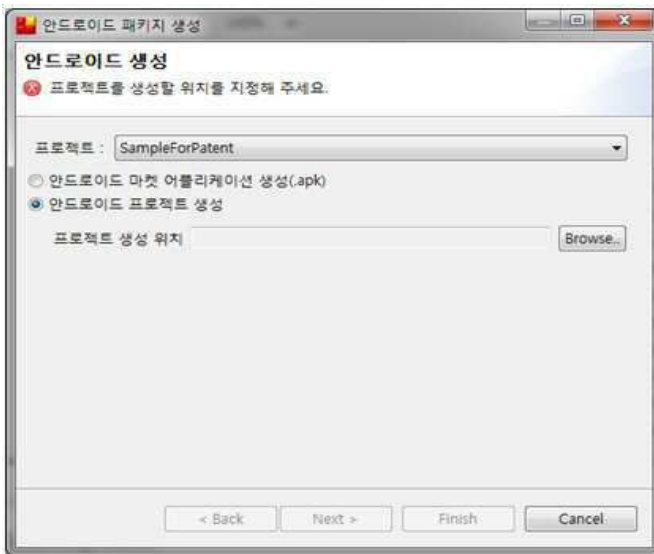
도면49



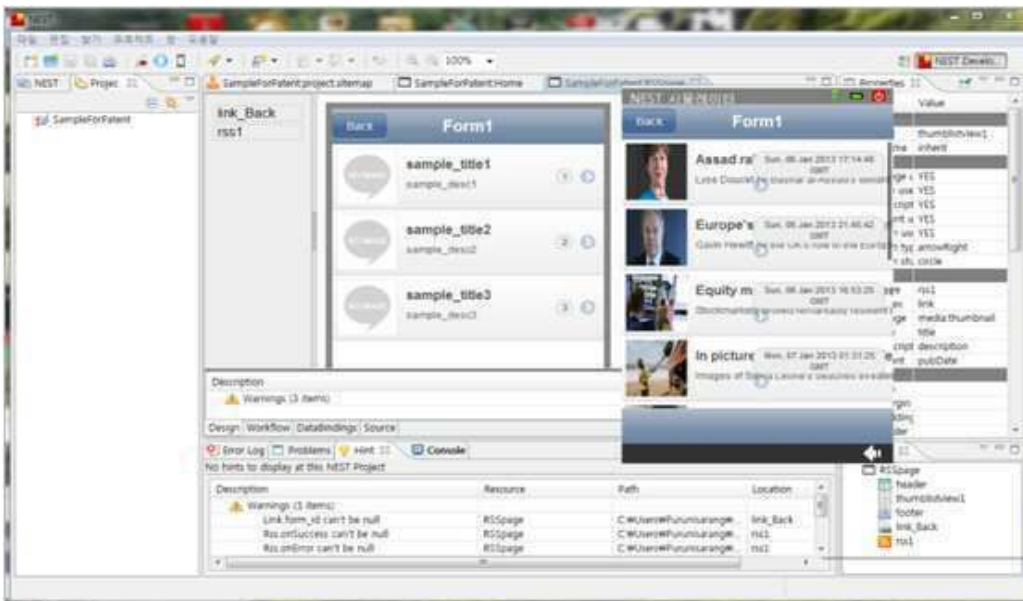
도면50



도면51



도면52



도면53

