

# CH 16

---

## 실전 프로젝트 : 맛집 앱 만들기

## 1. 맛집 관리 하이브리드 앱 개발 요약

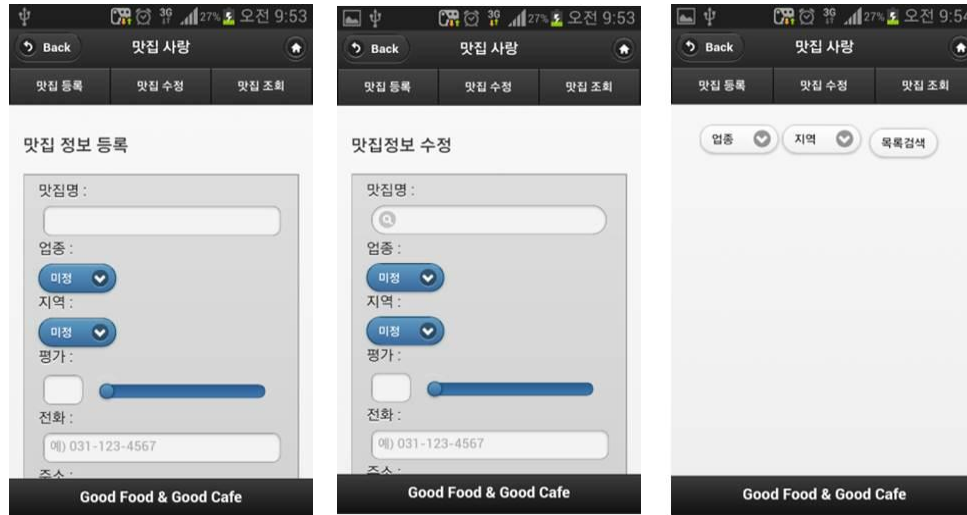
---

>> 학습목표 <<

- ❖파일 시스템 관련 폰갭 API를 살펴본다.
- ❖제이쿼리 모바일과 폰갭을 사용한 맛집 관리 하이브리드앱을 개발해본다.

# 1.1 맛집 앱 : matzipApp

## ● [그림 16-1] 맛집 앱 기본 화면 구성

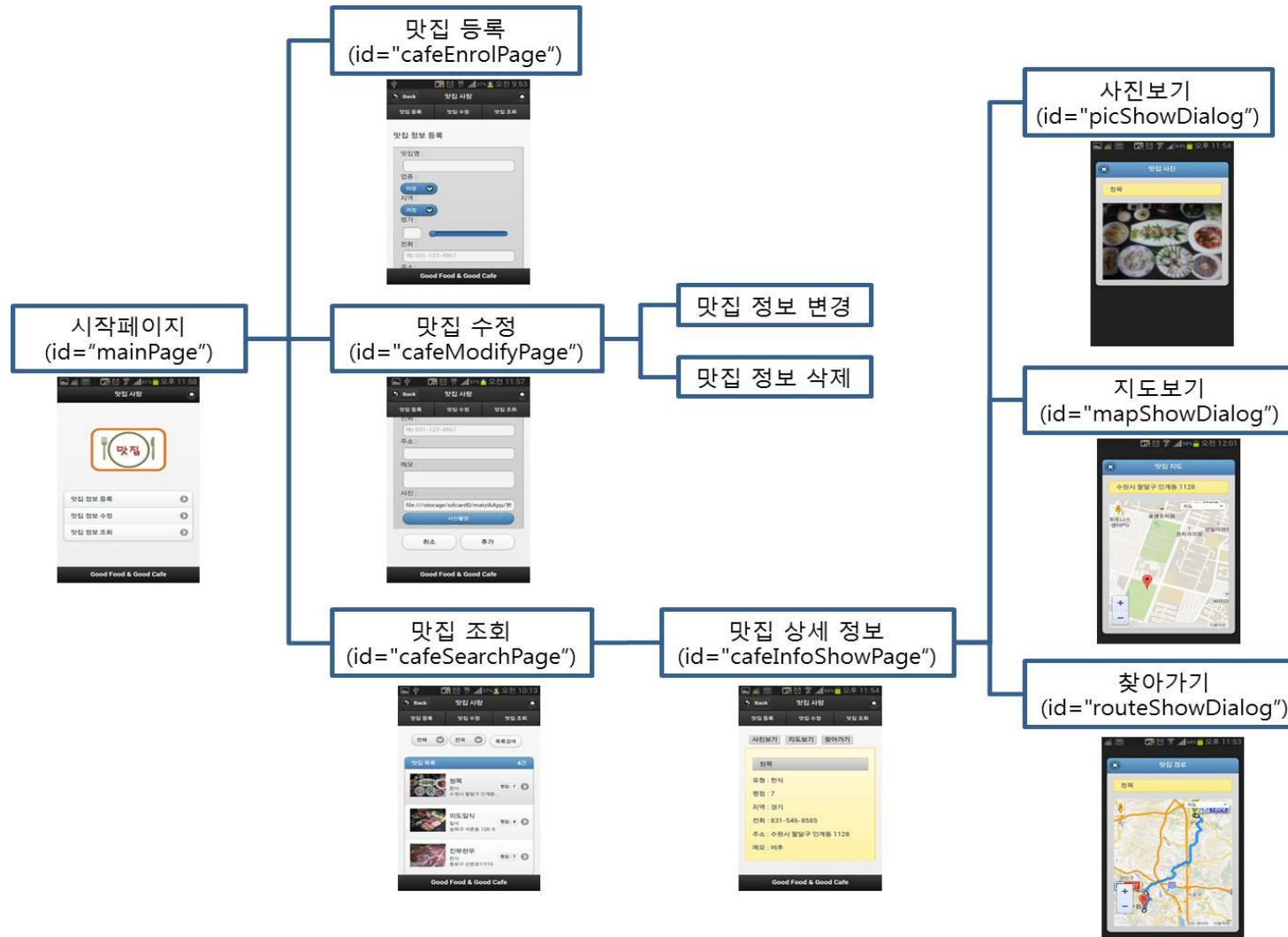


## ● 맛집 앱(matzipApp)의 기능

- 맛집 등록 : 새로운 맛집 정보를 입력하여 모바일 장치 안의 데이터베이스에 저장  
입력하는 맛집 정보 중에는 맛집 사진도 직접 촬영하여 등록 가능
- 맛집 수정 : 맛집 이름으로 맛집 정보 검색 후, 원하는 맛집 정보를 수정하거나 삭제  
삭제된 맛집 정보는 데이터베이스에서 제거되므로 되돌릴 수 없음
- 맛집 조회 : 업종과 지역을 선택함으로써 원하는 맛집만을 선별하여 맛집 목록을 확인  
맛집 목록 중에서 특정 맛집을 선택, 상세한 맛집 정보(맛집 사진, 맛집 위치 지도, 현 위치로부터의 경로 지도 등)를 조회

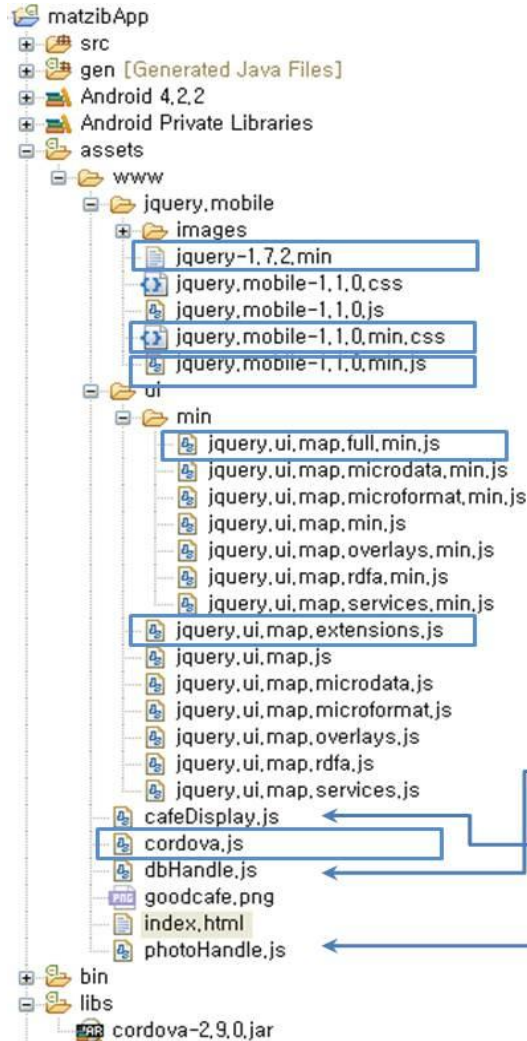
# 맛집 관리 앱의 화면 구성

## ● [그림 16-2] 전체 페이지와 메뉴 구성



# 맛집 앱 프로젝트 구성

## • [그림 16-3] matzipApp 프로젝트의 라이브러리 구성



<!-- 제이쿼리모바일, 제이쿼리, 폰갭 라이브러리 파일(다운로드 방식) -->

<link rel="stylesheet" href="jquery.mobile/jquery.mobile-1.1.0.min.css"/>

<script src="jquery.mobile/jquery-1.7.2.min"> </script>

<script src="jquery.mobile/jquery.mobile-1.1.0.min.js"> </script>

<script src="cordova.js"> </script>

<!-- 구글맵 API 라이브러리 파일(CDN 방식) -->

<script src="http://maps.google.com/maps/api/js?sensor=true&language=ko"> </script>

<!-- jquery-ui-map 플러그인 라이브러리 파일(다운로드 방식) -->

<script src="ui/min/jquery.ui.map.full.min.js"> </script>

<script src="ui/jquery.ui.map.extensions.js"> </script>

<!-- 사용자 정의 자바스크립트 파일 -->

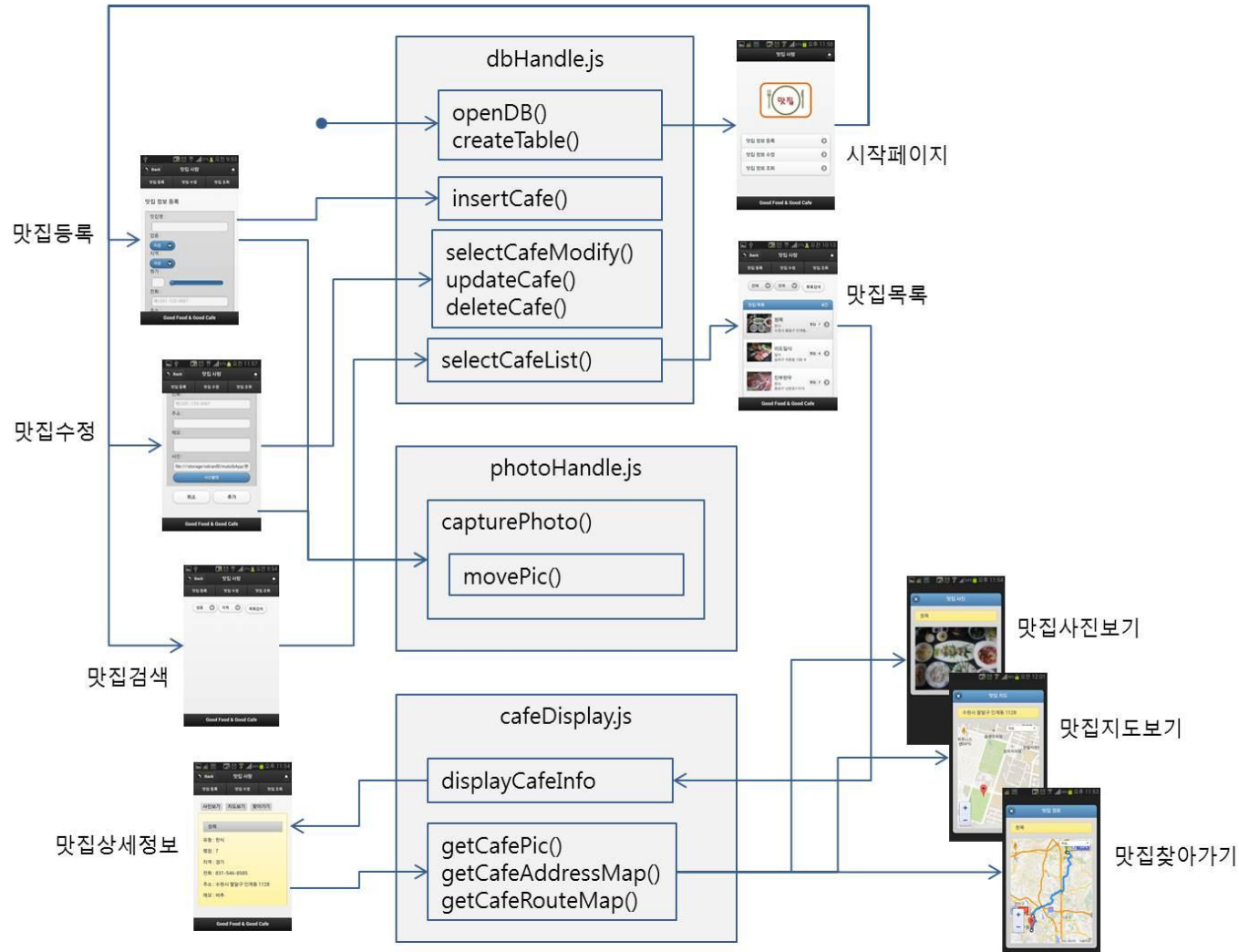
<script src="dbHandle.js"> </script>

<script src="photoHandle.js"> </script>

<script src="cafeDisplay.js"> </script>

# 맛집 앱의 함수 구성

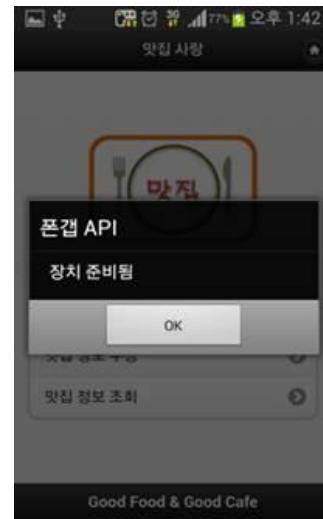
## ● [그림 16-4] matzipApp 앱 함수간의 연관 관계



## 1.2 맛집 앱 시작하기

### ● [표 16-1] cafe 테이블 구조(스키마)

컬럼명	컬럼 유형	의미
id	integer(정수형), 기본 키, 자동 증가	테이블 각 튜플(행)을 식별하기 위한 식별자
name	varchar(20)(문자열)	맛집 이름(최대 20 문자 크기)
type	varchar(20)(문자열)	맛집 유형(최대 20 문자 크기)
score	integer(정수형)	맛집 평점
region	varchar(20)(문자열)	맛집 지역(최대 20 문자 크기)
phone	varchar(20)(문자열)	맛집 연락처(최대 20 문자 크기)
address	varchar(30)(문자열)	맛집 주소(최대 30 문자 크기)
memo	varchar(200)(문자열)	맛집 관련 메모(최대 200 문자 크기)
pic	varchar(50)(문자열)	맛집 사진 파일 경로명(최대 50 문자 크기)



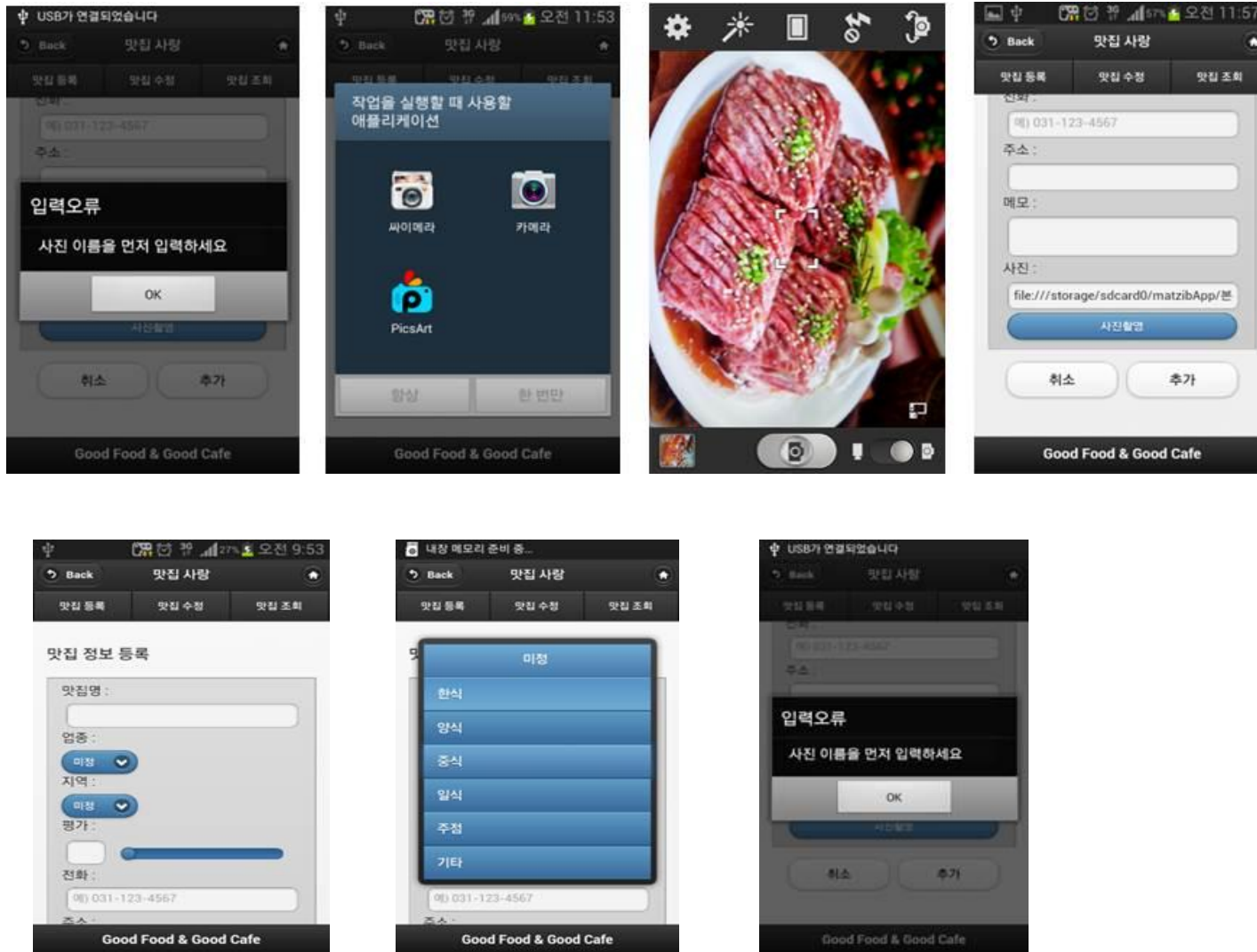
## 1.3 맛집 정보 등록하기

- 이진(binary) 데이터의 2가지 DB 저장 방법
  - SQLite 데이터베이스의 테이블 안에 직접 저장하는 방식
  - 로컬 파일 시스템에 이진 파일을 저장하고 파일 경로 정보만 DB에 저장하는 방식
- BLOB(Binary Large Object) 유형
  - 많은 데이터베이스에서 지원
  - 문자형과 비슷하며 이진 데이터를 저장할 수 있다는 점만 다름
  - DB에 저장하려면 이진 객체를 직렬화 즉, 문자열로 변환(Base64로 인코딩해서 저장)해야 함
    - Base64는 8비트 이진 데이터를 문자 코드에 영향받지 않는 공통 아스키 문자로만 구성된 일련의 문자열로 바꾸는 인코딩 방식(화면에 출력할 수 있는 64개 문자(ABCD...89+/e 등)들만을 이용해 이진 데이터를 문자열로 표현)



# 맛집 사진 정보 등록

## ● [그림 16-5] matzipApp 앱의 맛집 사진 등록



# 파일 시스템 폰갭 API

- 파일(디렉토리) 검색

```
window.resolveLocalFileSystemURI(URI, 성공콜백함수, 실패콜백함수); // FileEntry(DirectoryEntry) 반환
```

- FileSystem 객체 얻기

```
window.requestFileSystem(type, size, 성공콜백함수, 실패콜백함수); // FileSystem 객체 반환
```

- 디렉토리 찾기(생성하기)

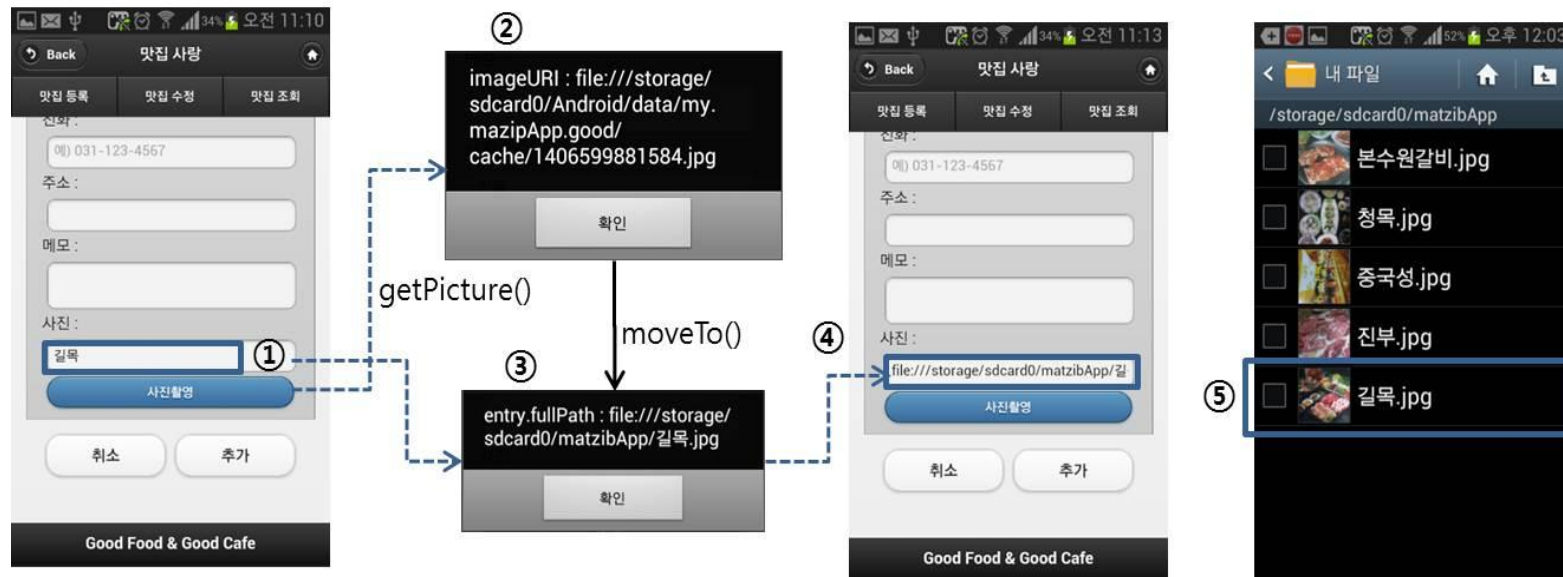
```
getDirectory(path, options, 성공콜백함수, 실패콜백함수); // DirectoryEntry 객체 반환
```

- 파일 이동하기

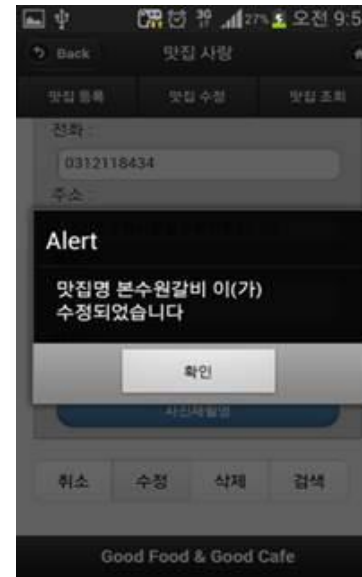
```
moveTo(parent, newName, 성공콜백함수, 실패콜백함수); // FileEntry 객체 반환
```

# 사진 파일 저장

## ● moveTo() 사진 파일 영구 저장



# 1.4 맛집 정보 수정하기



# 1.5 맛집 목록 조회하기

## ● [그림 16-6] matzipApp 앱 맛집 목록 검색



## 1.6 맛집 상세정보 조회하기



## 1.7 맛집 추가정보 조회하기

- 맛집 사진정보 검색하기
  - 맛집 지도정보 검색하기
  - 맛집 경로정보 검색하기
- [그림 16-7] matzipApp 앱의 맛집 상세정보 보기

