

ReadyRun iOS App 상세기획서

1. 시스템 아키텍처



2. API 명세서

2.1 마라톤 관련 API

GET /api/v1/marathons

- **설명:** 마라톤 목록 조회
- **Query Parameters:**
 - `page`: 페이지 번호
 - `limit`: 페이지 당 항목 수
 - `sort`: 정렬 기준
 - `country`: 국가 필터
 - `distance`: 거리 필터
 - `date_from`: 시작 날짜
 - `date_to`: 종료 날짜

GET /api/v1/marathons/{id}

- **설명:** 특정 마라톤 상세 정보 조회

GET /api/v1/marathons/search

- **설명:** 마라톤 검색
- **Query Parameters:**
 - `q`: 검색 키워드
 - `filters`: 필터 조건

2.2 즐겨찾기 관련 API

POST /api/v1/favorites

- **설명:** 즐겨찾기 추가

DELETE /api/v1/favorites/{marathon_id}

- **설명:** 즐겨찾기 삭제

GET /api/v1/favorites

- **설명:** 사용자 즐겨찾기 목록 조회

2.3 관리자 관련 API

GET /api/v1/admin/marathons/pending

- **설명:** 승인 대기 중인 마라톤 목록 조회
- **권한:** 관리자만 접근 가능

POST /api/v1/admin/marathons/{id}/approve

- **설명:** 마라톤 승인
- **권한:** 관리자만 접근 가능

POST /api/v1/admin/marathons/{id}/reject

- **설명:** 마라톤 승인 거부
- **권한:** 관리자만 접근 가능

GET /api/v1/admin/users

- **설명:** 사용자 목록 조회
- **권한:** 관리자만 접근 가능

3. 데이터 모델





3.1 마라톤 데이터 구조 (JSON)

json

```
{
  "marathon": {
    "id": "string",
    "name": "string",
    "date": "ISO 8601",
    "location": {
      "country": "string",
      "city": "string",
      "address": "string",
      "coordinates": {
        "lat": "number",
        "lng": "number"
      }
    },
    "distances": ["5K", "10K", "Half", "Full"],
    "registration": {
      "fee": "number",
      "currency": "string",
      "deadline": "ISO 8601",
      "url": "string"
    },
    "images": {
      "thumbnail": "string",
      "hero": "string",
      "course_map": "string"
    },
    "metadata": {
      "created_at": "ISO 8601",
      "updated_at": "ISO 8601",
      "source": "string",
      "confidence_score": "number"
    }
  }
}
```

4. 디자인 시스템

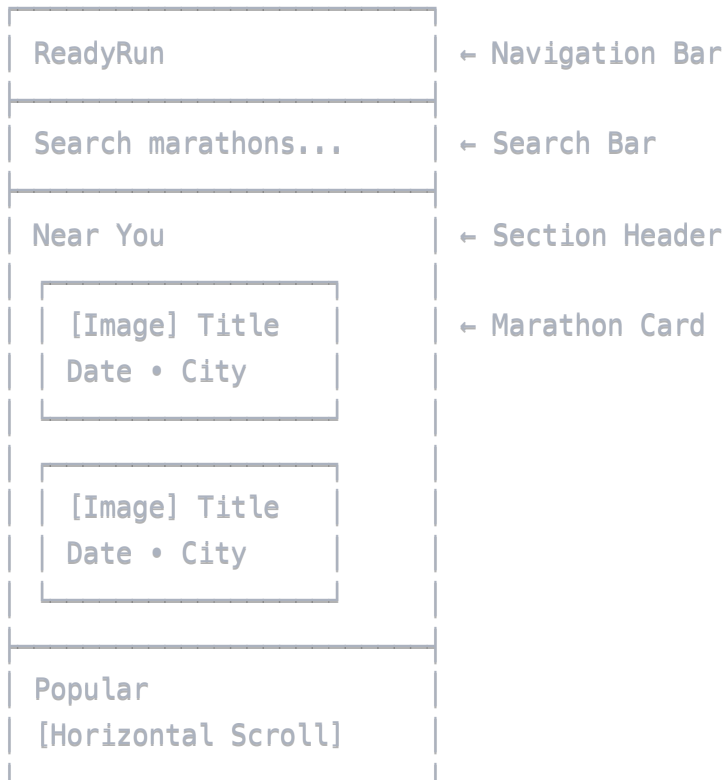
4.1 컬러 팔레트

- Primary Blue:  #007AFF
- Secondary Orange:  #FF6B35
- Success Green:  #34C759
- Warning Orange:  #FF9500

- **Error Red:** ● #FF3B30
- **Background Gray:** #F2F2F7
- **Text Black:** ● #000000

4.2 주요 화면 구성

메인 화면



검색/필터 화면



5. 데이터베이스 스키마

5.1 사용자 테이블 (users)

sql

```
CREATE TABLE users (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  firebase_uid VARCHAR(255) UNIQUE NOT NULL,  
  email VARCHAR(255),  
  display_name VARCHAR(100),  
  preferred_language VARCHAR(5) DEFAULT 'en',  
  preferred_units VARCHAR(10) DEFAULT 'metric', -- metric, imperial  
  push_notifications_enabled BOOLEAN DEFAULT true,  
  created_at TIMESTAMP DEFAULT NOW(),  
  updated_at TIMESTAMP DEFAULT NOW(),  
  last_login_at TIMESTAMP  
);
```

5.2 관리자 사용자 테이블 (admin_users)

sql

```
CREATE TABLE admin_users (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  firebase_uid VARCHAR(255) UNIQUE NOT NULL,  
  email VARCHAR(255) NOT NULL,  
  display_name VARCHAR(100),  
  role VARCHAR(20) DEFAULT 'admin', -- admin, super_admin  
  permissions TEXT[], -- ['marathon_approval', 'user_management', 'system_admin']  
  created_at TIMESTAMP DEFAULT NOW(),  
  updated_at TIMESTAMP DEFAULT NOW(),  
  last_login_at TIMESTAMP,  
  created_by UUID REFERENCES admin_users(id),  
  is_active BOOLEAN DEFAULT true  
);
```

5.3 마라톤 테이블 (marathons)

```
sql
```

```
CREATE TABLE marathons (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  name VARCHAR(255) NOT NULL,  
  date_start TIMESTAMP NOT NULL,  
  date_end TIMESTAMP,  
  country VARCHAR(100) NOT NULL,  
  city VARCHAR(100) NOT NULL,  
  address TEXT,  
  latitude DECIMAL(10, 8),  
  longitude DECIMAL(11, 8),  
  distances TEXT[], -- ['5K', '10K', 'Half', 'Full']  
  registration_fee INTEGER, -- cents  
  currency VARCHAR(3) DEFAULT 'USD',  
  registration_deadline TIMESTAMP,  
  registration_url TEXT,  
  official_website TEXT,  
  thumbnail_url TEXT,  
  hero_image_url TEXT,  
  course_map_url TEXT,  
  description TEXT,  
  organizer VARCHAR(255),  
  max_participants INTEGER,  
  status VARCHAR(20) DEFAULT 'pending', -- pending, active, cancelled, completed, re.  
  created_at TIMESTAMP DEFAULT NOW(),  
  updated_at TIMESTAMP DEFAULT NOW(),  
  source VARCHAR(100), -- manual, ai_crawler  
  confidence_score INTEGER DEFAULT 100, -- 0-100  
  approved_by UUID REFERENCES admin_users(id),  
  approved_at TIMESTAMP,  
  rejection_reason TEXT  
);
```

5.4 즐겨찾기 테이블 (favorites)

```
sql
```

```
CREATE TABLE favorites (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,  
  marathon_id UUID NOT NULL REFERENCES marathons(id) ON DELETE CASCADE,  
  created_at TIMESTAMP DEFAULT NOW(),  
  UNIQUE(user_id, marathon_id)  
);
```

5.5 마라톤 승인 이력 테이블 (marathon_approvals)

sql

```
CREATE TABLE marathon_approvals (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  marathon_id UUID NOT NULL REFERENCES marathons(id) ON DELETE CASCADE,  
  admin_user_id UUID NOT NULL REFERENCES admin_users(id),  
  action VARCHAR(20) NOT NULL, -- approved, rejected, modified  
  notes TEXT,  
  previous_status VARCHAR(20),  
  new_status VARCHAR(20),  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

6. 엔티티 관계도

```
User ( )  
├─ Favorites ( ) ↔ Marathon ( )  
└─ Notifications ( )
```

```
Marathon ( )  
├─ Location ( )  
├─ Distances ( )  
├─ Registration ( )  
├─ Images ( )  
└─ ApprovalHistory ( )
```

```
AdminUser ( )  
├─ MarathonApproval ( )  
└─ CreatedAdmins ( )
```

```
MarathonApproval ( )  
├─ Marathon ( )  
└─ AdminUser ( )
```

7. 주요 기능 명세

7.1 사용자 기능

- 마라톤 검색 및 필터링
- 마라톤 상세 정보 조회
- 즐겨찾기 추가/삭제
- 위치 기반 마라톤 추천

- 푸시 알림 설정

7.2 관리자 기능

- 마라톤 승인/거부
- 사용자 관리
- 데이터 품질 관리
- AI 크롤러 결과 검토
- 시스템 통계 조회

7.3 AI 크롤러 기능

- 외부 소스에서 마라톤 정보 수집
- 데이터 정확성 검증
- 중복 데이터 제거
- 신뢰도 점수 계산

8. 보안 및 권한 관리

8.1 인증

- Firebase Authentication 사용
- JWT 토큰 기반 인증

8.2 권한 레벨

- 일반 사용자: 마라톤 조회, 즐겨찾기 관리
- 관리자: 마라톤 승인, 사용자 관리
- 슈퍼 관리자: 관리자 계정 생성, 시스템 전체 관리

8.3 API 보안

- CORS 설정
- Rate Limiting
- Input Validation
- SQL Injection 방지

9. 성능 최적화

9.1 데이터베이스 인덱스

sql

-- 마라톤 검색 최적화

```
CREATE INDEX idx_marathons_date_country ON marathons(date_start, country);
```

```
CREATE INDEX idx_marathons_location ON marathons(latitude, longitude);
```

```
CREATE INDEX idx_marathons_status ON marathons(status);
```

-- 즐겨찾기 최적화

```
CREATE INDEX idx_favorites_user_id ON favorites(user_id);
```

-- 관리자 기능 최적화

```
CREATE INDEX idx_marathons_pending ON marathons(status) WHERE status = 'pending';
```

9.2 캐싱 전략

- Redis를 이용한 API 응답 캐싱
- 이미지 CDN 활용
- 정적 리소스 캐싱

10. 모니터링 및 로깅

10.1 로그 수집

- API 요청/응답 로그
- 에러 로그
- 사용자 행동 분석
- 관리자 액션 로그

10.2 메트릭

- API 응답 시간
- 데이터베이스 성능
- 사용자 활성도
- 마라톤 등록률