

우리무리(UriMuri / OurGroup) 프로젝트 완전 문서화

1. 프로젝트 개요 및 정의

1.1 프로젝트 배경

우리무리(UriMuri)는 한국의 단체 활동 환경에 특화된 실시간 위치 추적 애플리케이션입니다. 학교 현장학습, 산악회 등반, 기업 워크샵 등 다양한 단체 활동에서 발생할 수 있는 인원 누락이나 실종 사고를 예방하기 위해 개발됩니다.

기존의 글로벌 위치 공유 앱들(Life360, Glympse 등)이 가족 단위나 소규모 그룹에 최적화되어 있다면, 우리무리는 한국의 집단주의 문화와 교육 현장의 특수성을 반영한 대규모 단체 관리에 특화되어 있습니다.

1.2 핵심 가치 제안

- **실시간 안전 관리:** 방장(인솔자)이 모든 구성원의 위치를 실시간으로 모니터링
- **자동화된 추적:** 참가자가 별도 조작 없이 백그라운드에서 자동 위치 전송
- **응급상황 대응:** 즉시 알림을 통한 신속한 상황 파악 및 대응
- **배터리 최적화:** 장시간 활동에도 스마트폰 배터리 부담 최소화

1.3 목표 사용자

주 사용자 (방장/인솔자):

- 초중고 교사 및 인솔 교사
- 산악회 리더 및 가이드
- 기업 워크샵 주최자
- 동호회 모임 주관자

보조 사용자 (참가자):

- 학생들
- 산악회 회원들
- 직장인들
- 일반 참가자들

2. 비즈니스 요구사항 및 기능 정의

2.1 핵심 비즈니스 요구사항

안전성 요구사항:

- 구성원의 실시간 위치를 99% 이상의 정확도로 추적
- 응급상황 발생 시 5초 이내에 방장에게 알림 전달
- 네트워크 연결이 불안정한 환경에서도 최종 위치 정보 보존

사용성 요구사항:

- 6자리 간단한 코드만으로 그룹 참여 가능
- 70세 이상 고령자도 쉽게 사용할 수 있는 직관적 인터페이스
- 한 번의 설정으로 하루 종일 자동 동작

성능 요구사항:

- 배터리 소모량을 일반 사용 대비 20% 이내로 제한
- 30초 이내 간격으로 위치 정보 업데이트
- 동시에 100명까지 한 그룹에서 추적 가능

2.2 MVP 기능 범위

1단계 필수 기능:

- 방 생성 및 6자리 코드 기반 참여 시스템
- 사용자 프로필 등록 (이름, 전화번호, 색상 선택)
- 실시간 지도 기반 위치 공유
- 백그라운드 위치 추적
- 방장의 구성원 승인/거절 권한
- 응급상황 즉시 알림 기능

2단계 추가 기능 (향후 개발):

- 안전 구역 설정 및 이탈 알림
- 음성 안내 시스템
- 대중교통 연계 정보
- 활동 기록 및 분석

2.3 제외되는 기능

- 메시징 및 채팅 기능
- 소셜 네트워킹 요소
- 복잡한 권한 관리 시스템
- 개인 활동 추적 및 분석

3. 기술 아키텍처 및 설계 철학

3.1 기술 스택 선택 근거

React Native 선택 이유:

- 개발 리소스 효율성: iOS와 Android를 동시에 개발
- 빠른 프로토타이핑: JavaScript 기반으로 신속한 MVP 개발
- 풍부한 위치 기반 라이브러리 생태계
- Firebase와의 원활한 연동 지원

Firebase 백엔드 선택 이유:

- 서버리스 아키텍처로 초기 운영 비용 최소화
- Realtime Database로 실시간 위치 동기화 최적화
- 자동 확장성으로 사용자 증가에 대응
- 한국 내 안정적인 서비스 품질 보장

3.2 데이터 아키텍처 설계

계층적 데이터 구조:

```

Firebase Realtime Database
├── users (사용자 프로필 정보)
│   ├── 개인정보 (이름, 전화번호, 프로필 색상)
│   └── 계정 생성 시간
├── rooms (방 정보)
│   ├── 방 메타데이터 (방장 ID, 생성 시간)
│   ├── 승인된 구성원 목록 및 위치 정보
│   ├── 승인 대기 목록
│   ├── 일반 알림 (배터리, 네트워크 상태)
│   └── 응급상황 알림
```

데이터 흐름 설계:

1. 클라이언트에서 위치 정보 수집
2. Firebase Realtime Database로 실시간 전송
3. 다른 클라이언트들이 실시간으로 위치 정보 수신
4. 지도 UI에서 시각적으로 표현

3.3 보안 및 개인정보보호 설계

데이터 접근 제어:

- 방 구성원만 해당 방의 데이터 접근 가능
- 개인 위치 정보는 본인과 방장만 수정 가능
- 방 해체 시 모든 위치 데이터 자동 삭제

개인정보보호 원칙:

- 최소한의 개인정보만 수집 (이름, 선택적 전화번호)
- 위치 정보는 활동 종료 후 즉시 삭제
- 사용자 동의 없는 위치 추적 절대 금지

4. 개발 프로세스 및 방법론

4.1 개발 방법론

애자일 스크럼 방식 채택:

- 1주 단위 스프린트로 진행
- 매 스프린트마다 동작 가능한 기능 완성
- 사용자 피드백을 즉시 다음 스프린트에 반영

MVP 우선 개발 전략:

- 가장 핵심적인 기능부터 순차적 개발
- 각 기능의 기본 동작 확인 후 다음 단계 진행
- 완벽함보다는 빠른 검증과 개선에 집중

4.2 4주 개발 로드맵 상세

1주차: 기반 인프라 구축

- 개발 환경 설정 및 프로젝트 초기화
- Firebase 프로젝트 생성 및 연동 테스트
- 기본 화면 네비게이션 구조 구현
- 사용자 인증 시스템 (익명 로그인) 구축
- 프로필 등록 화면 개발

2주차: 핵심 위치 추적 기능

- 위치 권한 요청 및 관리 시스템
- iOS/Android 크로스플랫폼 위치 추적 서비스
- Firebase Realtime Database 연동
- 방 생성 및 참여 로직 구현

- 기본 지도 화면 및 마커 표시

3주차: 사용자 경험 완성

- 지도 인터페이스 고도화 (구성원 정보 표시)
- 실시간 위치 업데이트 및 동기화
- 응급상황 알림 시스템
- 방 관리 기능 (승인/거절/강퇴)
- 위치 공유 제어 기능

4주차: 품질 향상 및 배포 준비

- 전체 기능 통합 테스트
- 배터리 최적화 및 성능 튜닝
- 다양한 네트워크 환경 테스트
- 사용자 인터페이스 최종 개선
- 앱 스토어 배포 자료 준비

4.3 테스트 전략

기능 테스트:

- 각 화면별 사용자 시나리오 테스트
- 다양한 기기 사양에서의 성능 테스트
- 네트워크 연결 불안정 상황 시뮬레이션

성능 테스트:

- 배터리 소모량 지속 모니터링
- 메모리 사용량 및 누수 검사
- 대량 사용자 동시 접속 테스트

5. 핵심 기능별 동작 프로세스

5.1 방 생성 및 참여 프로세스

방 생성 과정:

1. 방장이 "방 만들기" 버튼 선택
2. 시스템이 6자리 고유 코드 자동 생성 (중복 검사 포함)
3. Firebase에 방 정보 저장 (방장 ID, 생성 시간)
4. 방장이 자동으로 첫 번째 구성원으로 등록
5. 위치 추적 서비스 즉시 시작
6. 생성된 코드를 다른 참가자들에게 공유

방 참여 과정:

1. 참가자가 "방 참여하기" 선택 후 6자리 코드 입력
2. 시스템이 코드 유효성 검증 (존재 여부, 형식 확인)
3. 참여 요청이 "대기 목록"에 추가
4. 방장에게 새 참여 요청 알림 전송
5. 방장이 승인/거절 결정
6. 승인 시 참가자가 구성원 목록에 추가되고 위치 추적 시작

5.2 백그라운드 위치 추적 메커니즘

위치 수집 프로세스:

1. 앱 시작 시 사용자에게 위치 권한 요청
2. 사용자 동의 후 위치 추적 서비스 활성화
3. GPS를 통해 현재 위치 좌표 수집 (위도, 경도, 정확도)
4. 수집된 위치 정보를 Firebase로 실시간 전송
5. 다른 구성원들의 앱에서 즉시 위치 정보 수신 및 지도 업데이트

플랫폼별 백그라운드 처리:

- **iOS:** watchPosition API를 활용하여 백그라운드에서도 지속적 위치 추적
- **Android:** Background Job Service와 Foreground Notification을 조합하여 안정적 추적

배터리 최적화 전략:

- 사용자가 정지 상태일 때 위치 업데이트 간격 자동 조정
- 움직임 감지 센서를 활용하여 불필요한 GPS 사용 최소화
- 위치 정확도를 활동 유형에 맞게 동적 조절

5.3 실시간 지도 시각화 프로세스

지도 표시 로직:

1. React Native Maps를 기반으로 한국 지역 지도 로드
2. Firebase에서 실시간으로 구성원 위치 정보 수신
3. 각 구성원을 고유 색상의 마커로 지도에 표시
4. 마커 클릭 시 구성원 상세 정보 (이름, 전화번호, 마지막 업데이트 시간) 표시
5. 현재 사용자 위치 중심으로 50미터 반경 원 표시

자동 화면 조정:

- 새 구성원 추가 시 모든 구성원이 화면에 보이도록 지도 범위 자동 조정
- 구성원이 크게 분산된 경우 적절한 줌 레벨로 자동 설정
- 방장은 특정 구성원 터치로 해당 위치로 즉시 이동 가능

5.4 응급상황 알림 시스템

응급상황 발생 프로세스:

1. 참가자가 지도 화면의 "응급상황" 버튼 터치
2. 확인 다이얼로그를 통한 오발송 방지
3. 사용자 확인 시 Firebase에 응급 알림 데이터 즉시 전송
4. 방장의 앱에서 Firebase 리스너가 응급 알림 감지
5. 방장에게 팝업 알림과 함께 발신자 위치로 지도 자동 이동
6. 필요 시 방장이 다른 구성원들에게 상황 전파 가능

알림 우선순위 처리:

- 응급상황 알림은 모든 다른 알림보다 최우선 처리
- 방장의 앱이 백그라운드 상태여도 강제로 포그라운드로 전환
- 연속된 응급 알림 발생 시 시간순으로 처리하되 최신 알림 우선 표시

6. 사용자 경험 및 인터페이스 설계

6.1 사용자 여정(User Journey) 설계

방장(인솔자)의 여정:

1. 앱 설치 후 프로필 등록 (이름, 연락처, 색상 선택)
2. "방 만들기" 선택하여 새 그룹 생성
3. 생성된 6자리 코드를 참가자들에게 공유 (문자, 구두 전달 등)
4. 참가자들의 참여 요청을 개별 승인/거절
5. 모든 구성원의 실시간 위치를 지도에서 모니터링
6. 필요 시 특정 구성원에게 집중하여 위치 확인
7. 활동 종료 후 방 해체로 모든 데이터 삭제

참가자의 여정:

1. 앱 설치 후 프로필 등록
2. 방장으로부터 받은 코드로 "방 참여하기"
3. 방장의 승인 대기 (자동 알림으로 승인 상태 확인)
4. 승인 후 자동으로 위치 공유 시작
5. 지도에서 다른 구성원들 위치 확인 가능
6. 필요 시 응급상황 버튼으로 즉시 도움 요청
7. 활동 종료 시 자동으로 위치 공유 중단

6.2 직관적 인터페이스 설계 원칙

단순성 우선:

- 한 화면에 하나의 주요 기능만 집중
- 텍스트보다는 아이콘과 색상을 활용한 시각적 정보 전달
- 복잡한 메뉴 구조 대신 버튼 기반의 직관적 조작

접근성 고려:

- 고령자도 쉽게 인식할 수 있는 큰 폰트 사이즈
- 색맹 사용자를 위한 색상 외 구분 요소 추가
- 터치 영역을 충분히 크게 설정하여 오터치 방지

한국 문화 반영:

- 한글 폰트 최적화 및 자연스러운 어순
- 한국인이 익숙한 색상 조합 및 디자인 패턴
- 존댓말 기반의 정중한 안내 메시지

7. 기술적 도전과제 및 해결방안

7.1 백그라운드 위치 추적의 기술적 한계

iOS 플랫폼 제약사항:

- 백그라운드 위치 사용 시 상태바에 파란색 표시 상시 노출
- 사용자에게 주기적으로 위치 사용 알림 표시
- 앱이 완전히 종료되면 위치 추적도 중단

해결 방안:

- 사용자에게 백그라운드 위치 사용의 필요성 명확히 설명
- 배터리 최적화를 통해 사용자 불편 최소화
- 앱 종료 방지를 위한 사용자 교육 및 가이드 제공

Android 플랫폼 제약사항:

- Android 10 이상에서 백그라운드 위치 권한 별도 요청 필요
- Foreground Service 사용 시 알림창 상시 표시 의무
- 제조사별 배터리 최적화 정책으로 인한 앱 강제 종료

해결 방안:

- 제조사별 배터리 최적화 해제 가이드 제공
- Foreground Service 알림을 최대한 간결하고 유용하게 설계
- 주요 제조사 기기에서의 충분한 테스트 수행

7.2 네트워크 연결 불안정 환경 대응

문제 상황:

- 산간 지역이나 지하 공간에서의 불안정한 인터넷 연결
- 일시적 네트워크 단절로 인한 위치 정보 손실
- 대량의 데이터 전송으로 인한 요금 부담

해결 전략:

- 오프라인 상황에서 위치 정보를 로컬에 임시 저장
- 네트워크 복구 시 저장된 위치 데이터 일괄 동기화
- 위치 업데이트 간격을 네트워크 상태에 따라 동적 조정
- 필수 데이터만 전송하여 데이터 사용량 최소화

7.3 대규모 동시 사용자 처리

확장성 고려사항:

- 한 그룹에 100명 이상 참여 시 성능 저하 가능성
- Firebase Realtime Database의 동시 연결 제한
- 실시간 업데이트로 인한 과도한 트래픽 발생

성능 최적화 방안:

- 지도 화면에서 보이는 영역의 마커만 렌더링
- 위치 업데이트 빈도를 사용자 움직임에 따라 조절
- Firebase의 자동 확장 기능을 활용한 트래픽 분산
- 필요 시 Firebase Functions를 통한 서버 사이드 처리

8. 운영 및 유지보수 계획

8.1 모니터링 및 분석 체계

성능 모니터링:

- Firebase Analytics를 통한 사용자 행동 분석
- 앱 크래시 및 오류 자동 수집 및 분석
- 배터리 사용량 및 네트워크 트래픽 모니터링
- 사용자 피드백 수집 및 분류 시스템

품질 지표 관리:

- 위치 추적 정확도 (목표: 95% 이상)
- 앱 안정성 (크래시율 1% 미만)
- 응답 시간 (위치 업데이트 30초 이내)

- 사용자 만족도 (앱 스토어 평점 4.0 이상)

8.2 지속적 개선 전략

사용자 피드백 반영:

- 앱 내 피드백 기능을 통한 즉시 의견 수렴
- 교육 현장 및 산악회와의 정기적 사용자 인터뷰
- A/B 테스트를 통한 UI/UX 개선 효과 검증

기능 확장 로드맵:

- 2단계: 안전 구역 설정 및 이탈 알림 기능
- 3단계: 음성 안내 및 대중교통 연계 서비스
- 4단계: AI 기반 위험 상황 예측 및 알림

8.3 사업적 지속가능성

수익 모델:

- 기본 기능은 무료 제공으로 사용자 확보
- 프리미엄 기능 (대용량 그룹, 고급 분석) 유료 제공
- 교육기관 및 기업 대상 B2B 서비스 확장

파트너십 전략:

- 교육청 및 학교와의 공식 파트너십 구축
- 산악회 연합회 및 등산 관련 업체와의 협력
- 관광청 및 지자체와의 안전 관광 사업 연계

이 종합 문서는 우리무리 프로젝트의 전체적인 비전부터 구체적인 구현 방법까지 모든 측면을 다루고 있으며, 개발팀이 일관된 방향성을 가지고 프로젝트를 진행할 수 있도록 상세한 가이드라인을 제공합니다.