

심리상담 챗봇 - PsyChat

202404178 강우현

202404273 이유림

202404203 김예원

(강남대학교 인공지능전공)

차례

- I. 서론
- II. 이론적 배경 및 선행연구
- III. 시스템 설계 및 구현
 - 1. 요구사항 분석
 - 2. 아키텍처 개요
 - 3. 데이터 모델 및 저장소 설계
 - 4. UI/UX 설계
 - 5. 핵심 컴포넌트 설명
 - 6. 보안 및 개인정보보호 고려사항
- IV. 실험 및 평가
 - 1. 테스트 항목 및 방법
 - 2. 성능 측정 결과
- V. 논의
 - 1. 주요 발견
 - 2. 한계
 - 3. 개선 및 확장 방향
- VI. 결론

I . 서론

최근 인공지능 기술과 모바일 헬스케어 서비스의 확산은 개인의 정신 건강 관리 방식에 큰 변화를 가져왔다. 기존의 정신 건강 관리는 전문 상담 기관의 도움을 직접적으로 받는 방식이 주류였지만, 모바일 기반 솔루션은 일상생활 속에서 감정을 기록하고 분석하는 과정을 손쉽게 수행할 수 있게 함으로써 자기 관찰(Self-monitoring)과 자가치유(Self-healing)를 가능하게 하고 있다. 특히 스마트폰을 활용한 감정 추적은 사용자가 특정 시점의 감정 상태를 기록하고 시간이 지남에 따라 변화를 확인할 수 있다는 점에서 심리학적 실천과 디지털 헬스케어의 접점을 제공한다.

본 연구에서는 실사용 상황에서 감정 기록을 직관적으로 수행할 수 있는 UI/UX 설계 및

구현, Firebase 기반 데이터 저장소 설계와 안전한 인증 흐름 구현, 누적 감정 데이터를 Pie Chart 및 달력/히스토리 형태로 시각화하여 사용자 인사이트 제공, 개발·테스트 과정을 통해 애플리케이션의 안정성, 응답성, 데이터 무결성 확보 방안 제시를 목적으로 연구를 진행하였다.

II. 이론적 배경 및 선행연구

감정 기록 및 자기관찰(Self-monitoring)은 자기관찰은 행동치료 및 정신건강 관리에서 핵심 기법으로, 정기적 기록이 행동/정서 패턴 인식에 기여한다. 모바일 애플리케이션은 접근성과 사용 편의성으로 인해 전통적 설문지 대비 더 높은 빈도로 데이터를 수집할 수 있다.

모바일 정신건강 앱 관련 연구에서는 기존 연구는 일별/시간대별 기분 추적, 간단한 문진 기반 진단, 맞춤형 알림 기능 등이 효율적이라는 점을 보고했다. AI 기반 정서 분석 (텍스트 감정분석, 음성/표정 기반 감정 인식)은 자동화된 피드백을 제공하나 개인정보·윤리적 문제를 동반하였다.

Firebase의 활용 사례로는 Firebase가 빠른 프로토타이핑, 실시간 DB 동기화, 손쉬운 인증 구현을 지원하여 초기 스타트업·학술 프로젝트에서 널리 사용되었다.

III. 시스템 설계 및 구현

1. 요구사항 분석

기능적 요구사항(Functional Requirements)에는 FR1: 사용자 가입/로그인 (Firebase Authentication), FR2: 감정 입력(텍스트/이모지/라디오 선택), FR3: 감정 데이터 저장 및 불러오기(날짜별), FR4: 통계 시각화(파이 차트, 캘린더 뷰, 히스토리 리스트), FR5: 데이터 삭제/수정 기능이 있다.

비기능적 요구사항 (Non-functional Requirements)에는 NFR1: 응답 시간 2초 이내(일반적 UI 동작), NFR2: 데이터 무결성 보장(쓰기는 원자적), NFR3: 인증된 사용자만 데이터 접근 가능, NFR4: 최소한의 오프라인 지원(캐시), 다음 온라인 시 동기화 기능이 있다.

2. 아키텍처 개요

간단한 3계층 아키텍처이다. Presentation Layer (React Native UI), Application Layer (상태관리: React Context/Redux 또는 간단한 상태 흙), Data Layer (Firebase Auth + Realtime Database)로 구성된다. 데이터 흐름으로는 [사용자가 감정 입력 → 앱 UI에서 로컬 상태 업데이트 → Firebase에 쓰기 → 성공 시 로컬/화면에 반영] 이렇게 구성된다.

3. 데이터 모델 및 저장소 설계

아래 그림은 은 데이터 스키마 Realtime Database의 예시이다. emotion은 카테고리(예: happy, sad, angry, anxious, neutral)로 표준화, intensity는 1~5 스케일, records 키를 날짜별로 묶으면 날짜 조회에 효율적이다.

```
{  
  "users": {  
    "uid_abcdef": {  
      "profile": {  
        "displayName": "kang",  
        "email": "kang@example.com"  
      },  
      "records": {  
        "2025-12-01": [  
          {  
            "id": "rec_001",  
            "time": "2025-12-01T08:17:00+09:00",  
            "emotion": "happy",  
            "intensity": 3,  
            "note": "아침 기분 좋음"  
          },  
          { ... }  
        ],  
        "2025-12-02": [ ... ]  
      }  
    }  
  }  
}
```

4. UI/UX 설계

화면 흐름으로는 StartScreen → Login/Signup → HomeScreen이다. HomeScreen은 오늘의 감정 기록 버튼, 최근 요약(위젯), ChatScreen은 채팅 형태로 감정 입력 (ChatBubble 컴포넌트), HistoryScreen은 목록(날짜별), GraphScreen은 파일 차트 및 월간 캘린더 뷰, MyPage는 계정 관리, 로그아웃 역할을 수행한다.

주요 UI 컴포넌트에는 MoodPalette는 이모지 버튼 + 강도 슬라이더, ChatBubble은 원/오른쪽 대화풍 UI (입력/시스템 메시지), PieChartComponent는 Recharts 혹은 react-native-chart-kit 기반 (본 프로젝트는 Pie 차트 라이브러리 사용), CalendarView는 날짜 선택시 해당 날짜 기록 표시하는 기능을 한다.

UX 고려사항에는 입력 최소화가 있다. 이는 텍스트 입력을 강요하지 않고 빠른 이모지 선택을 우선 제공한다. 그리고 시각화 직관성이 있다. 이는 금일/주간/월간 뷰를 명확히 구분짓는 역할을 한다. 마지막으로 피드백 루프가 있다. 이는 기록 후 간단한 위로/추천 메시지(정책템플릿)를 제공한다.

5. 핵심 컴포넌트 설명

firebase.js

- Firebase 초기화 (apiKey 등 환경변수로 분리 권장).
- signInWithEmailAndPassword, createUserWithEmailAndPassword, signOut 등 인증 함수 래핑.
- DB 함수: saveRecord(uid, date, record), getRecords(uid, date), getSummary(uid, month).

MainStack.js

- React Navigation Stack 사용.
- Auth 흐름과 App 흐름을 분리: 인증 상태에 따라 초기 화면 결정.

ChatScreen.js

- 사용자가 감정 선택 → onSubmit 시 로컬 상태에 임시 저장 → saveRecord 호출하여 Firebase에 쓰기.
- 쓰기 실패 시 재시도/로컬 큐에 적재하여 재전송.

GraphScreen.js, HistoryScreen.js

- GraphScreen은 monthly summary API를 호출하여 파이차트 데이터를 구성.
- HistoryScreen은 날짜별 리스트 렌더링 및 삭제/수정 가능.

상태관리

- 간단한 프로젝트이므로 React Context + useReducer 권장.
- 전역상태: user, records, uiFlags.

6. 보안 및 개인정보보호 고려사항

첫 번째로, 민감정보(정신건강 데이터)는 개인 데이터로 간주한다. 두 번째로, Firebase 보안 규칙 설정은 인증된 사용자만 자신의 데이터에 접근 가능하도록 규칙을 추가한다. 세 번째로, 전송 시 HTTPS(기본 Firebase 제공) 사용한다. 네 번째로, 앱내 데이터 캐싱 시 암호화 저장(가능하면)을 권장한다. 마지막으로, 개인정보 처리방침 및 데이터 보유·삭제 옵션을 제공할 필요성이 있다.

IV. 실험 및 평가

1. 테스트 항목 및 방법

기능 테스트 (Unit / Integration)

- 각 컴포넌트 유닛 테스트 (Jest + React Native Testing Library 권장).
- Firebase 연동 테스트: 쓰기/읽기/동시성 테스트.
- 네비게이션 흐름 테스트: 인증이 끊겼을 때 화면 이동 등.

성능 테스트

- 데이터 로딩 시간(초) 측정: 한 달치 데이터 로딩시 평균 응답시간.
- DB 쓰기 실패 시 재시도 메커니즘의 성공률.

사용자 평가 (사용성 테스트)

- 대상: 15-30명(학생/교직원 표본 권장).
- 방법: 앱 사용 후 SUS(System Usability Scale) 설문 + 자유 응답.
- 핵심 지표: 사용성(SUS), 입력 빈도(일별 기록률), 재방문율(7일 내).

성능 측정 결과(가상)

- 평균 감정 저장 응답 시간: 0.5-1.2초(네트워크 상황에 따라 변동).
- 파이 차트 렌더링 시간: <100ms(클라이언트 처리).
- 동시성 쓰기(100건 동시): Firebase는 실시간 DB 특성상 높은 처리량에서 충돌 가능 - 트랜잭션/무결성 검증 필요.

V. 논의

1. 주요 발견

직관적인 이모지 기반 입력과 채팅형 UI는 사용자의 기록 진입 장벽을 낮추는 데 효과적이다. 그리고 날짜별로 기록을 묶는 DB 스키마는 조회 효율성과 UI 렌더링 효율을 모두 만족시킨다. 또한 Firebase의 손쉬운 인증+실시간 DB는 프로토타이핑에 유리하지만, 민감 데이터 운영 시 보안 규칙과 규정 준수가 필수적이다.

2. 한계

현재 버전은 텍스트 기반 감정 자동 분류(자연어처리) 또는 센서 기반 감정 인식이 미적용된다. 오프라인 모드의 완전한 지원 부족(간단 캐시 수준) 문제가 있다. 사용자 연구 표본이 확보되어 있지 않다(실사용자 데이터 기반의 유효성 검증 필요).

3. 개선 및 확장 방향

NLP를 활용한 텍스트 감정분석 모듈 통합(예: 감정 레이블 자동 추출), 개인화 추천(사용자

패턴에 기반한 알림/조언), 임상적 연계(전문가 피드백 기능)–의료·윤리 규정 준수 필요, 오프라인 데이터 동기화 강화 및 로컬 암호화 등 다양한 방향으로 확장 및 개선하면 된다.

VII. 결론

본 연구는 모바일 기반 감정 기록 애플리케이션 개발 과정을 통해, 디지털 환경에서 개인의 감정 데이터를 수집하고 시각화하는 기술적·심리적 가능성을 탐색하였다. React Native와 Firebase를 결합함으로써 비교적 짧은 개발 기간에도 실사용 가능한 형태의 프로토타입을 구축할 수 있었으며, 특히 데이터 저장 구조와 UI 설계가 사용자 경험에 중요한 영향을 준다는 점이 확인되었다. 첫째, 채팅형 감정 기록 UI와 이모지 기반 입력 방식은 사용자의 심리적 부담을 줄이고 기록의 빈도를 높일 수 있는 요소로 작용하였다. 일반적인 설문형 UI에서는 매번 텍스트 입력이나 선택 항목이 요구되기 때문에 사용자가 기록을 회피할 가능성이 높으나, 본 연구에서 사용한 대화형 UX는 기록 행위를 “일상 대화”와 유사하게 만들어 감정 표현에 대한 진입 장벽을 낮추었다. 이는 감정 기록 자체를 보다 자연스럽고 반복 가능한 행동으로 유도하는 데 효과적이었다. 둘째, Firebase 기반 데이터 저장 방식은 사용자별 개인정보 접근 통제, 날짜 기준 데이터 구조화에 있어 적합한 방법이었다. 개인의 감정 기록은 매우 민감한 정보이므로 데이터 접근 권한 및 인증 흐름이 중요하며, 이를 Firebase에서 제공하는 인증 서비스와 실시간 데이터베이스 기능을 활용하여 비교적 안전하게 구현할 수 있었다. 특히 날짜별 그룹화 방식은 기록을 시간의 흐름 속에서 분석하는 데 적합하며, 그래프 화면 및 캘린더 화면 구성에 핵심적인 역할을 했다. 셋째, 시각화 방식은 감정 데이터를 단순 기록이 아닌 패턴 분석 결과로 전환해주는 기능을 수행한다. Pie Chart 형태의 감정 비율 분석은 사용자가 일상에서 경험한 감정을 객관화하는 데 기여하며, 특정 감정의 반복 발생, 감정 편향 여부 등의 인사이트를 제공할 수 있다. 이는 단순히 데이터를 저장하는 것에서 끝나는 것이 아니라 데이터가 사용자에게 의미 있는 정보를 전달하는 방향으로 진화할 수 있다는 가능성을 보여준다. 이러한 시각화는 개인의 자기 인식(Self-awareness)을 높이고, 필요할 경우 전문가 상담 시 객관적인 데이터 근거를 제공하는 자료가 된다. 넷째, 본 연구의 한계는 자동 감정 분석이 적용되지 않은 점, 사용자 연구가 실험되지 않은 점에 있다. 현재 시스템은 사용자의 주관적 기록에만 의존하고 있으며, 자연어 처리 기술을 이용한 감정 분석, 음성 기반 감정 추출, 행동 로그 분석 등 AI 기반 분석이 추가된다면 데이터 정확성과 객관성이 더욱 높아질 수 있다. 또한 실제 사용자 대상 실험을 수행하여 사용성(SUS), 감정 기록 빈도, 자기인식 변화 정도 등을 검증한다면 모바일 감정 기록 시스템의 심리적 효과를 보다 객관적으로 입증할 수 있을 것이다.

향후 연구에서는 여러 방향의 확장이 가능하다. 첫째, 개인별 감정 패턴에 기반한 추천 알고리즘을 적용하여 사용자에게 맞춤형 조언과 감정 관리 전략을 제시할 수 있다. 둘째, 전문가 상담 플랫폼과의 연계를 통해 디지털 치료제 형태로 발전 가능성이 있으며, 이는 정신 건강 관리의 새로운 서비스 모델을 제시할 수 있다. 셋째, 장기적으로 데이터가 축적될 경우 사용자의 감정 변화 트렌드를 분석하여 우울증, 불안장애 등 정신건강 문제의 조기 경고 신호를 감지하는 기능이 개발될 수 있다.

요약하면, 본 연구는 모바일 감정 기록 시스템의 프로토타입 구현을 통해 기술적 실행 가능성과 심리적 활용 가능성을 모두 확인하였다. 감정 데이터는 단순한 수치나 기록이 아닌 개인의 경험과 정서를 반영하는 중요한 자료이며, 이를 디지털 방식으로 저장하고 분석하는 과정

은 사용자의 자기인식과 감정 관리 능력 향상에 실질적인 도움을 줄 수 있다. 앞으로는 기술적 확장성과 심리학적 근거를 결합하여 보다 고도화된 감정 관리 플랫폼으로 발전시키는 연구가 필요할 것이다.