****

**기획 제안서**

**Team6 : ChicChic PokPok**

Member :

201421019 고보원

201521000 정소영

201520996 박준민

201520967 고광표

내용

[1. 개발 내용 5](#_Toc512193173)

[1.1 개발 목표 5](#_Toc512193174)

[1.2 Use Cases 6](#_Toc512193175)

[1.3 Features 16](#_Toc512193176)

[2. 설계 36](#_Toc512193177)

[2.1 시스템 구조 36](#_Toc512193178)

[2.1.2 SW ARCHITECTURE 38](#_Toc512193179)

[2.2 Components 39](#_Toc512193180)

[2.3 데이터 정의 40](#_Toc512193181)

[2.4 API 및 Interface 정의 43](#_Toc512193182)

[2.4.1 외부 API 43](#_Toc512193183)

[2.4.2 Restful API 43](#_Toc512193184)

[3. 수행 계획 46](#_Toc512193185)

[3.1 개발환경 46](#_Toc512193186)

[3.2 리스크 분석 47](#_Toc512193187)

[3.3 개발 일정 48](#_Toc512193188)

[3.4 비용 분석 50](#_Toc512193189)

[3.5 업무분장 계획 50](#_Toc512193190)

[3.5.1 Feature / 컴포넌트 기반 50](#_Toc512193191)

[3.5.2 기타 과제관리/Deploy/발표 등 부가적 업무 분장 51](#_Toc512193192)

[3.6 협업 방안 52](#_Toc512193193)

[3.6.1 소스코드 관리 52](#_Toc512193194)

[3.6.2 통합 및 테스트 52](#_Toc512193195)

[3.6.3 이슈 관리 53](#_Toc512193196)

[4. 출시 계획 53](#_Toc512193197)

[4.1 성과 측정 방안 53](#_Toc512193198)

[4.2 데모 시나리오 55](#_Toc512193199)

[4.2.1 데모 시나리오 1 55](#_Toc512193200)

[4.2.2 데모시나리오 2 56](#_Toc512193201)

[4.2.3. 데모시나리오 3 56](#_Toc512193202)

[4.2.4. 데모시나리오 4 57](#_Toc512193203)

[4.2.5. 데모시나리오 5 58](#_Toc512193204)

[4.2.5. 데모시나리오 6 59](#_Toc512193205)

[그림 7. Use Case Diagram 6](#_Toc512193206)

[그림 8. 초기 화면 21](#_Toc512193207)

[그림 9. Helpee 메인 페이지 21](#_Toc512193208)

[그림 10. Helpee 봉사 신청 페이지 22](#_Toc512193209)

[그림 11. Helpee가 봉사 2 신청 후 메인 페이지 23](#_Toc512193210)

[그림 12. 봉사 신청 후 근처 Helper 알림 23](#_Toc512193211)

[그림 13. Helper 메인 페이지 24](#_Toc512193212)

[그림 14. Helper 주변 봉사 검색 페이지 24](#_Toc512193213)

[그림 15. Helper 근처 봉사 조회 25](#_Toc512193214)

[그림 16. Helper 봉사 신청 26](#_Toc512193215)

[그림 17. Helper 맞춤 봉사 검색 페이지 26](#_Toc512193216)

[그림 18. Helper 맞춤 봉사 리스트 페이지 27](#_Toc512193217)

[그림 19. Helpee 봉사 수락 알림 28](#_Toc512193218)

[그림 20. Helper가 봉사 지원한 후 Helpee 메인 페이지 28](#_Toc512193219)

[그림 21. Helpee 봉사 수락 페이지 29](#_Toc512193220)

[그림 22. Helper 매칭 완료 알림 29](#_Toc512193221)

[그림 23. Helper의 매칭 완료 후 메인 페이지 30](#_Toc512193222)

[그림 24. Helper와 Helpee 봉사 시작 인증 페이지 31](#_Toc512193223)

[그림 25. Helper와 Helpee 봉사 시작 인증 후 페이지 32](#_Toc512193224)

[그림 26. Helper와 Helpee feedback 설문조사 페이지 32](#_Toc512193225)

[그림 27. 관리자 로그인 페이지 33](#_Toc512193226)

[그림 28. 관리자 유저 검색 페이지 34](#_Toc512193227)

[그림 29. 관리자 봉사 승인 페이지 34](#_Toc512193228)

[그림 30. 관리자 유저 feedback 관리 페이지 35](#_Toc512193229)

[그림 31. 시각 장애인 용 페이지 36](#_Toc512193230)

[그림 32. 전체 시스템 36](#_Toc512193231)

[그림 33. SW Architecture 38](#_Toc512193232)

[그림 34. 데모 시나리오 1 55](#_Toc512193233)

[그림 35. 데모 시나리오 2 56](#_Toc512193234)

[그림 36. 데모 시나리오 3 56](#_Toc512193235)

[그림 37. 데모 시나리오 4 57](#_Toc512193236)

[그림 38. 데모 시나리오 5 58](#_Toc512193237)

[그림 39. 데모 시나리오 6 59](#_Toc512193238)

Table 1. Use Case 1-1 6

Table 2. Use Case 1-2 7

Table 3. Use Case 1-3 8

Table 4. Use Case 2-1 9

Table 5. Use Case 2-2 10

Table 6. Use Case 3-1 11

Table 7. Use Case 3-2 11

Table 8. Use Case 3-3 12

Table 9. Use Case 3-4 12

Table 10. Use Case 4-1 14

Table 11. Use Case 4-2 15

Table 12. 모든 사용자 관점 Feature 16

Table 13. “Helper” 사용자 관점 Feature 17

Table 14. “Helpee” 사용자 관점 Feature 18

Table 15. 관리자 관점 Feature 20

Table 16. Components 39

Table 17. User schema 40

Table 18. Volunteer Schema 41

Table 19. Feedback Schema 42

Table 20. 외부 API 43

Table 21. Restful API 43

Table 22. 개발환경 46

Table 23. Framework 46

Table 24. Risk 47

Table 25. 개발 일정 48

Table 26. 비용 분석 50

Table 27. Feature / 컴포넌트 기반 업무분장 50

Table 28. 부가적 업무 분장 51

Table 29. 소스코드 관리 52

Table 30. 통합 및 테스트 52

Table 31. 이슈 관리 53

Table 32. 성과 측정 방안 53

Table 33. Demo Scenario 1 55

Table 34. Demo Scenario 2 56

Table 35. Demo Scenario 3 56

Table 36. Demo Scenario 4 57

Table 37. Demo Scenario 5 58

Table 38. Demo Scenario 6 59

# 1. 개발 내용

## 1.1 개발 목표

본 소프트웨어는 다음과 같은 개발 목표를 가진다.

**[위치 기반의 근거리 지원자 매칭]**

봉사 자원, 도움 요청 시 사용자의 현 위치를 gps로 받아와 가까운 위치에서 봉사를 자원한 사람, 도움을 요청한 사람을 볼 수 있게 해준다. 이는 GOOGLE에서 제공하는 Google Maps API를 이용한다.

**[데이터베이스 기반의 상대방 맞춤 추천]**

위치 기반 매칭과 마찬가지로 상대방을 매칭시켜줄 때 원하는 조건을 선택하면 이를 데이터베이스에서 필터링하여 요구조건과 가장 유사한 상대방을 추천해 준다. MongoDB로 설계된 NoSQL형태의 데이터베이스를 이용한다.

**[푸시 알림을 이용한 실시간 통신]**

“HELPEE”들은 “HELPER”들이 봉사를 자원하여도 그들의 정보를 확인하고 자신이 요구한 조건에 부합되지 않을 시 거절할 수 있는데, 이를 실시간으로 처리하여 “HELPER”들도 다른 봉사를 알아보고 “HELPEE”들도 도움을 즉각적으로 받을 수 있도록 한다. 이는 FCM(Firebase Cloud Messaging)을 이용한 앱 푸시 기능을 통해 실현한다.

**[중앙서버 제어를 통한 봉사시간 통제]**

봉사시간을 “HELPER”들에게 부여할 때 객관적인 기준에 따라 봉사시간을 부여해야 하는 것이 원칙이다. 이를 위해 시간을 정확히 측정해야 하므로 “HELPEE”와 “HELPER”가 만난 시간을 랜덤하게 생성된 인증번호를 서로 확인함으로써 봉사 시작 시간이 서버에 저장되고, 종료도 마찬가지의 방법으로 이루어짐으로써 서버시간을 통해 최종적인 봉사 시간을 부여할 수 있다.

## 1.2 Use Cases

☐ Use Case Diagram

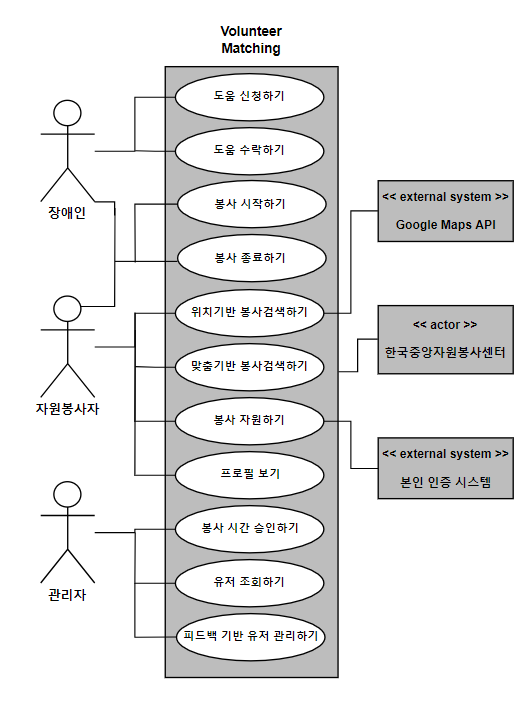


그림 7. Use Case Diagram

☐ 1. 관리자 시나리오

[Use Case 1-1: 봉사 시간 승인하기]

Table 1. Use Case 1-1

|  |
| --- |
| Text description: 관리자는 봉사 시간을 승인할지 결정한다.  Precondition: 관리자는 로그인되어 있다.  Primary actor: 관리자  Success Main Scenario:   1. 관리자는 봉사 시간 승인 대기 리스트를 본다. 2. 관리자는 리스트의 각 봉사 피드백을 읽는다. 3. 관리자는 봉사 시간을 승인 할지 결정한다.   Extensions:  1a. 관리자가 봉사 시간 승인 대기 리스트를 다시 받고 싶을 경우  1. 관리자는 “새로고침” 버튼을 누른다.  2. System은 관리자에게 최신의 봉사 시간 승인 대기 리스트를 보여준다.  2a. 리스트에서 한 봉사에 대하여 더 자세한 정보를 보고 싶은 경우  1. 관리자는 “봉사 상세 정보” 버튼을 누른다.  2. System은 그 봉사의 상세 정보를 팝업창으로 띄워 준다.  3a. “봉사 시간 승인” 버튼을 눌렀을 경우  1. System은 “봉사 승인 되었습니다” 알림창을 띄운다.  2. System은 유저의 봉사 시간을 업데이트한다.  3.b. “봉사 시간 거부” 버튼을 눌렀을 경우  1. System은 유저에게 “봉사 시간이 승인되지 않았습니다” 알림을 보낸다. |

[Use Case 1-2: 유저 조회하기]

Table 2. Use Case 1-2

|  |
| --- |
| Text description: 관리자는 유저를 조회한다.  Precondition: 관리자는 로그인되어 있다.  Primary actor: 관리자  Success Main Scenario:   1. 관리자는 유저 리스트를 본다. 2. 관리자는 유저 리스트에서 관심 유저를 조회한다.   Extensions:  1a. 관리자가 봉사 시간 승인 대기 리스트를 다시 받고 싶을 경우  1. 관리자는 “새로고침” 버튼을 누른다.  2. System은 관리자에게 최신의 봉사 시간 승인 대기 리스트를 보여준다.  2.a. 관리자가 유저의 상세 정보를 조회하고 싶은 경우  1. 관리자는 “유저 상세 정보” 버튼을 누른다.  2. System은 그 유저의 상세 정보를 팝업창으로 띄워 준다.  2.b. 관리자가 관심 유저를 찾고 싶은 경우  1. 관리자는 “유저 검색” 버튼을 누른다.  2. 관리자는 유저의 정보를 입력한다.  3. System은 관심 유저를 보여준다. |

[Use Case 1-3: 피드백 기반 유저 관리하기]

Table 3. Use Case 1-3

|  |
| --- |
| Text description: 관리자는 유저들의 피드백을 보고 유저를 관리한다.  Precondition: 관리자는 로그인되어 있다.  Primary actor: 관리자  Success Main Scenario:   1. 시스템은 유저들의 피드백 리스트를 출력한다. 2. 관리자는 유저들의 피드백 리스트를 보고 유저들을 관리한다.   Extensions:  1a. 관리자가 봉사 시간 승인 대기 리스트를 다시 받고 싶을 경우  1. 관리자는 “새로고침” 버튼을 누른다.  2. System은 관리자에게 최신의 유저들의 피드백 리스트를 보여준다.  1.b 관리자가 특정 유저를 조회하고 싶은 경우   1. 관리자는 “검색” 버튼을 누른다. 2. 관리자는 특정 유저의 정보를 입력한다. 3. System은 그 정보에 맞는 유저를 찾아 유저를 출력한다.   2.a. 관리자가 유저의 상세 정보를 조회하고 싶은 경우  1. 관리자는 “유저 피드백 상세 정보” 버튼을 누른다.  2. System은 그 유저의 피드백 상세 정보를 팝업창으로 띄워 준다.  2.b. 관리자가 피드백이 안좋은 유저를 중지시키고 싶은 경우  1. 관리자는 리스트의 유저에 “서비스 중지” 버튼을 누른다.  2. System은 그 유저를 제거한다. |

☐ 2. 장애인 시나리오

[Use Case 2-1: 도움신청하기]

Table 4. Use Case 2-1

|  |
| --- |
| Text description: 장애인은 봉사를 등록한다.  Precondition: 장애인은 로그인되어 있다.  Primary actor: 장애인 및 노약자  Success Main Scenario:   1. 장애인은 “도움 받으실 건가요?”와 “도움 주실 건가요?” 버튼 중 “도움 받으실 건가요?” 버튼을 누른다. 2. 장애인은 “도움 신청” 버튼을 누른다. 3. System은 장애인에게 “위치, 도움의 종류, 예상 소요 시간, 기타 요청사항”을 요구한다. 4. 장애인은 “위치, 도움의 종류, 예상 소요 시간, 기타 요청사항”을 기재하고 “확인”버튼을 누른다. 5. System은 주변에 있는 자원봉사자에게 새로운 봉사가 등록되었다는 알림을 보내고 봉사 리스트에 해당 봉사를 추가한다. |

[Use Case 2-2: 도움수락하기]

Table 5. Use Case 2-2

|  |
| --- |
| Text description: 장애인은 자원봉사자가 자원한 봉사를 수락한다.  Precondition: 장애인이 등록한 봉사에 자원봉사자로부터 봉사자원이 들어와 장애인에게 알림이 와있다.  Primary actor: 장애인 및 노약자  Success Main Scenario:   1. 장애인은 봉사에 자원한 자원봉사자의 정보를 확인하고 “승인” 버튼을 누른다. 2. System은 자원봉사자에게 도움 수락 여부를 알리고 장애인과 자원봉사자의 현재 상태 정보와 봉사 리스트를 업데이트한다.   Extensions:  1a. 장애인이 “거부” 버튼을 누르는 경우  1 System은 자원봉사자에게 거부 알림을 보낸다. |

☐ 3. 자원봉사자 시나리오

[Use Case 3-1: 위치기반 봉사검색하기]

Table 6. Use Case 3-1

|  |
| --- |
| Text description: 자원봉사자는 자신의 위치를 기반으로 봉사를 검색한다.  Precondition: 자원봉사자는 로그인되어 있다.  Primary actor: 자원봉사자  Success Main Scenario:   1. 자원봉사자는 “도움 받으실 건가요?”와 “도움 주실 건가요?” 버튼 중 “도움 주실 건가요?” 버튼을 누른다. 2. System은 자원봉사자에게 “매칭 항목”을 보여준다. 3. 자원봉사자는 “위치검색” 버튼을 누른다. 4. System은 자원봉사자에게 “현재 위치”와 이에 기반한 주변 “봉사 요청 리스트”를 지도상에 표시해준다. 5. 자원봉사자는 지도상 관심 있는 위치에 자리잡은 버튼을 누른다. 6. System은 해당 위치에서 요청한 봉사의 세부사항을 보여준다. |

[Use Case 3-2: 조건기반 봉사검색하기]

Table 7. Use Case 3-2

|  |
| --- |
| Text description: 자원봉사자는 원하는 내용의 봉사를 검색한다.  Precondition: 자원봉사자는 로그인되어 있다.  Primary actor: 자원봉사자  Success Main Scenario:   1. 자원봉사자는 “도움 받으실 건가요?”와 “도움 주실 건가요?” 버튼 중 “도움 주실 건가요?” 버튼을 누른다. 2. System은 자원봉사자에게 “매칭 항목”을 보여준다. 3. 자원봉사자는 “맞춤검색” 버튼을 누른다. 4. System은 자원봉사자에게 “위치, 봉사 종류, 봉사 대상, 시간” 의 입력을 요구한다. 5. 자원봉사자는 “위치, 봉사 종류, 봉사 대상, 시간”을 선택하고 “검색” 버튼을 누른다. 6. System은 자원봉사자가 입력한 내용에 알맞은 “봉사 요청 리스트”를 보여준다. 7. 자원봉사자는 System이 보여준 “봉사 요청 리스트” 중 관심있는 봉사를 누른다. 8. System은 해당 봉사의 세부사항을 보여준다. |

[Use Case 3-3: 봉사 자원 하기]

Table 8. Use Case 3-3

|  |
| --- |
| Text description: 자원봉사자는 관심 있는 봉사를 자원 한다.  Precondition: 자원봉사자는 로그인 후 “봉사 요청 리스트”를 확인하였다.  Primary actor: 자원봉사자  Success Main Scenario:   1. 자원봉사자는 관심 있는 봉사를 누른다. 2. System은 해당 봉사의 세부사항을 보여준다. 3. 자원봉사자는 “봉사 지원” 버튼을 누른다. 4. System은 자원봉사자에게 “봉사 유의 사항”을 보여준다. 5. 자원봉사자는 “유의 사항 읽음” 버튼을 누른다. 6. System은 장애인에게 자원봉사자의 정보를 전달한다. 7. 장애인은 System에게 자원봉사자의 신상정보에 대한 알림을 받는다. |

[Use Case 3-4: 프로필 보기]

Table 9. Use Case 3-4

|  |
| --- |
| Text description: 자원봉사자는 프로필을 본다.  Precondition: 자원봉사자는 로그인되어 있다.  Primary actor: 자원봉사자  Success Main Scenario:   1. 자원봉사자는 “도움 받으실 건가요?”와 “도움 주실 건가요?” 버튼 중 “도움 주실 건가요?” 버튼을 누른다. 2. System은 자원봉사자에게 “매칭 항목”을 보여준다. 3. 자원봉사자는 “프로필” 버튼을 누른다.   Extensions:  3a. 자원봉사자가 회원정보수정을 원하는 경우   1. 자원봉사자는 “회원 정보 수정” 버튼을 누른다. 2. System은 자원봉사자에게 “회원 정보”를 보여준다. 3. 자원봉사자는 수정을 원하는 항목의 입력값을 수정한다. 4. 자원봉사자는 “수정 완료” 버튼을 누른다. 5. System은 수정된 정보를 해당 회원의 정보로 갱신한다.   3b. 자원봉사자가 자신이 자원한 봉사의 매칭상태를 보고 싶은 경우   1. 자원봉사자는 “내 신청 상황” 버튼을 누른다. 2. System은 자원봉사자에게 자원 신청 처리가 완료된 봉사들의 리스트를 보여준다.   2a. 자원봉사자가 자원한 봉사를 취소하기를 원하는 경우   1. 자원봉사자는 보여진 리스트 중 취소를 원하는 봉사의 영역을 누른다. 2. System은 자원봉사자에게 해당 봉사의 매칭 상태를 보여준다. 3. 자원봉사자는 “자원 취소” 버튼을 누른다. 4. System은 해당 자원봉사자의 자원 상태를 되돌리고 해당 봉사를 요청한 장애인에게 “취소” 알림을 보낸다.   3c. 자원봉사자가 자신이 완료한 봉사내역을 보고 싶은 경우   1. 자원봉사자는 “봉사 내역” 버튼을 누른다. 2. System은 자원봉사자에게 해당 자원봉사자가 완료한 봉사들의 리스트를 보여준다.   3d. 자원봉사자가 자신의 봉사가 제대로 인증되었는지 확인하고 싶은 경우   1. 자원봉사자는 “봉사 포털 바로가기” 버튼을 누른다. 2. System은 자원봉사자에게 “1365 자원봉사포털 사이트”를 연결해준다. |

[Use Case 4-1: 봉사 시작하기]

Table 10. Use Case 4-1

|  |
| --- |
| Text description: 자원봉사자, 장애인 및 노약자는 봉사를 시작한다.  Precondition: 자원봉사자는 봉사를 자원하였고 장애인은 봉사를 수락하여 매칭이 완료된 상태이며 각자 로그인되어 있고 메인화면에 들어와있다.  Primary actor: 자원봉사자, 장애인 및 노약자  Success Main Scenario:   1. 자원봉사자와 장애인은 “매칭 항목”에 나타난 봉사의 “시작” 버튼을 누른다. 2. System은 자원봉사자와 장애인에게 랜덤으로 생성한 4자리 숫자를 인증번호로 보여준다. 3. 자원봉사자와 장애인은 System이 상대방에게 생성한 인증번호를 입력하고 “입력완료” 버튼을 누른다. 4. System은 자원봉사자와 장애인에게 봉사 시작을 알리고 두 명 모두 인증이 완료된 시간을 봉사 시작 시간을 저장한다.   Extensions:  3a. 인증번호의 입력이 상대방의 인증번호와 일치하지 않는 경우   1. System은 “인증 실패 메세지”를 띄워주고 다시 입력을 기다린다. 2. 자원봉사자와 장애인은 System이 상대방에게 생성한 인증번호를 다시 입력하고 “입력완료” 버튼을 누른다. |

[Use Case 4-2: 봉사 종료하기]

Table 11. Use Case 4-2

|  |
| --- |
| Text description: 자원봉사자, 장애인 및 노약자는 봉사를 종료한다.  Precondition: 자원봉사자와 장애인이 서로 매칭되어 봉사 시작을 한 상태이며 각각 로그인이 되어 메인화면에 들어와있다.  Primary actor: 자원봉사자, 장애인 및 노약자  Success Main Scenario:   1. 자원봉사자와 장애인을 “매칭 항목”에 나타난 봉사의 “종료” 버튼을 누른다. 2. System은 자원봉사자와 장애인에게 랜덤으로 생성한 4자리 숫자를 인증번호로 보여준다. 3. 자원봉사자와 장애인은 System이 상대방에게 생성한 인증번호를 입력하고 “입력완료” 버튼을 누른다. 4. System은 자원봉사자와 장애인에게 각각 설문조사를 요청한다. 5. 자원봉사자와 장애인은 설문조사를 작성하고 “제출”버튼을 누른다. 6. System은 자원봉사자에게 시작 시간과 종료 시간을 바탕으로 봉사 시간을 부여하고 설문조사 정보를 관리자에게 전달한다.   Extensions:  3a. 인증번호의 입력이 상대방의 인증번호와 일치하지 않는 경우   1. System은 “인증 실패 메세지”를 띄워주고 다시 입력을 기다린다. 2. 자원봉사자와 장애인은 System이 상대방에게 생성한 인증번호를 다시 입력하고 “입력완료” 버튼을 누른다. |

## 1.3 Features

[모든 사용자 관점 Feature]

Table 12. 모든 사용자 관점 Feature

|  |  |
| --- | --- |
| **Features** | **Description** |
| 회원가입 하기 | 사용자는 Volunteer Matching 서비스를 이용하기 위해 아이디와 비밀번호를 입력하고 사용자의 이름, 생년월일, 사용자의 사진, 성별을 입력한다. 이때 “HELPEE”가 회원가입 하는 경우 장애인 등록증을 찍어 업로드해야한다. 또한 범죄를 예방하기 위해 회원가입시 허위사실을 기입하는 경우 형사처벌을 받을 수 있다는 서약에 동의해야한다. 사용자가 입력한 데이터는 아이디를 통해 구분되며 데이터베이스에 저장된다. |
| 본인 인증 하기 | 회원가입시 기재한 이름, 성별, 생년월일이 허위사실이 아님을 인증하기 위해 휴대폰, 주민번호를 통한 본인 인증 후 회원가입이 승인된다. |
| “HELPER”/”HELPEE”선택하기 | 로그인 후 사용자는 “HELPER”인지 “HELPEE”인가를 선택한다. 선택에 따라 “HELPER”화면과 “HELPEE”화면으로 넘어간다. |
| 봉사 시작 하기 | “HELPER”와 “HELPEE”는 약속한 시간에 약속 장소에서 만나 봉사를 시작하기 앞서 각자의 앱의 현재 상태 화면에 있는 봉사 리스트에서 해당 봉사의 “봉사 시작하기” 버튼을 누른다. 버튼이 눌리면 서버에서 랜덤으로 인증용 4자리 숫자를 각 사용자의 화면에 출력하고 “HELPER”와 “HELPEE”는 서로의 번호를 교환하여 앱에 입력하면 서버로 해당 번호가 전송된다. 서버에서 번호 인증이 완료되면 인증완료 시간이 봉사 시작 시간으로 데이터베이스에 저장된다. 사용자의 화면에는 “봉사 중”을 출력한다. |
| 봉사 종료 하기 | “HELPER”와 “HELPEE”는 봉사가 끝나면 앱의 현재 상태 화면에 있는 봉사 리스트에서 해당 봉사의 “봉사 종료” 버튼을 누른다. 버튼이 눌리면 서버에서 데이터베이스에 저장된 봉사 시작시간을 바탕으로 봉사 시간을 계산한다. 계산된 봉사 시간은 봉사 정보, “HELPER” 정보, “HELPEE” 정보와 함께 관리자에게 전송되어 승인을 기다린다. |

[“HELPER” 사용자 관점 Feature]

Table 13. “Helper” 사용자 관점 Feature

|  |  |
| --- | --- |
| **Features** | **Description** |
| 현재 상태 화면 출력 | “HELPER”의 현재 상태를 표시하는 화면이 출력된다. “HELPER”가 신청한 봉사 리스트가 출력되고 각각의 봉사에는 봉사의 제목, 봉사의 진행 상태, “상세 정보 보기” 버튼이 출력된다. 봉사의 진행 상태가 “매칭 대기중”인 경우 “봉사 시작” 버튼이 비활성화 되어 있고 “매칭 완료”인 경우 시작 버튼이 활성화되어 봉사활동을 시작할 수 있게 된다. “상세 보기 버튼”을 누르는 경우 해당 봉사활동의 상세 내용을 출력한다. |
| 위치 검색 하기 | “HELPER”가 위치 검색에서 “나의 위치”버튼을 누르면 현재 “HELPER”가 위치한 위치를 기반으로 주변에 등록된 도움 요청이 출력된다. |
| 맞춤 검색 하기 | “HELPER”가 맞춤 검색에서 위치, 봉사의 종류, 봉사 대상, 시간 중 원하는 항목을 선택하면 데이터 베이스에 저장된 봉사 리스트 중 필터링 하여 해당 봉사들을 화면에 출력한다. |
| 봉사 자원 하기 및 서약 동의 하기 | “HELPER”는 위치 검색이나 맞춤 검색을 통해 원하는 봉사를 선택하고 “봉사 자원 하기”버튼을 눌러서 봉사에 자원 할 수 있다. 이때 “봉사 자원 하기”버튼을 누르게 되면 봉사 주의 사항 및 서약서 화면이 출력되고 “HELPER”가 “동의” 버튼을 누르게 되면 봉사 자원이 완료된다. 자원이 완료되면 해당 봉사의 “HELPEE”에게 봉사에 자원한 사람이 있다는 알림이 전송된다. |
| 근처 봉사 실시간 알림 받기 | “HELPER”는 푸쉬 알림을 통해 주변에 실시간으로 등록되는 봉사 활동에 대해 알림을 서버로 부터 받을 수 있다. |
| 봉사 매칭 완료 알림 받기 | “HELPER”는 자신이 지원한 봉사에 대해서 “HELPEE”에게 봉사 자원이 승낙되면 봉사 매칭이 완료되었다는 알림을 서버로 부터 받을 수 있다. |
| 봉사 내역 확인하기 | “HELPER”는 봉사 내역 확인 페이지에서 자신이 이제까지 수행한 봉사에 대한 내역과 승인된 봉사 시간을 확인할 수 있다. |
| 설문조사 하기 | 봉사가 종료되면 “HELPER”는 “HELPEE”에 대한 설문조사를 요청받는다. 설문 조사를 완료하면 해당 설문 조사 데이터는 데이터베이스에 저장되고 관리자에 의해 피드백에 반영된다. |

[“HELPEE” 사용자 관점 Feature]

Table 14. “Helpee” 사용자 관점 Feature

|  |  |
| --- | --- |
| **Features** | **Description** |
| 현재 상태 화면 출력 | “HELPEE”의 현재 상태를 표시하는 화면이 출력된다. “HELPEE”가 도움을 요청한 봉사 리스트가 출력되고 각각의 봉사에는 봉사의 제목, 봉사의 진행 상태, “지원자 정보 보기” 버튼이 출력된다. 봉사의 진행 상태가 “매칭 대기중”인 경우 “봉사 시작” 버튼이 비활성화 되어 있고 “매칭 완료”인 경우 “봉사 시작” 버튼이 활성화되어 봉사활동을 시작할 수 있게 된다. 진행 상태가 “매칭 대기중”인 경우 “지원자 정보 보기” 버튼을 누르면 해당 봉사에 자원한 봉사자들의 리스트가 출력된다. “HELPEE”는 해당 지원자들 중에서 선택할 수 있다. 선택된 자원봉사자에게는 매칭이 완료되었다는 알림이 전송된다. 그 외의 봉사자들에게는 매칭에 실패되었다는 알림이 전송된다.. 선택이 완료되면 “지원자 보기 버튼”을 누르는 경우 자신이 지정한 자원봉사자의 정보만 출력된다. |
| 도움 요청하기 | “HELPEE”는 도움이 필요할때 “신청하기” 버튼을 눌러서 위치, 봉사 종류, 요구 시간, 기타 사항을 입력하고 “신청”버튼을 눌러 도움을 요청할 수 있다. |
| 자원봉사자 자원 알림 받기 | “HELPEE”는 자신이 등록한 봉사에 대해 자원봉사자의 자원이 들어오면 서버로부터 알림을 받을 수 있다. 알림에는 자원한 봉사자의 정보가 출력된다. |
| 자원봉사자 봉사 자원 승낙하기 | “HELPEE”는 자신이 등록한 봉사에 자원한 자원봉사자들의 리스트 중 한 사람을 선택하여 봉사 자원을 승낙할 수 있다. 선택된 자원봉사자에게는 봉사 매칭이 완료되었다는 알림이 서버로부터 전송된다. |
| 설문조사 하기 | 봉사가 종료되면 “HELPEE”는 “HELPER”에 대한 설문조사를 요청받는다. 설문 조사를 완료하면 해당 설문 조사 데이터는 데이터베이스에 저장되고 관리자에 의해 피드백에 반영된다. |
| 콜센터 연결하기 | 시각 장애인의 경우 앱의 초기 페이지에서 두번 연속 터치를 하면 콜센터에 연결하여 직접 앱을 조작하지 않고도 콜센터 직원에게 원하는 도움의 종류, 위치, 요구 시간을 전달하면 도움 요청이 등록된다. 콜센터 직원은 시각 장애인의 요구에 맞게 도움 요청을 등록한다. |

[관리자 관점 Feature]

Table 15. 관리자 관점 Feature

|  |  |
| --- | --- |
| **Features** | **Description** |
| 관리자 페이지 암호 입력하기 | 관리자 페이지 초기 화면에 관리자 인증 암호 입력 창이 출력된다. 관리자는 해당 페이지에 암호를 입력하여 관리자 기능에 접근할 수 있다. |
| 사용자 조회하기 | 관리자 기능에 접근하면 사용자 정보를 조회할 수 있는 페이지가 나온다. 해당 페이지에 접근하면 데이터 베이스에서 사용자 정보를 받아와 화면에 출력한다. 관리자는 원하는 사용자를 선택하여 사용자의 정보, 봉사활동 내역, 피드백 정보를 확인할 수 있다. 또한 사용자의 이름 검색을 통해 원하는 사용자를 조회할 수 있다. |
| 피드백을 바탕으로 사용자 관리하기 | 관리자가 피드백 관리 페이지에 접근하면 데이터 베이스에서 완료된 봉사활동에 대해 “HELPER”와 “HELPEE”의 피드백들을 가져와 화면에 출력한다. 관리자는 해당 페이지에서 각각의 피드백을 확인하여 평가가 좋지 않은 사용자에 대하여 사용 중지 처리를 할 수 있다. |
| 봉사 시간 승인하기 | 관리자가 봉사 시간 승인 페이지에 접근하면 데이터베이스로부터 “봉사 시간 승인 대기 리스트”를 받아와 화면에 출력한다. 관리자는 해당 페이지에서 종료된 봉사에 대해 봉사 시간을 승인할 수 있다. 승인 정보는 데이터베이스에 저장된다. |

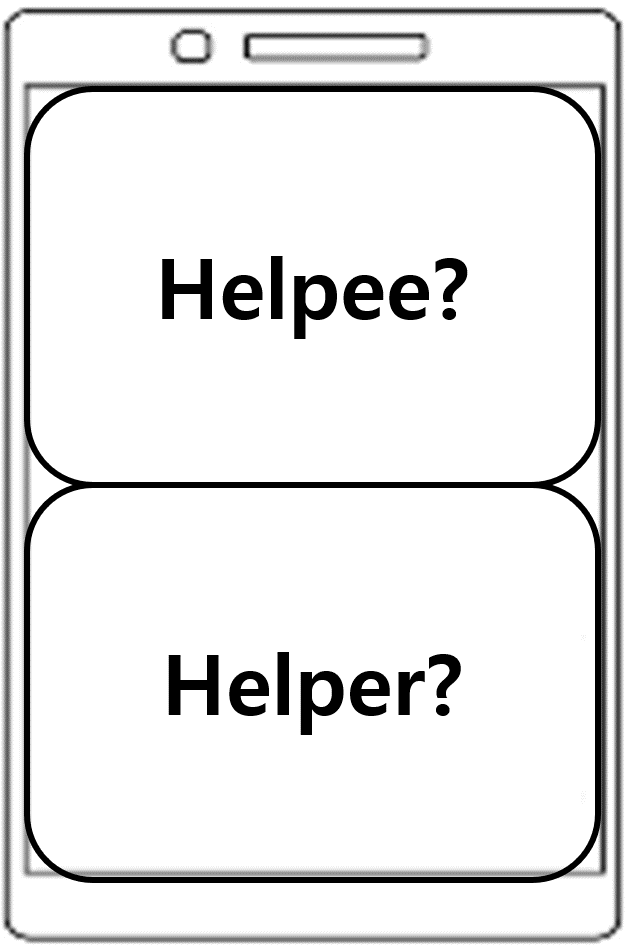


그림 8. 초기 화면

그림 8은 본 어플리케이션의 초기 화면, “HELPER”/”HELPEE”선택하기 Feature이다. 유저는 Helpee나 Helper를 선택하여 본 서비스를 사용할 수 있다. Helpee는 도움을 받고자 하는 유저이며 Helper는 자원봉사자로 사용하고자 하는 유저이다.



그림 9. Helpee 메인 페이지

유저가 Helpee로 선택했을 때 그림9와 같은 페이지로 넘어간다. 그림 9는 Helpee의 메인 페이지인 현재 상태 화면 출력 Feature이다. Helpee의 메인 페이지에는 “봉사 신청하기” 버튼과 현재 신청한 봉사 리스트를 보여준다. 봉사 리스트에는 봉사의 정보, 봉사 매칭 여부와 시작 버튼이 있다. Helpee가 새로운 봉사를 신청하고 싶으면 “봉사 신청하기” 버튼을 누른다.



그림 10. Helpee 봉사 신청 페이지

그림 10은 그림 9의 페이지에서 “봉사 신청하기” 버튼을 눌렀을 때 봉사의 정보를 적는 페이지인 도움 요청하기 Feature이다. 봉사의 위치, 시간, 봉사의 종류 등을 적고 봉사를 신청할 수 있다. “도움 신청” 버튼을 누르면 이 정보의 봉사를 Helper가 조회, 신청할 수 있다.

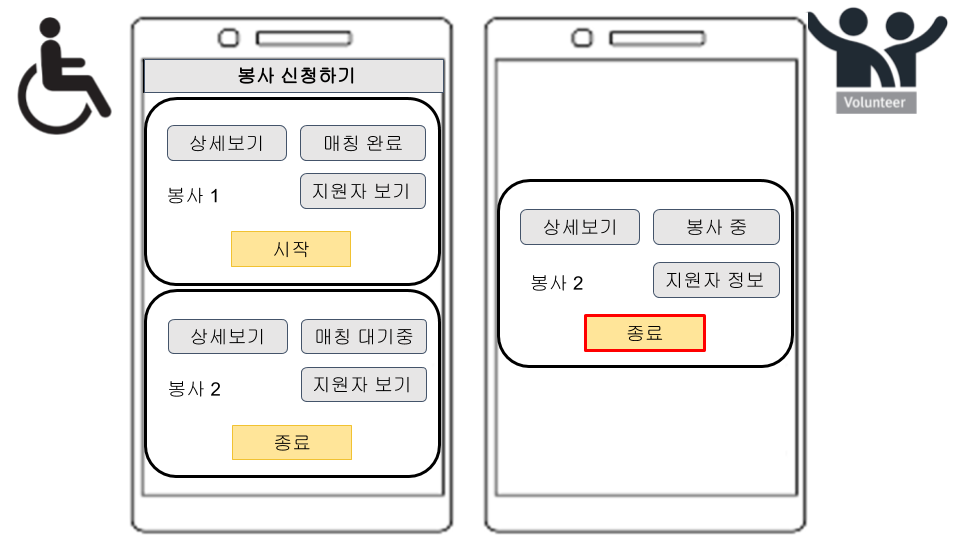


그림 11. Helpee가 봉사 2 신청 후 메인 페이지

Helpee가 “도움 신청” 버튼을 눌렀을 때 봉사 리스트에 새로운 봉사가 “매칭 대기중”이라는 상태로 생성된 것을 그림 11에서 볼 수 있다.



그림 12. 봉사 신청 후 근처 Helper 알림

그림 12은 근처 봉사 실시간 알림 받기 Feature이다. 근처의 Helpee가 신청한 봉사에 대해 Helper는 알림을 받을 수 있다.

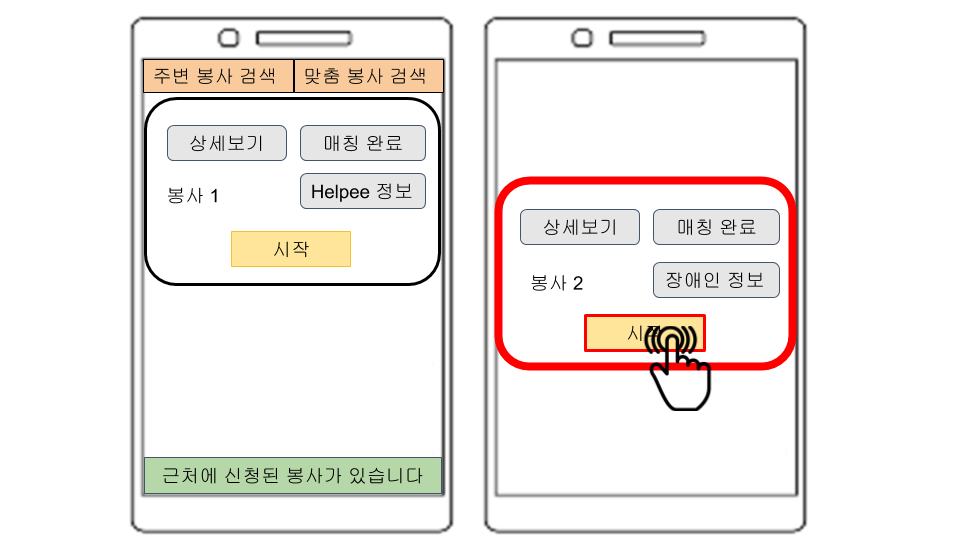


그림 13. Helper 메인 페이지

그림 13은 그림 8의 “HELPER”/”HELPEE”선택하기 Feature에서 Helper로 선택했을 때 넘어가는 Helper의 현재 상태 화면 출력 Feature이다. “주변 봉사 검색” 과 “맞춤 봉사 검색” 버튼이 있으며 현재 자원한 봉사의 리스트와 근처 봉사를 조회할 수 있는 버튼이 있다.

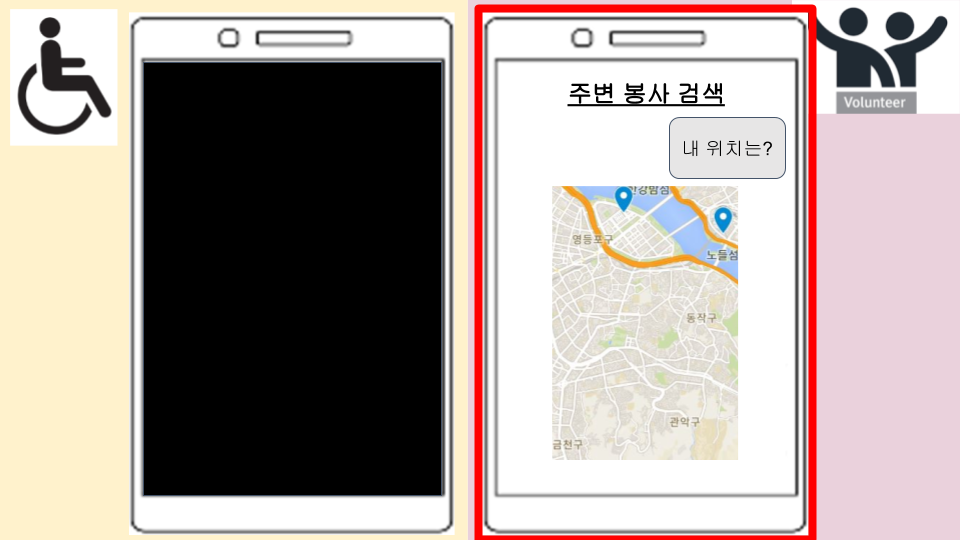


그림 14. Helper 주변 봉사 검색 페이지

그림 14은 Helper의 현재 상태 화면 출력 Feature에서 “주변 봉사 검색” 버튼을 눌렀을 때 나오는 위치 검색 하기 Feature이다. “내 위치는?” 버튼을 누르면 자신의 위치 근처의 봉사들을 지도에서 조회할 수 있다.

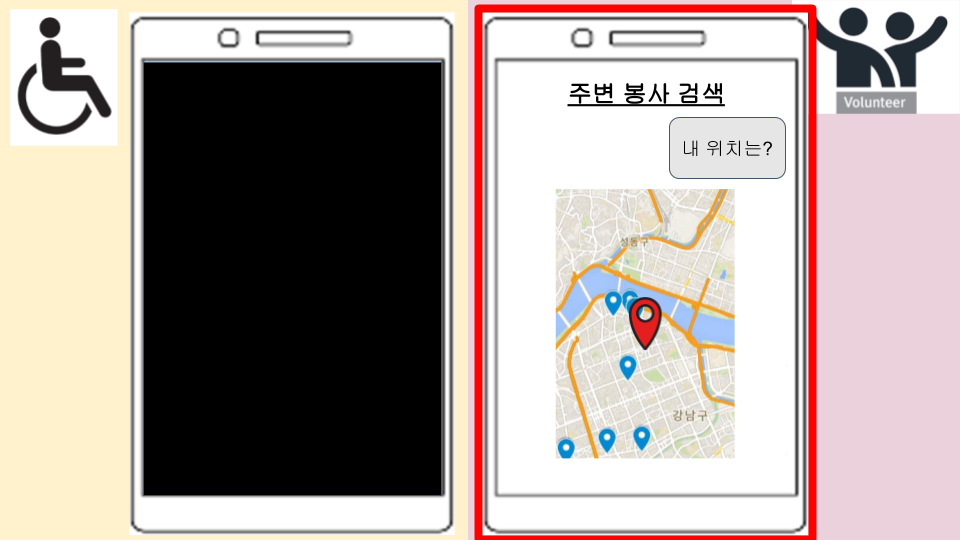


그림 15. Helper 근처 봉사 조회

위치 검색 하기 Feature에서 “내 위치는” 버튼을 눌렀을 때 그림 15와 같이 자신의 위치 근처의 봉사들을 지도에서 보여준다.



그림 16. Helper 봉사 신청

그림 16은 Helper가 지도에서 자신이 원하는 봉사를 눌러을 때 나오는 봉사 자원 하기 및 서약 동의 하기 Feature이다. “신청하기” 버튼을 누르면 봉사 주의 사항 및 서약서 화면이 출력되고 Helpee에게 수락 알림이 간다.

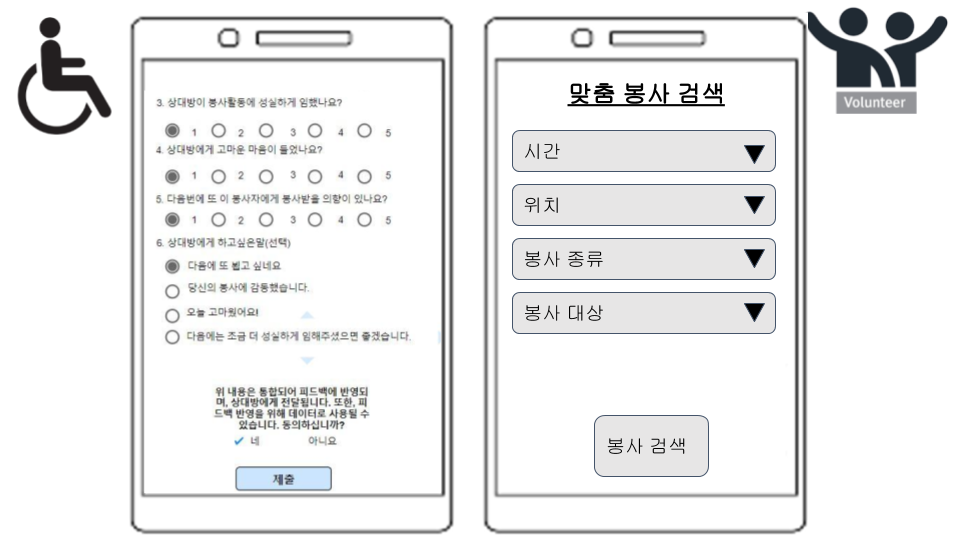


그림 17. Helper 맞춤 봉사 검색 페이지

그림 17은 맞춤 검색 하기 Feature이다. 이 페이지에서는 Helper가 자신이 원하는 시간, 위치, 봉사 종류, 대상 별로 봉사를 검색할 수 있다. “봉사 검색” 버튼을 누르면 자신이 선택한 특징의 봉사 리스트를 조회할 수 있다.

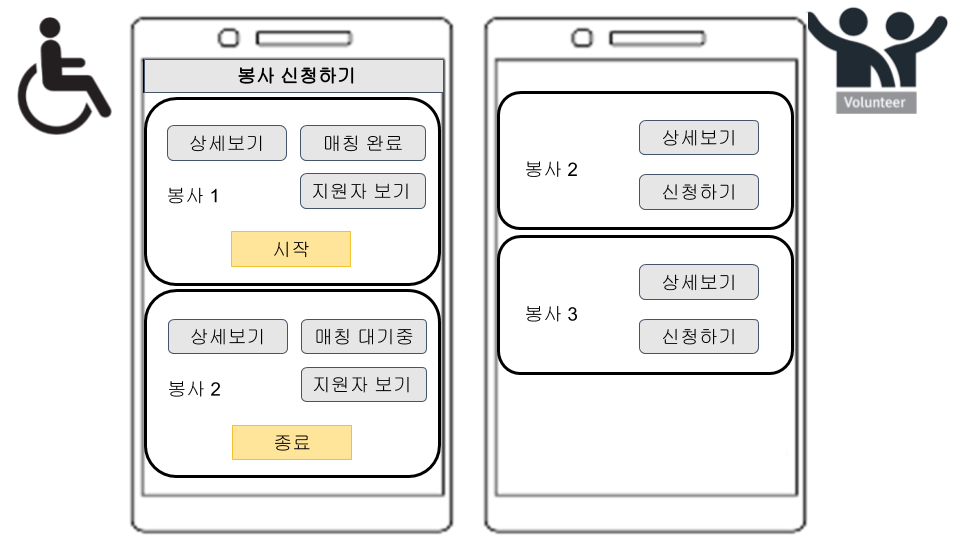


그림 18. Helper 맞춤 봉사 리스트 페이지

그림 18의 맞춤 검색 하기 Feature에서 시간, 위치 등을 입력한 후 “봉사 검색” 버튼을 누르면 그에 맞는 봉사들의 리스트를 볼 수 있으며 Helper는 자신이 원하는 봉사를 “신청하기” 버튼으로 자원 할 수 있다.



그림 19. Helpee 봉사 수락 알림

Helper가 맞춤 검색하기 Feature 또는 위치 검색하기 Feature로 봉사를 자원하면 그림 19처럼 자원봉사자 자원 알림 받기 Feature에 해당하는 Helpee에게 봉사 수락 알림이 간다.

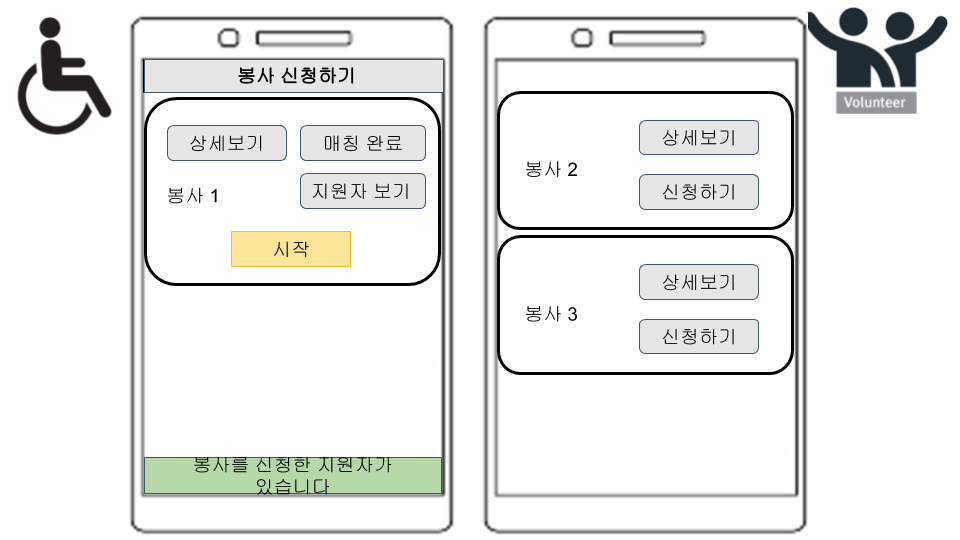


그림 20. Helper가 봉사 지원한 후 Helpee 메인 페이지

Helper가 Helpee가 등록한 봉사를 자원하면 Helpee의 메인 페이지에는 그림 20과 같이 “봉사를 신청한 지원자가 있습니다”라는 알림이 생긴다.



그림 21. Helpee 봉사 수락 페이지

그림 21은 Helpee가 Helper의 봉사 자원을 수락하는 자원봉사자 봉사 자원 승낙하기 Feature이다. HELPEE는 자신이 등록한 봉사에 자원한 자원봉사자들의 리스트 중 한 사람을 선택하여 봉사 자원을 승낙할 수 있다.



그림 22. Helper 매칭 완료 알림

자원봉사자 봉사 자원 승락하기 Feature에서 Helpee가 Helper의 봉사 자원을 수락하면 Helper에게 봉사 매칭이 완료되었다는 알림이 가는 봉사 매칭 완료 알림 받기 Feature가 그림 22과 같이 수행된다.

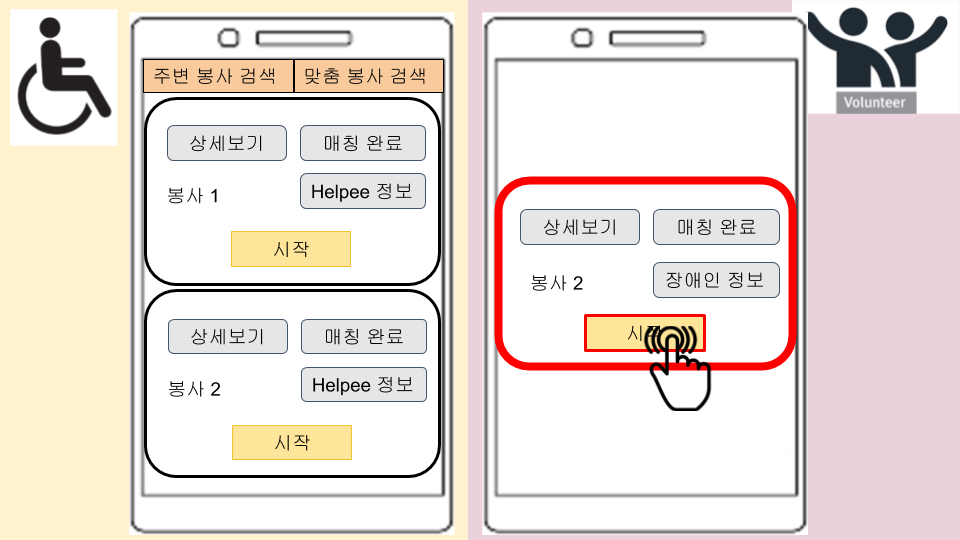


그림 23. Helper의 매칭 완료 후 메인 페이지

Helpee가 Helper의 봉사 자원을 수락하면 그림 23와 같이 매칭 정보가 “매칭 대기 중”에서 “매칭완료”로 변경된다.



그림 24. Helper와 Helpee 봉사 시작 인증 페이지

Helper와 Helpee가 약속한 시간에 약속 장소에서 만나 봉사를 시작하기 위해 메인 페이지에서 “시작하기” 버튼을 누르면 봉사 시작 하기 Feature로 넘어간다. 먼저 그림 24와 같은 인증 페이지가 나온다. 이 페이지에는 버튼이 눌리면 서버에서 랜덤으로 인증용 4자리 숫자를 각 사용자의 화면에 출력하고 “HELPER”와 “HELPEE”는 서로의 번호를 교환하여 앱에 입력하면 서버로 해당 번호가 전송된다. 서버에서 번호 인증이 완료되면 인증완료 시간을 봉사 시작 시간으로 간주한다.



그림 25. Helper와 Helpee 봉사 시작 인증 후 페이지

인증을 완료하면 그림 25와 같이 Helper와 Helpee의 메인 페이지에서 해당하는 봉사의 상태가 “봉사 중”으로 변경된다. 만약 봉사가 종료되면 Helper와 Helpee는 “종료” 버튼을 누른다.

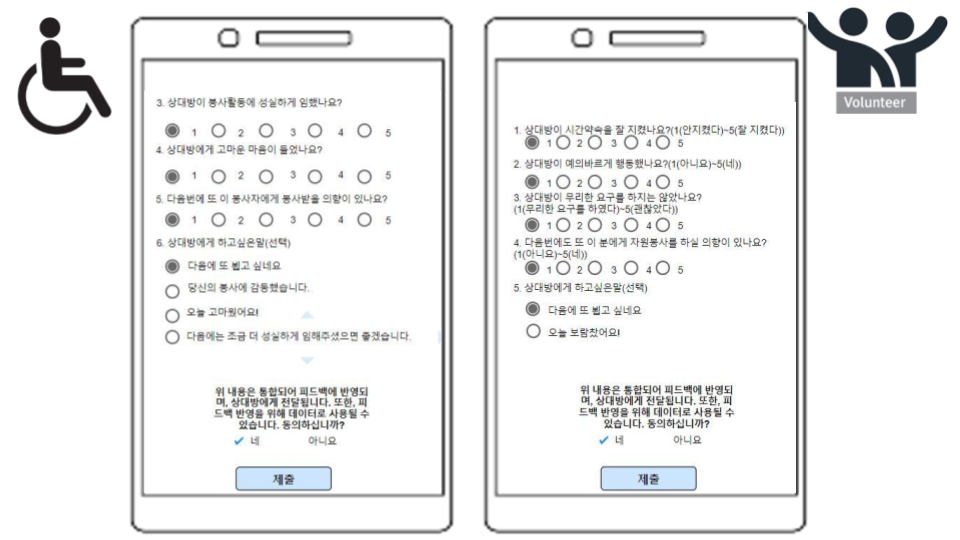


그림 26. Helper와 Helpee feedback 설문조사 페이지

마지막으로 종료 후 Helper와 Helpee는 서로에 대한 feedback을 매기는 설문조사 하기 Feature로 넘어간다. 그림 26과 같이 Feedback을 작성 후 “제출” 버튼을 누르면 추후에 관리자가 feedback 점수를 보고 Helpee에게 봉사 시간을 부여한다.

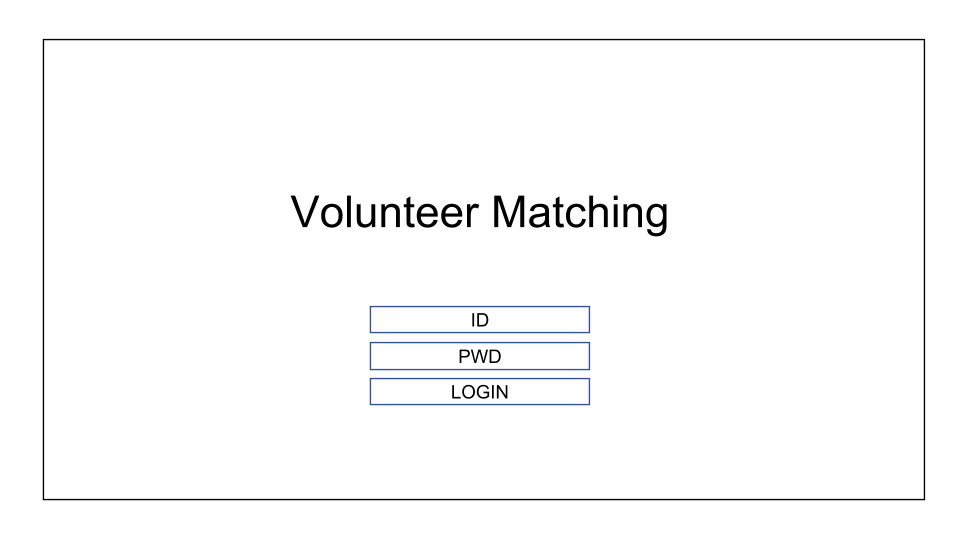


그림 27. 관리자 로그인 페이지

그림 27은 관리자가 관리자 인증 암호를 입력하는 관리자 페이지 암호 입력하기 Feature이다. 관리자는 해당 페이지에 암호를 입력하여 관리자 기능에 접근할 수 있다.

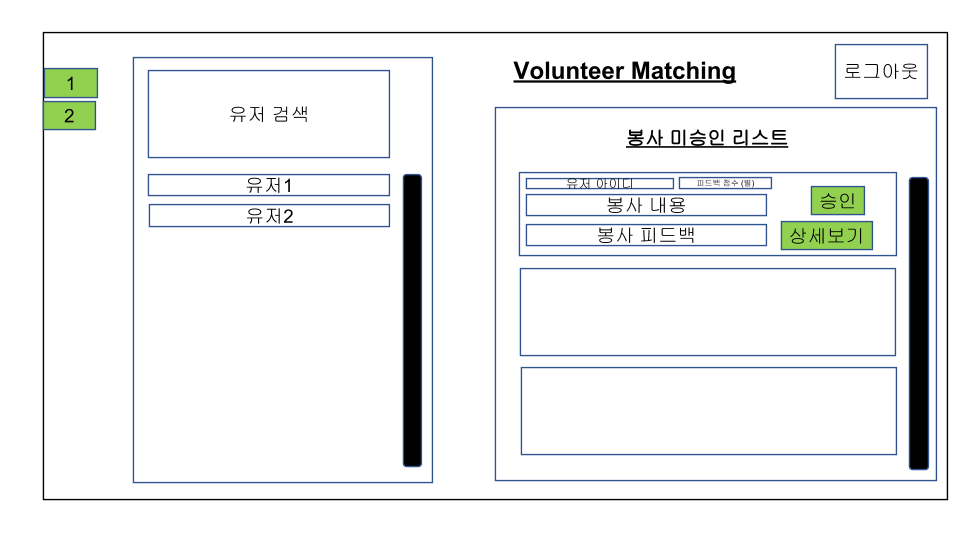


그림 28. 관리자 유저 검색 페이지

그림 28는 관리자의 기능 중 사용자 조회하기 Feature이다. 해당 페이지에 접근하면 사용자들을 조회할 수 있다. 관리자는 원하는 사용자를 선택하여 사용자의 정보, 봉사활동 내역, 피드백 정보를 확인할 수 있다. 또한 사용자의 이름 검색을 통해 원하는 사용자를 조회할 수 있다.

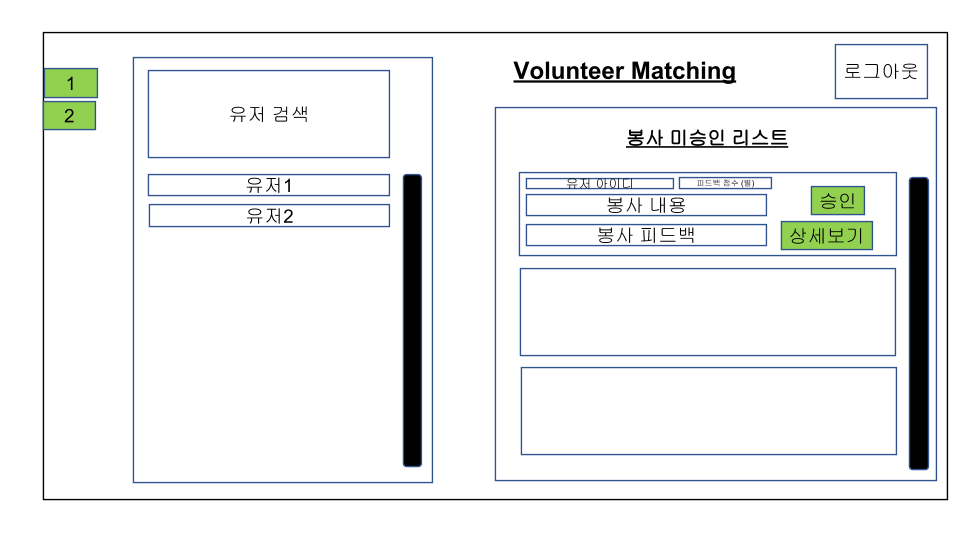


그림 29. 관리자 봉사 승인 페이지

그림 29는 봉사 시간 승인하기 Feature이다. 관리자는 해당 페이지에서 종료된 봉사에 대해 봉사 feedback을 봉사 시간을 승인할 수 있다. 그림 29에서 볼 수 있듯 “상세보기” 버튼과 “승인” 버튼이 있다. 이 봉사에 대해 자세히 보고 싶다면 “상세보기” 버튼을 눌러 볼 수 있으며 “승인” 버튼을 누르면 종료된 봉사에 대해 자원봉사자에게 봉사 시간을 할당한다.

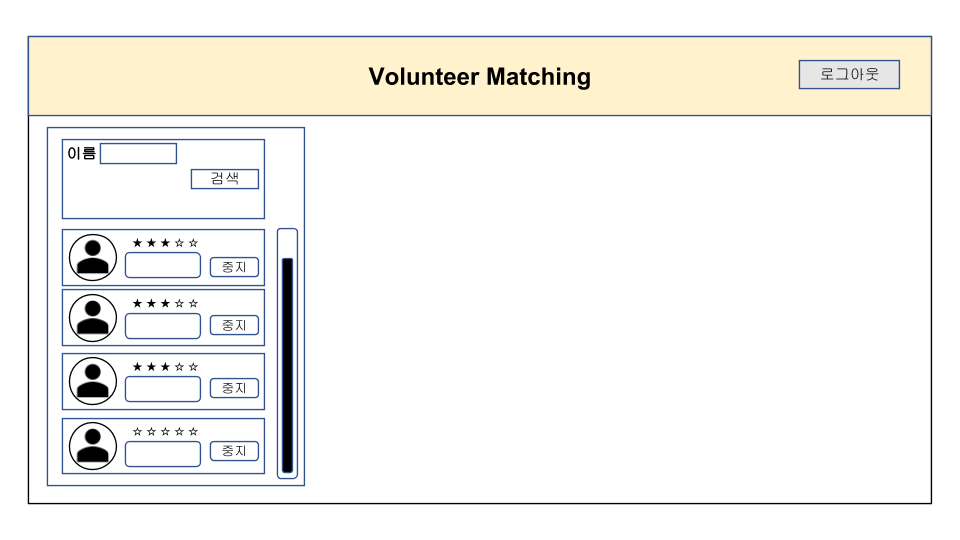


그림 30. 관리자 유저 feedback 관리 페이지

그림 30은 피드백을 바탕으로 사용자 관리하기 Feature이다. 관리자는 이 페이지에서 “HELPER”와 “HELPEE”의 피드백들을 조회할 수 있다. 관리자는 피드백들을 읽고 만약 평가가 좋지 않다면 “중지” 버튼을 눌러 그 사용자에 대해 사용 중지 처리를 할 수 있다.

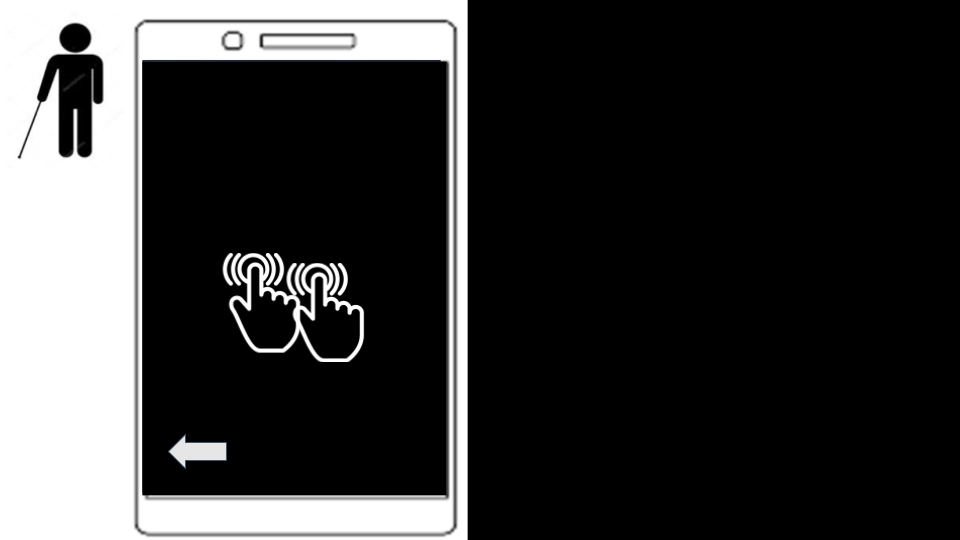


그림 31. 시각 장애인 용 페이지

그림 31 은 콜센터 연결하기 Feature이다. 시각 장애인의 경우 봉사의 시간, 위치, 종류 등 봉사를 신청하기 위한 불편한 단계들을 하기 힘들기 때문에 화면을 두 번 클릭하면 콜센터 직원에게 연결된다. 콜센터 직원은 시각 장애인의 요구에 맞게 도움 요청을 등록한다.

# 2. 설계

## 2.1 시스템 구조

2.1.1 OVERALL 시스템 구조

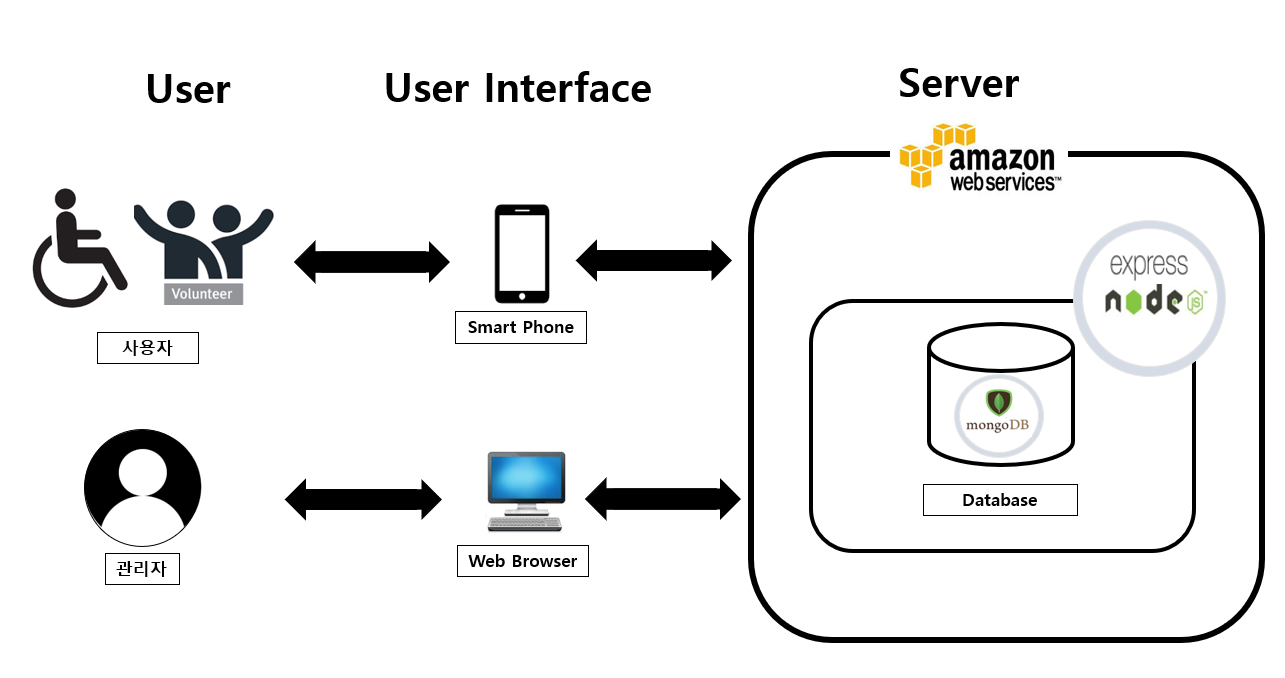


그림 32. 전체 시스템

전체 시스템은 User, User Interface, Server로 구성되어 있다.

▪ User

User에는 도움을 받고자 하는 장애인, 노약자, 환자와 도움을 주고자 하는 자원봉사자, 그리고 봉사 시간을 제공하고 시스템의 전반적인 관리를 책임지는 관리자가 있다.

▪ User Interface

User Interface는 Smart Phone, Web Browser로 구성되어 있다. 도움을 받고자 하는 사람들은 Smart Phone을 통해서 도움을 요청할 수 있고 자원봉사자들은 Smart Phone을 통해서 도움 요청에 대한 알람을 받고 봉사를 하겠다는 요청을 보낼 수 있다. 그리고 관리자는 Web Browser를 통해 봉사활동 내역을 확인한 후 자원봉사자들에게 봉사시간을 제공하고 봉사활동에 대해 전반적으로 검토한다.

▪ Server

Server는 Amazon web service상에서 Node Server와 MongoDB를 사용하는 구조로 되어 있다. Server는 User Interface로 부터 요청을 받고 database에 접근하여 여러 작업을 수행한다.

### 2.1.2 SW ARCHITECTURE

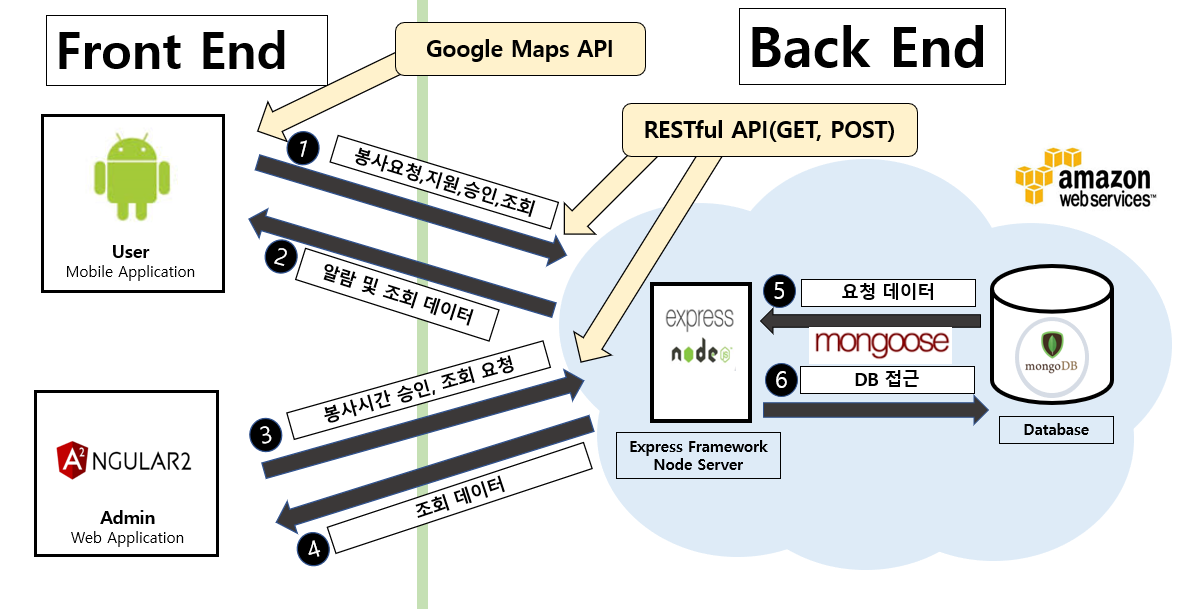


그림 33. SW Architecture

▪ Front End

Front End의 Mobile Application은 자원 봉사자 및 도움을 받는 사람들(장애인, 노약자, 환자) 이 사용하는 것이고 Web Application은 관리자가 사용하는 것이다.

사용자는 주변에 있는 봉사 처를 지도형태로 볼 수 있어야 하기 때문에 본 시스템에서는 Google Maps API를 사용한다. 그리고 Front end 에 수행한 작업을 Back End의 데이터 베이스에 반영하기 위하여 RESTful API인 GET과 POST를 사용한다. 외부에 노출되어도 되는 데이터를 서버로 전송할 때는 GET Method를 사용하고 외부에 노출되어서는 안되는 사용자 정보와 같은 데이터를 서버로 전송할 때는 POST Method를 사용한다. 또한, 앱에서는 안드로이드 xml을 통해서 사용자에게 화면을 보여주고, 웹에서는 Angular2(Typescript,Html,Css)를 통하여 관리자에게 화면을 보여준다.

▪ Back End

Back End에서는 Amazon Web Service 상에서 Express Framework 를 사용하여 Nodejs 서버를 구성하고 MongoDB를 사용하여 데이터 베이스를 구축하였다. Front-end에서 GET요청 혹은 POST 요청이 오면 Nodejs가 mongoose를 통해 MongoDB에 접근하여 요청데이터를 가지고 온다.

▪ System Flow

본 Volunteer Matching 서비스의 System Flow는 다음과 같다.

User가 서버에 봉사 요청을 하거나 봉사 지원을 승인한다는 요청 메시지를 보내거나 혹은 데이터를 조회할 경우 서버에 GET이나 POST와 같은 RESTful API를 통해 요청 메시지를 보낸다.(①,③) 그러면 서버는 DB에 접근하여(⑥) 요청 데이터를 가져와서(⑤) Front End 단에 데이터를 전달하거나(②,④) APP에 푸시알람을 전송한다.(②) 그리고 User가 자신 주변에 있는 봉사요청들을 조회할 경우 Google Maps API를 사용하여 Google Map을 띄운다.

## 2.2 Components

Table 16. Components

|  |  |  |
| --- | --- | --- |
| **Type** | **Component** | **Description** |
| 신규 | 현 위치 기반 주변 봉사처 추천 | 사용자 위치 기반의 근거리 매칭을 위해 본 서비스를 이용할 시 사용자의 Gps를 이용한다. 이는 Google Maps API를 이용해 위치좌표값을 받아올 것이다. 그리고 UI를 간단한 지도로 단순화하여 사용자들을 표시하고 원하는 위치의 사용자를 선택할 수 있도록 하여 접근성을 높일 것이다. |
| 신규 | 봉사 시간 부여를 위한 시간 측정 | 기존 자원 봉사 시스템에서는 자원봉사자들에게 서면으로 봉사 시간을 부여함으로써 자원 봉사가 인증 되지만 해당 프로젝트에서는 모바일을 통해 인증이 이루어짐으로써 서면을 통한 인증이 불가능하다.  따라서 정확한 시간 측정을 통해 봉사 시간을 부여할 것이다. 이를 위해 매칭된 사용자들이 만났음을 서버에서 랜덤하게 생성한 인증번호를 통해 확인하며 기록된 시간을 이용하여 정확한 서버시간을 통해 측정해 봉사 시간을 부여할 것이다. 따라서 즉각적인 봉사 인증이 이루어져 봉사자들이 인증을 직접 해야하는 번거로움을 줄여 편리성을 제공한다. |
| 기존 | 푸시 알람 | Helper와 Helpee는 푸시 알람을 통해 상대방이 요청한 자원봉사 내역이나 승인한 자원봉사 내역을 확인할 수 있다. |

## 2.3 데이터 정의

Volunteer System에서는 총 3개의 데이터 베이스 스키마가 필요하다.

Table 17. User schema

|  |  |  |
| --- | --- | --- |
| **userSchema** | | |
| **Volunteer Matching 시스템을 사용하는 사용자와 관리자에 대한 정보를 저장한다.** | | |
| **Description** | **KEY** | **VALUE** |
| 사용자의 유형(Helper,Helpee,Blind,Admin)을 구분한다. | userType | Number |
| 사용자의 ID | userID | String |
| 사용자의 Password | userPWD | String |
| 사용자의 본명 | userName | String |
| 사용자의 휴대폰 번호 | userPhoneNum | String |
| 주민등록번호 | ID\_Num | String |
| 장애인 복지카드 일련번호(장애인일 경우 입력) | handicappedNum | String |
| 사용자의 직업(자원봉사자일 경우 입력) | userJob | String |
| 사용자의 주민등록상 표기되어 있는 주소 | userAddr | String |
| 사용자의 email(email id) | userEmail\_before | String |
| 사용자의 email(domain address) | userEmail\_after | String |
| 봉사경력 | userVolCareer | String |
| 사용자 평점 | userFeedbackVal | Number |
| 사용자 고유 인증번호 | userAuthenNum | Number |

Table 18. Volunteer Schema

|  |  |  |
| --- | --- | --- |
| **volunteerSchema** | | |
| **봉사내용에 대한 상세한 정보를 저장한다.** | | |
| **Description** | **KEY** | **VALUE** |
| 봉사 ID | volunteerID | Number |
| Helper의 ID | helperID | String |
| Helpee의 ID | helpeeID | String |
| 봉사의 유형(외출,가사활동 등)을 구분한다. | volunteerType | String |
| 봉사 날짜 | volunteerDate | Date |
| 봉사 시작 시간 | volunteerStartTime | Date |
| 봉사 종료 시간 | volunteerFinishTime | Date |
| 봉사 진행 시간 | volunteerTime | Number |
| 봉사처의 위도 좌표 | volunteerPositionLat | Number |
| 봉사처의 경도 좌표 | volunteerPositionLon | Number |
| Helper의 인증여부 | volunteerHelperAuthen | Number |
| Helpee의 인증여부 | volunteerHelpeeAuthen | Number |
| 봉사 완료 여부 | volunteerDone | Number |

Table 19. Feedback Schema

|  |  |  |
| --- | --- | --- |
| **feedbackSchema** | | |
| **봉사가 끝난 뒤 봉사활동 전반의 상대방 태도를 평가하기 위한 정보를 저장한다.** | | |
| **Description** | **KEY** | **VALUE** |
| 봉사 ID | volunteerID | Number |
| Helper의 평점 | helperScore | Number |
| Helpee의 평점 | helpeeScore | Number |
| Helper가 Helpee에 대해 평가한 내용 | helperComment | String |
| Helpee가 Helper에 대해 평가한 내용 | helpeeComment | String |
| Helper가 Helpee를 다시 만나기 원하는가 여부 | helperWanted | Number |
| Helpee가 Helper를 다시 만나기 원하는가 여부 | helpeeWanted | Number |

## 2.4 API 및 Interface 정의

### 2.4.1 외부 API

Table 20. 외부 API

|  |  |  |
| --- | --- | --- |
| **API** | **전달 데이터** | **설명** |
| Google Maps API | GPS 좌표 | Google Cloud Platform에서 제공하는 지도 사용 API |

### 2.4.2 Restful API

Table 21. Restful API

|  |  |  |  |
| --- | --- | --- | --- |
| **Restful API type** | **Path** | **전달 데이터** | **설명** |
| POST | /login | ID/PWD | 로그인 |
| GET | /logout |  | 로그아웃 |
| POST | /signup | 회원 정보 | 회원가입 요청 후 저장 |
| POST | /authorization | 본인 정보 | 본인인증 |
| POST | /startVolunteer | 인증번호 | Server가 Helper와 Helpee한테 인증번호를 전송한다. |
| POST | /endVolunteer | 인증번호 | Server가 Helper와 Helpee한테 인증번호를 전송한다. |
| GET | /helpee/getVolunteertList | 봉사 리스트 | Helpee가 요청한 봉사내역들을 전송한다. |
| GET | /helper/getVolunteertList | 봉사 리스트 | Helper가 앞으로 수행할 봉사내역들을 전송한다. |
| GET | /getLocation | 위치 좌표, 주변 봉사처 | Helper의 현재 위치와 근처에 있는 주변 봉사처의 위치를 전송한다. |
| GET | /getWantedVolunteerList | 봉사 리스트 | 사용자의 맞춤검색 조건에 맞는 봉사 리스트를 전송한다. |
| POST | /helper/choiceVolunteer | 봉사 항목 | Helper의 봉사 리스트에 봉사항목이(매칭중) 추가되고 Helpee에게 푸시알람을 전송한다. Helper가 선택한 Helpee의 봉사항목을 승인대기로 바꾼다. |
| POST | /helpee/acceptVolunteer | 봉사 항목 | Helpee가 승인할 봉사 항목의 상태를 “승인 완료”로 바꾸고 Helper의 봉사 항목 상태도 “매칭 완료”로 바꾼다. Helper에게 푸시알람을 전송한다. |
| POST | /helpee/requestVolunteer | 봉사 항목 | Helpee의 봉사 리스트에 봉사항목(승인대기중)이 추가 되고 Helper에게 푸시알람을 전송한다. |
| GET | /helper/getCompleteList | 봉사 리스트 | Helper가 현재까지 수행 완료한 봉사 내역들을 전송한다. |
| POST | /helper/survey | Feedback 리스트 | Helpee의 태도를 평가한 feedback 리스트 저장 |
| POST | /helpee/survey | Feedback 리스트 | Helper의 태도를 평가한 feedback 리스트 저장 |
| GET | /admin/getUserList | 회원 정보 | 회원정보를 전송한다. |
| GET | /admin/getDoneVolunteerList | 봉사 리스트 | 이제까지 완료된 봉사 내역을 전송한다. |
| POST | /admin/dropoutUser | 회원 | 사용자를 시스템 상에서 삭제시킨다. |
| GET | /helper/getUserAuthenNum | Helper 회원의 인증번호 | 회원의 인증번호를 전송한다. |
| GET | /helpee/getUserAuthenNum | Helpee 회원의 인증번호 | 회원의 인증번호를 전송한다. |
| POST | /helper/volunteerHelperAuthen | Helper 회원의 인증 여부 | Helper 회원의 인증 여부를 변경한다. 인증의 성공 = 1, 인증의 실패 = 0 |
| POST | /helpee/volunteerHelpeeAuthen | Helper 회원의 인증 여부 | Helpee 회원의 인증 여부를 변경한다. 인증의 성공 = 1, 인증의 실패 = 0 |

# 3. 수행 계획

## 3.1 개발환경

**[개발 환경 및 언어]**

Table 22. 개발환경

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Client** | **Server** | **Framework** | **Tool** | **OS** | **Browser** |
| -JAVA  -HTML5  -CSS3  -TYPESCRIPT | -Node.js | -Express  -Angular2 | -WEBSTORM  -ANDROID STUDIO 3.1  -SUBLIME TEXT | -WINDOWS 10  -ANDROID  -LINUX UBUNTU 16.04 LTS | -GOOGLE  -CHROME |

**[Framework 활용 및 운영 방안]**

Table 23. Framework

|  |  |
| --- | --- |
| **Framework** | **Description** |
| Angular.js | SPA방식의 프론트엔드 웹 개발을 위한 프레임워크 |
| Bootstrap | 오픈소스 기반의 HTML5, CSS3 UI 프레임워크 |
| Express | Node.js 런타임 환경에서 제공되는 웹 프레임워크 |

## 3.2 리스크 분석

Table 24. Risk

|  |  |  |
| --- | --- | --- |
| **Type** | **Risk** | **Solution** |
| Technology | 정확한 위치 기반의 추천 서비스를 제공하기 위해 Google Maps API를 이용할 것인데, 이를 본 프로젝트에 맞게 활용하기 위해 UI도 변경하고 구성을 변형하는 데 어려움이 있다. | API를 이용할 때, 함수나 변수들은 최대한 변형하지 않고 원본 그대로를 유지하는 방향으로 개발하고, 변경해야 할 값이 있을 때에는 회의를 통해 결정하며 변경 후 반드시 테스트를 통해 오류 발생이 나지 않는지 확인한다. |
| Technology | Android Studio를 이용해 앱을 개발하는데 버전이 계속 업데이트 되며 필요한 API나 툴을 사용 시 협업에 문제가 생길 수 있다. | 기존 상태였던 각자의 Android Studio에서 모든 version을 최신으로 업데이트를 하고 시작한다. 이후 .gradle 파일에 생성된 version은 섞이지 않도록 ignore파일로 만들고 이후에 필요한 API나 툴을 설치 시에는 다른 파일에 정리하여 각자 설치할 수 있도록 공유한다. |
| Service | 본 프로젝트의 결과물을 서비스로 출시하였을 때 자원봉사자나 도움을 요청한 분들의 태도에 따라 사회적 문제가 발생할 수 있다. | 본 서비스를 이용하여 사회적 문제가 발생하지 않기 위해서는 서로간의 예의가 가장 중요할 것이므로 매칭이 되기 전 상대방의 정보를 충분히 인지할 수 있도록 시나리오를 작성하고 매칭이 되고 난 뒤에는 동영상 시청이나 퀴즈 등을 통해 윤리적 사고를 배양할 수 있도록 한다. |
| Integration | Git을 이용하여 협업을 진행하는데 각자의 push가 엉켜 오류가 난 시점을 알기 힘들 수 있다. | Git에 각자가 push를 진행할 때에는 꼭 commit을 하고 난 뒤 진행하여 이전 상황을 인지해야 하며 오류가 나지 않도록 test 코드를 통해 확인해본 후 정상적일 때에만 push를 진행한다. |
| Team Work | 개발 목표에 맞게 개발을 하지만 시간 부족 등의 이유로 중간에 더 나은 방향이나 차선책을 선택해야하는 시점이 올 수 있다. | 본 프로젝트를 위해 작성한 본 기획서를 토대로 최대한 개발을 맞추고 다른 방향을 위한 결정을 할 때에는 팀원간의 회의를 통해 다수결로 결정한다. |
| Testing | 테스트를 진행하기 위해 실제 도움이 필요한 분들과 자원봉사자를 섭외하기에 무리가 있을 수 있다. | 본 프로젝트의 결과물은 상업을 위한 서비스가 아닌 복지 차원의 도움이 필요하신 분들의 삶의 질 향상을 위한 것임으로 국가적 복지단체를 통해 테스트를 진행하며 불가능할 시에는 불가피하게 테스트는 진행하지 않는다. |

## 3.3 개발 일정

Table 25. 개발 일정

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | Feature | Task Description | Owner |
| 1  (3.20~4.19) | 프로젝트 기획 | 프로젝트 기획서 작성 | 모두 |
| 개발준비 | Back end 구축(Web 서버) | 박준민 |
| Back end 구축(App 서버) | 고광표 |
| 데이터베이스 구축 | 정소영 |
| Front end 구축(Angular2) | 고보원 |
| 2  (4.20~5.7) | 로그인 및 회원가입 하기 | 안드로이드 로그인 화면 및 회원가입 화면 구성, 회원 정보 데이터베이스에 저장 | 정소영 |
| 관리자 페이지 로그인 하기 | 웹 관리자 로그인 화면 구성 및 로그인 기능 구현, 데이터베이스에 저장 | 고보원 |
| 사용자 조회하기 | 웹 사용자 조회 화면 구성 및 기능 구현, 데이터베이스 정리 | 고보원 |
| 현재 상태 조회하기 | 안드로이드 사용자의 현재 상태(매칭 대기중, 매칭 완료) 화면 구성 | 고광표 |
| 위치기반 봉사 검색하기 | Google API 사용(Google Map)한 안드로이드 위치기반 봉사 검색 구현 | 박준민 |
| 맞춤 기반 봉사 검색하기 | 안드로이드 맞춤 기반 봉사 검색 기능 구현(데이터 베이스에서 봉사 리스트 뽑아오기) | 정소영 |
| 3  (5.8~5.25) | 도움 요청하기 | 안드로이드 도움 요청 기능 구현, 데이터 베이스에 새로운 도움 등록 | 고광표 |
| 봉사 자원하기 | 안드로이드 봉사 자원하기 기능 구현 | 박준민 |
| 봉사 시작 및 종료하기 | 안드로이드 봉사 시작, 종료 기능 구현, App 서버에서 봉사 시작/종료 인증 번호 생성, 검증 구현 | 정소영, 박준민 |
| 설문조사 하기 | 안드로이드 설문 조사 화면 구성 및 설문조사 정보 데이터베이스에 저장 | 고보원 |
| 피드백 바탕으로 사용자 관리하기 | 데이터베이스에 저장된 피드백 바탕으로 사용자 관리하기 구현 | 고광표 |
| 봉사 시간 승인하기 | 웹 봉사 시간 승인 기능 구현 | 고보원 |
| 4  (5.28~6.11) | 봉사 자원 알림 받기 | App서버에서 FCM을 이용한 봉사 자원 푸쉬 알림 구현 | 박준민 |
| 봉사 자원 승낙하기 | 등록된 봉사에서 자원한 자원봉사자 선택 기능 구현, 해당 봉사자에게 FCM을 통한 푸쉬 알림 전송 | 고광표 |
| 봉사 매칭 완료 알림 받기 | App서버에서 FCM을 이용한 봉사 매칭 완료 푸쉬 알림 구현 | 정소영 |
| 근처 봉사 실시간 알림 받기 | App서버에서 FCM을 이용한 봉사 실시간 푸쉬 알림 구현 | 고보원 |
| 콜센터 연결하기 | 안드로이드 앱에서 콜센터 전화번호로 바로 연결할 수 있도록 구현 | 박준민 |

## 3.4 비용 분석

Table 26. 비용 분석

|  |  |  |
| --- | --- | --- |
| **비용 분석 대상** | **설명** | **비용** |
| Amazon Web Service | 데이터베이스 및 back-end 구성을 위한 서버 유지비 (Free Stack 사용 예정) | 약 0원 |
| API 사용 비용 | 무료로 release된 오픈소스 사용 예정 | 약 0원 |
| 부스 비용 | 부스 설치 비용 | 약 10만원 |
| 합계 | 약 10만원 | |

## 3.5 업무분장 계획

### 3.5.1 Feature / 컴포넌트 기반

Table 27. Feature / 컴포넌트 기반 업무분장

|  |  |
| --- | --- |
| 팀원 | Feature 기반 업무 분장 |
| 고광표 | Web 및 App 서버 구축 |
| 현재 상태 조회하기 |
| 도움 요청하기 |
| 피드백 바탕으로 사용자 관리하기 |
| 봉사 자원 승낙하기 |
| 고보원 | 관리자 페이지 로그인 하기 |
| 사용자 조회하기 |
| 설문조사 하기 |
| 봉사 시간 승인하기 |
| 근처 봉사 실시간 알림 받기 |
| 박준민 | Web 서버 구축 |
| 위치기반 봉사 검색하기 |
| 봉사 자원하기 |
| 봉사 종료하기 |
| 봉사 자원 알림 받기 |
| 콜센터 연결하기 |
| 정소영 | 데이터 베이스 구축 |
| 로그인 및 회원가입하기 |
| 맞춤 기반 봉사 검색하기 |
| 봉사 자원하기 |
| 봉사 매칭 완료 알림 받기 |

### 3.5.2 기타 과제관리/Deploy/발표 등 부가적 업무 분장

Table 28. 부가적 업무 분장

|  |  |
| --- | --- |
| 팀원 | 기타 업무 분장 |
| 고광표 | 예산 관리, 최종 발표 |
| 고보원 | 멘토미팅 일정 조율, 팀 미팅 일정 조율 |
| 박준민 | 공모전 참가 신청 및 일정 관리, 중간 발표 |
| 정소영 | 컨셉발표, 기획발표, Deploy |

## 3.6 협업 방안

### 3.6.1 소스코드 관리

Table 29. 소스코드 관리

|  |  |
| --- | --- |
| **Tool** | **설명** |
| Github | ▪ 운영방식  Github의 하나의 Master branch 상에서 모든 팀원이 작업한다.  commit message는 ‘YYYYMMDD-이름-설명 ’ 순으로 작성한다.  Database는 자신의 local directory에 저장하고 git에 올리지 않는다.    ▪ 명령어 입력 순서   1. git pull 2. conflict 발생시 conflict 해결 3. git add \* 혹은 git add 파일명(.gitignore 파일은 해당 명령어를 통해 update되지 않을 수 있으므로 git add .gitignore 추가 입력) 4. git commit -m “commit message” 5. git push origin master |

### 3.6.2 통합 및 테스트

Table 30. 통합 및 테스트

|  |  |
| --- | --- |
| **Tool** | **설명** |
| Travis | Travis CI와 github 연동한다. 연동된 후 Travis는 .travis.yml 파일을 통해서 이벤트 발생시 Travis가 어떤 동작을 할 것인지 결정한다  requirements.txt 파일에 설치할 dependency들을 명시한다.  GitHub 저장소에 새로운 커밋이 들어오면, Travis CI가 바로 알게 되고, 설정에 따라 빌드 테스트를 수행한다.  설정을 위해서는 저장소에 .travis.yml이라는 YAML 파일을 넣어 두면 된다. Node.js용 설정 파일의 예는 아래와 같다.  language: node\_js  node\_js:  - "0.10"  - "0.11"  저장소에 git push를 하고 나면, Travis CI가 테스트를 수행한다. |

### 3.6.3 이슈 관리

Table 31. 이슈 관리

|  |  |
| --- | --- |
| **Tool** | **설명** |
| Trello | 제품을 개발하기 위한 전체적인 백로그(해야 할 일)들을 관리 한다.  백로그 목록에 있는 항목들은 "todo" -> "in process" -> "test" -> "done"으로 옮겨지게 되고 모든 백로그 데이터들이 done으로 오게 되면 해당 항목은 종료하여 진행 현황을 파악한다. |
| Slack | Trello와 연동하여 Trello에서 관리하는 진행 현황이나 완료 현황을 메신저 형태로 소통한다. |

# 4. 출시 계획

## 4.1 성과 측정 방안

Table 32. 성과 측정 방안

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **영역** | **세부 영역** | **성과 지표** | **목표** | **비중** | **타겟** |
| 접근성 | 사용자 접근성 | 어플리케이션 실행 시점부터 도움 요청까지의 시간 | 30s 이내 | 25% | 개발자 |
| 호환성 | 웹 호환성 | 지원하는 브라우저 개수 | 3개 이상 |  |  |
| 어플리케이션 호환성 | 지원하는 안드로이드 최소 버전 | Jelly bean(Android 4.1) 이상 |  |  |
| 기술개발 | 서버 시간 | 봉사 지원 시점 부터 해당 도움을 요청한 사람에게 푸시알람이 갈때까지의 시간 | 5s 이내 | 5% | 개발자 |
| 정확도 | 구글 지도 상 정확한 위치에 주변 가능 봉사처 로딩 | 오차범위 100m 이내 | 10% | 개발자, 사용자 |
| 만족도 | 사용자 만족도 | 서비스 출시 후 설문을 통한 사용자들의 만족도 측정 | 70% 이상 | 50% | 사용자 |
| 검색 만족도 | 맞춤 검색을 통한 상대방 검색 시 사용자 조건에 부합한 결과인지 측정 | 50% 이상 | 10% | 사용자 |
| 향후 확장 | 국가 기관의 협력 | 국가 복지 기관과 협력하여 실제 국가가 인정하는 사회 봉사시간 제공 | 국가 복지 기관의 승인 | 10% | 개발자 |

## 4.2 데모 시나리오

### 4.2.1 데모 시나리오 1

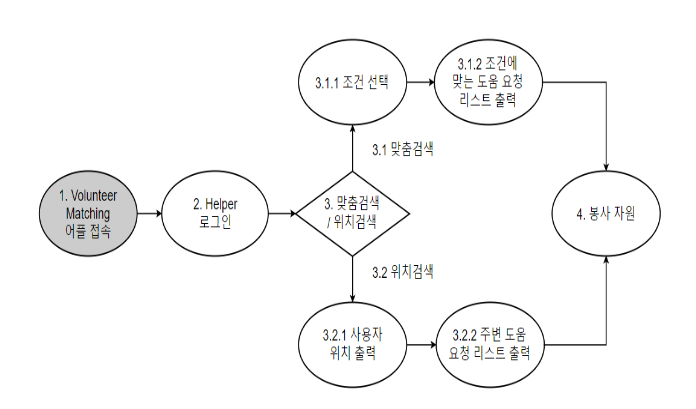
****

그림 34. 데모 시나리오 1

Table 33. Demo Scenario 1

|  |  |
| --- | --- |
| **Demo Scenario 1** | |
| **Title** | 봉사 자원 |
| **Actor** | Helper |
| **데모 시나리오** | 1. Helper는 Volunteer Matching 앱에 접속한다.  2. 로그인한다.  3.1 맞춤 검색 버튼을 누른다.  3.1.1 Helper가 원하는 조건을 선택한다.  3.1.2 조건에 맞는 도움 요청이 화면에 출력된다.  3.2 위치 검색 버튼을 누른다.  3.2.1 사용자의 위치를 구글 지도 상에 출력하고  3.2.2 사용자 근처의 봉사 가능처가 구글지도 상에 출력한다.  4. 원하는 봉사항목에 자원한다. |

### 4.2.2 데모시나리오 2

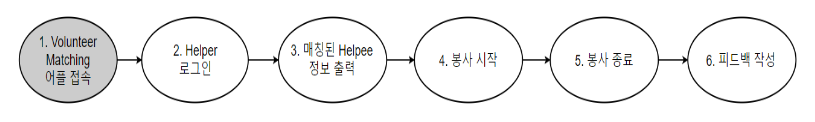


그림 35. 데모 시나리오 2

Table 34. Demo Scenario 2

|  |  |
| --- | --- |
| **Demo Scenario 2** | |
| **Title** | 봉사(Helper) |
| **Actor** | Helper |
| **데모 시나리오** | 1. Volunteer Matching 어플에 접속한다.  2. Helper 아이디로 로그인한다.  3. 매칭된 Helpee의 정보를 출력해준다.  4. 봉사를 시작한다.  5. 봉사를 종료한다.  6. 피드백을 작성한다. |

### 4.2.3. 데모시나리오 3

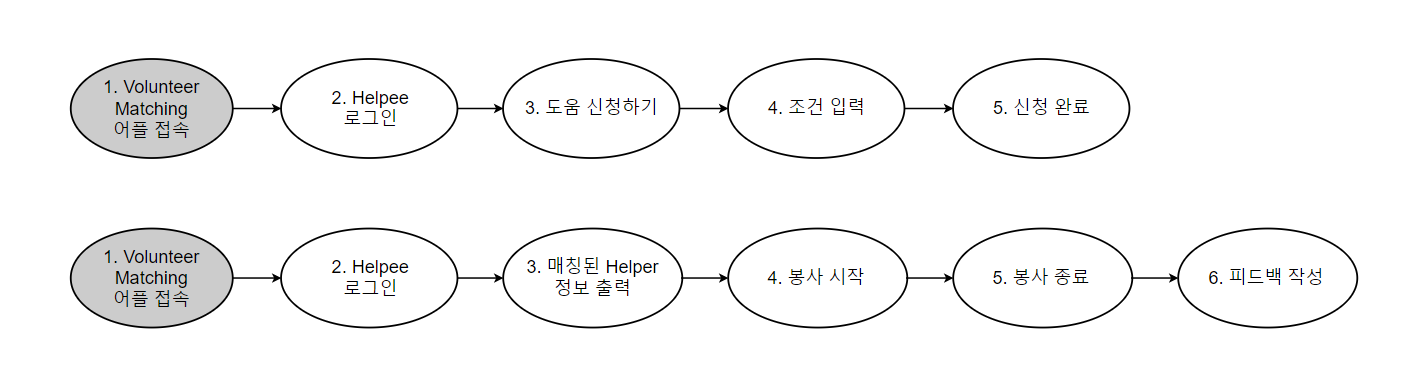


그림 36. 데모 시나리오 3

Table 35. Demo Scenario 3

|  |  |
| --- | --- |
| **Demo Scenario 3** | |
| **Title** | 도움 신청 |
| **Actor** | Helpee |
| **데모 시나리오** | 1. Volunteer Matching 어플에 접속한다.  2. Helpee 아이디로 로그인한다.  3. 도움 신청하기를 누른다.  4. 조건을 입력한다.  5. 신청을 완료한다. |

### 4.2.4. 데모시나리오 4

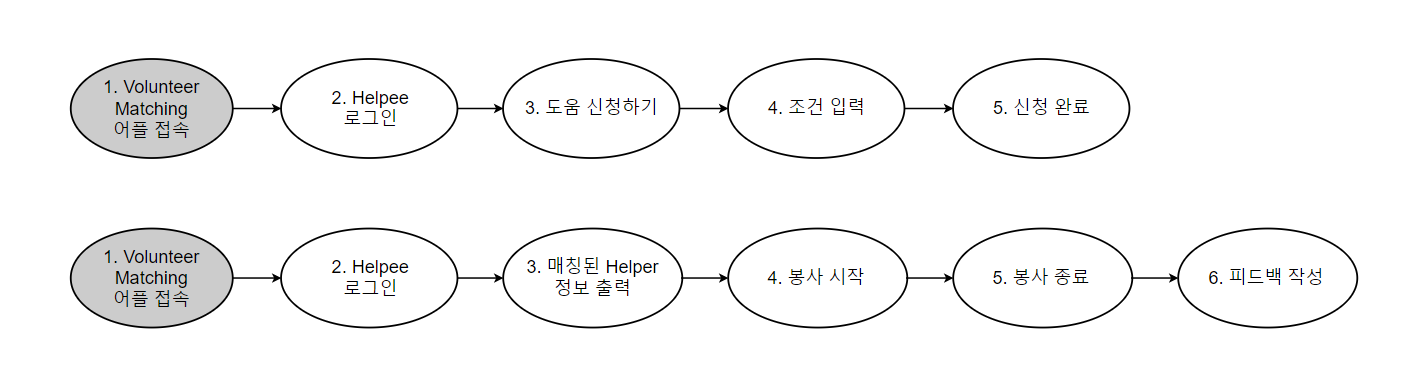


그림 37. 데모 시나리오 4

Table 36. Demo Scenario 4

|  |  |
| --- | --- |
| **Demo Scenario 4** | |
| **Title** | 봉사(Helpee) |
| **Actor** | Helpee |
| **데모 시나리오** | 1. Volunteer Matching 어플에 접속한다.  2. Helpee 아이디로 로그인한다.  3. 매칭된 Helper의 정보를 출력한다.  4. 봉사를 시작한다.  5. 봉사를 종료한다.  6. 피드백을 작성한다. |

### 4.2.5. 데모시나리오 5

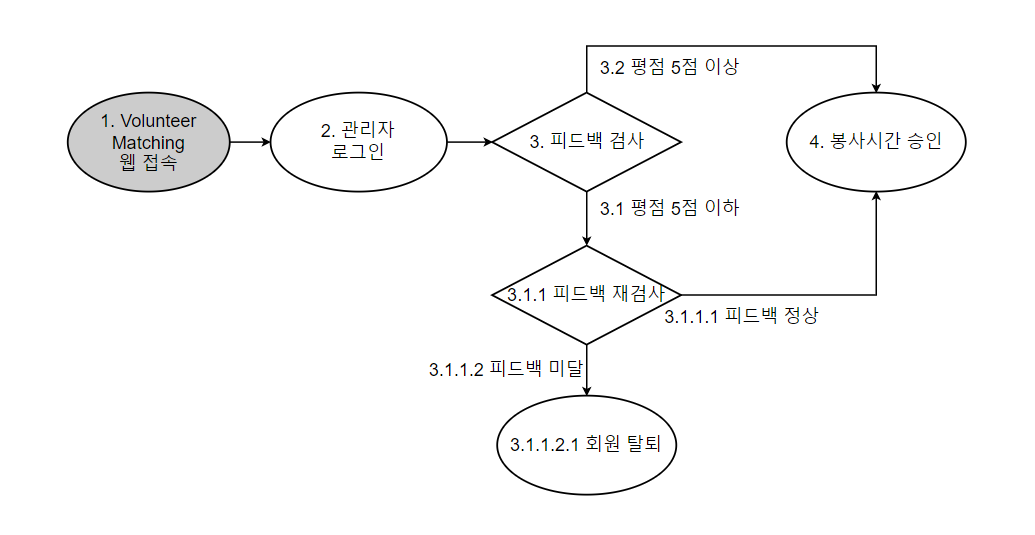


그림 38. 데모 시나리오 5

Table 37. Demo Scenario 5

|  |  |
| --- | --- |
| **Demo Scenario 5** | |
| **Title** | 봉사시간 승인 |
| **Actor** | 관리자 |
| **데모 시나리오** | 1. Volunteer Matching 관리자 페이지에 접속한다.  2. 관리자 아이디로 로그인한다.  3. 피드백을 검사한다.  3.1 피드백 평점이 5점 이하인 경우  3.1.1 피드백을 재검사한다.  3.1.1.1 피드백 내용이 정상인 경우  3.1.1.2 피드백 내용이 미달인 경우  3.1.1.2.1 해당 회원을 탈퇴시킨다.  3.2 피드백 평점이 5점 이상인 경우  4. 해당 봉사의 봉사시간을 승인한다. |

### 4.2.5. 데모시나리오 6

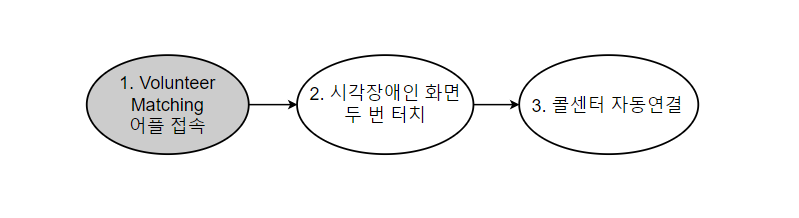


그림 39. 데모 시나리오 6

Table 38. Demo Scenario 6

|  |  |
| --- | --- |
| **Demo Scenario 6** | |
| **Title** | 시각장애인 콜센터 연결 |
| **Actor** | 시각장애인 |
| **데모 시나리오** | 1. Volunteer Matching 어플에 접속한다.  2. “시각 장애인 입니까?” 프레임 두번 클릭한다.  3. 콜센터로 자동 연결한다. |