0 개정 이력

개정일	버전	개정 내용	개정위치	작성자/검토자
2022.10.17	0.1	최초 작성		김혜수,유수연, 유영주,황성민
2022.10.30	0.2	프로젝트 주제 변경 (제품 관련 모든 부분)	1, 2.2, 4, 13	김혜수,유수연, 유영주,황성민
	0.3			유영주, 황성민
2022.11.23	0.4	`자원관리 비용, 문서화` 개정	2.1, 10	김혜수
2022.12.16	0.5	`유지보수` 개정	11	유수연
2022.12.17	0.6	`설치, 인수`개정	12	유수연
2022.12.18	0.7	`문서화` 개정	10	유영주

1 개 요

1.1 프로젝트 개요

울산제일제보에서 새내기 유권자를 대상으로 진행한 설문조사 자료를 활용하여 국회 정보 제공 현황의 개선점을 파악하였다. 응답한 학생 중 70%가 정치에 관심을 가지고 있다고 답했다. 그럼에도 정치에 대체적으로 무관심한 이유가 무엇이냐는 질문에 35%의 학생들이 ‘정치가 어렵게 느껴져서’라고 답했으며 거주지역 정치인에 대해서는 91%의 학생들이 모르거나 관심이 없다고 답했다.

이에 따라 새내기 유권자들이 보다 쉽게 입문할 수 있도록 돕고 국회데이터의 정보접근성을 향상하기 위한 서비스를 제공함을 목표로 한다. 앱 내에서는 거주 지역을 기반으로 해당 지역 의원을 탐색할 수 있도록 하고 정당별 등의 필터링을 통해 의원 상세 정보를 제공한다. 또한 직접 의견을 제시할 수 있도록 문의처와 연결하여 능동적인 참여를 독려한다.

1.2 프로젝트의 산출물

본 프로젝트의 산출물은 안드로이드 앱의 형태로 서비스된다. WENA(우리(we)가 찾는 국회정보(National Assembly))는사용자의 국회의원 관련 정보 탐색의 편의를 제공하는

목표를 가진다. 사용자는 앱 내에서 제21대 국회의원의 정보를 조회할 수 있고, 정당별 의원 목록을 확인하거나 국회의원을 검색할 수 있다. 의원 통계 메뉴에서는 필터링을 통한 의원 인원수 통계, 의원별 통계를 조회할 수 있다. 국회에서 운영하는 다른 서비스로 연결하여 사용자의 국회 활동 접근성을 높인다. 또한 문의하기 기능을 지원해 국회 또는 국회의원 연락처에 전화 혹은 메일을 직접 전송하도록 하여 소통창구로서 기능할 수 있다.

1.3 정의, 약어

단어	설명	약어	동의어
일반 사용자	본 서비스를 이용하기 위해 개인정보를 제공하여 회원가입을 마친 사람	사용자	유저, 고객, 회원
의원 목록	의원 정보가 RecyclerView에 목록 형태로 표시되는 화면	목록	목록 화면
의원 목록 필터링	의원 목록에서 필터링 조건(정당별)에 포함되는 의원만을 표시하는 작업	필터링	

2 자원 및 일정 예측

2.1 자원

가. 인력

역할	인원
디자인(UI/UX 설계)	2 / 유수연, 유영주
프론트 개발(kotlin)	2 / 유수연, 유영주
백엔드 개발(node.js+mongoDB)	2 / 김혜수, 황성민

테스트	4 / 김혜수, 유수연, 유영주, 황성민

나. 비용

총합 **895,549**
총비용 인건비+ 자원비 +계비용 = 160,000 + 355,549 +380,000

연간 지출내역서 년 월 일

지 출 내 역				
날 짜	구 분	지출항목	금 액	비 고
9월	인력	인건비	40,000	
	자원비	Amazon EC2	86,031	
	자원비	Amazon Simple Storage Service(S3)	26	
	기타 비용	참고 문헌비	120,000	참고 서적 30,000원/권X 4권
10월	인력	인건비	40,000	
	자원비	Amazon EC2	86,031	
	자원비	Amazon Simple Storage Service(S3)	26	
	기타 비용	재료비	125,500	Printer Ink 50,000/개 X 2개 Print 용지 250/매 X 10원 X 5권
11월	인력	인건비	40,000	
	자원비	Amazon EC2	86,031	
	자원비	Amazon Simple Storage Service(S3)	26	
	기타 비용	회의비	60,000	6000원/회 X 10주
11월	인력	인건비	50,000	정규 지출 비용 40,000원 디자인 로고 의뢰 10,000원
	자원비	Amazon EC2	86,031	
	자원비	Amazon Simple Storage Service(S3)	26	
	기타 비용	인쇄비	150,000	사용자지침서 10,000원/권 x 5권 개발문서 10,000원/권 x 5권 홍보용 팜플렛 10,000원/권 x 5권
합 계	380,000			

총비용 인건비+ 자원비 +기타비용 = 160,000 + 355,549 +380,000

연간지출내역서 wena.xlsx

2.2 일정

구분	수행 내용	일정								
		10월				11월				
		2주차	3주차	4주차	5주차	1주차	2주차	3주차	4주차	5주차
	프로젝트 주제 선정 및 구성									

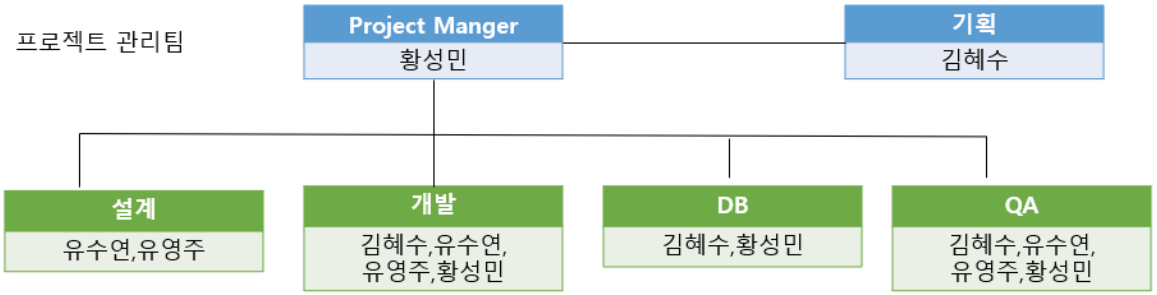
구분	수행 내용	일정								
		10월				11월				
		2주차	3주차	4주차	5주차	1주차	2주차	3주차	4주차	5주차
계획	기능 구체화									
	앱 ui 기획									
	수행계획서 작성									
분석	애플리케이션 시장조사 및 분석									
	서비스의 필요성 및 활용성 검토									
	요구사항 분석서 작성									
설계	앱 사용 흐름, UI/UX 세부 요소 설계									
	앱 프로토타입 및 동작 과정 설계									
	DB 스키마 설계 및 데이터 샘플 수집									
	REST API 설계									
	설계서 작성									
개발	회원가입&로그인									
	스플래시 화면									
	메인 페이지									
	의원 목록 페이지									
	의원 상세									

구분	수행 내용	일정								
		10월				11월				
		2주차	3주차	4주차	5주차	1주차	2주차	3주차	4주차	5주차
	페이지									
	의원 통계 페이지									
	국회 웹사이트 연결 페이지									
	문의하기 (전화)									
	문의하기 (메일)									
	메일 전송 페이지									
	부가기능									
테스트	기능 및 성능 테스트									
	피드백 수집, 기능 보충									
	최종 완성 버전 제출									
종료	완료보고서									

3 조직 구성 및 인력 배치

3.1 조직 구성

직능별 프로젝트 조직



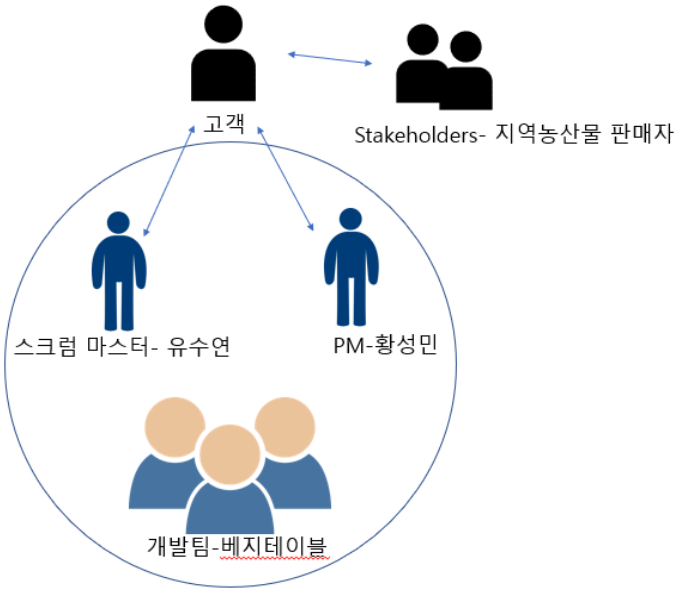
3.2 직무 기술

1. 조직별 역할

직무	역할 및 책임
개발자	- 프로젝트 관리 - 프로젝트 추진 전략 수립 및 프로젝트 계획,실행계획 수립 - 의원 데이터 설계 및 구축 - 정당별·성별·소속위원회 필터링 모듈 및 검색 커스터마이징 및 기술 - 연결 사이트 설계/ 구현 - 국회 페이지 담당 운영자 교육
PM, 기획자	- 사업총괄 - 주요사안의 최종 의사결정 - 시스템 구축 관련 주요 정책 결정 - 시스템 구축 방향 조정 - 프로젝트 과정에서 관리/감독 - 구매처 연결 관련 내역 지원 - 개발 업무 지원 및 산출물 검토 - 개발 업무 인수 및 운용
스크럼 마스터	-작업에 대한 우선순위 결정 -업무를 효과적으로 전달, 분배 -팀원들에게 스크럼 프랙티스 교육 -토론 참여도를 촉진 -스토리 완성에 어려움을 겪는 팀원을 도움

2. 애자일 조직

서로 밀접하게 협력이 필요한 작은 인원으로 구성된 팀이고 팀원이 여러 직능을 담당할 수 있고 스스로 책임을 지고 독립적 부서라는 특징을 기반으로 애자일 조직 구성도를 사용하며 결과 이슈에 대한 오너쉽 공유를 기대한다.



<애자일 조직 구성도>

4 WBS

단계	Task	내용	담당자	예상 기간
요구 분석	요구 추출 및 분석	기능 요구 분석	전원 참여	7일
		비기능 요구 분석		7일
	요구 모델링 추출	유스케이스 작성	유영주	4일
	요구 검증	요구 검증		3일
설계	UI 설계	와이어프레임 작성	유수연, 유영주	2일
	시스템 구조 설계	소프트웨어 인프라 관리	김혜수, 황성민	2일
	DB 설계	DB 관계 설계 및 라우터 구상	김혜수, 황성민	1일
디자인	레퍼런스 수집	레퍼런스 수집	유수연	2일
	UI 디자인 작업	와이어프레임 작성	유수연, 유영주	3일
		디자인 시안 작성	유수연, 유영주	2일

		디자인 시안 점검	전원 참여	1일
개발	회원 관리 기능	회원 가입, 로그인, 로그아웃, 회원 탈퇴	김혜수, 유수연, 황성민	1일
	스플래시 화면	로그인 여부 처리	유수연	1일
	메인 페이지	검색 기능	유수연	2일
	의원 목록 페이지	검색 기능 의원 목록 조회 정당별 필터링 기능	유수연	5일
	의원 상세 페이지	의원 상세 내용 조회	유영주	2일
	의원 통계 페이지	정당별·성별·소속위원회별 의원 통계	황성민	2일
	국회 웹사이트 연결 페이지	사이트 배너 나열	유영주	2일
	문의하기	전화 연결, 이메일 연결	김혜수, 유영주	3일
테스트	알파테스트	로컬 테스트		2주일
	베타테스트	배포환경 테스트		

5 기술관리 방법

5.1 변경 관리

1) 산출물 요구사항 관리: 매주 월요일 진행되는 정규 회의를 통해 소프트웨어의 기능적 측면에 추가하거나 수정 혹은 삭제할 부분을 검토한다. 변경 부분에 대하여 의견을 나누는 도중 의견 불일치가 일어났을 경우 익명 투표를 진행한다. 투표 결과 동률이 나올 경우 PM이 2표를 행사할 수 있도록 한다.

2) 형상 관리: 매 스크럼 회의에서 진행 상황을 확인하고, 프로젝트 진행 단계에 차질이 없도록 제품 백로그를 수정한다.

3) 문서 관리: 매주 월요일 정기 회의시간에 칸반보드(Trello)의 상태를 기록한다. 매주의 변경사항을 구글 문서에 업데이트하여 진행도를 추적하도록 한다.

** 3)해당 문서의 경우 스크럼 마스터가 작성하도록 한다.

5.2 위험 관리

1) 협업 지식의 부족

- 협업 툴인 Github 사용에 능숙하지 않아 프로젝트 형상관리에 어려움을 겪을 수 있다.
- 협업 툴 사용에 능숙한 팀원이 사용법에 대해 설명하는 시간을 갖는다. 작업 내역을 바로 main 브랜치에 반영할 경우 테스트 미실시, 코드 충돌 등의 문제가 발생할 수 있으므로 팀원 각자 branch를 만들어 작업한다. 작업 내역을 반영할 때에는 각 브랜치에서 pull request를 생성하고 코드가 요구 사항을 충족했는지 검토한 후 main에 merge한다.

2) 개인 사정으로 인하여 회의에 불참하거나 팀 규칙을 어기는 경우

- 2번 이상 위와 같은 일이 일어나는 경우 벌금을 부과하도록 한다.
- 불가피하게 회의에 불참하는 경우 최소 6시간 전 단체 채팅방을 통해 불참 사유와 함께 공유한다.

3) 연락 확인 빈도 문제로 급박한 공지 및 개발 관련 질문 등이 신속히 전달되지 않는 문제

- 디스코드 채팅의 참여 인원 '언급' 기능을 사용해 공지가 직접 전달되도록 한다. 매일 밤 12시 이전, 본 프로젝트 전용 카톡 채팅방을 확인하는 것을 팀 규칙으로 한다.
- 자정 전에 해결되어야 하는 급박한 공지사항, 의견이 있을 경우 전화, 개인 톡을 통한 다이렉트 연락을 이용한다.

4) 팀원 간 프로젝트 방향성에 대한 이해의 차이

- 일일 스크럼 회의를 통해 모든 팀원이 같은 방향성을 갖고 작업에 착수하는지 확인한다. 매일 회의가 시작되면 이전에 작성한 회의록을 상기하며 요구사항에 대한 이해의 차이를 줄이며 목표와 방향성을 재정비한다.
- 프로젝트 집중 진행 기간에 한하여 일일 스크럼 회의 후 개발 방향성에 대해 의문이 있는 경우 당일 회의를 소집할 수 있다. 의무 참여는 아니지만 가급적 팀원 모두가 참여할 수 있도록 한다.

5.3 비용 및 진도 관리

1) 비용 관리

   본 프로젝트 과정 중에는 금전적인 개발 비용을 투입하지 않아 비용 관리는 생략한다.

   ** 개발 혹은 프로젝트 진행 중 예상치 못한 비용이 발생했을 경우 PM(Project Manager)가 관리하며 영수증 처리 후 공유하도록 한다.

2) 진도 관리

매주 스프린트 계획 회의를 통해 스프린트 목표와 스프린트 백로그를 계획하고 세부적인 구현 내용과 작업자, 작업 예상 시간을 관리한다. 개발 집중 기간 동안 일일 스크럼 회의를 매일 점심시간 진행하며 각자 오늘 할 일, 어제 한 일, 개발 중 생긴 문제나 어려움 등을 간단히 공유하고 함께 해결할 수 있도록 하여 진도에 문제가 생기지 않도록 관리한다. 매 스프린트 종료 전에 스프린트 검토 회의를 진행해서 개선할 점 등에 대하여 피드백을 받는다. 스프린트 회고 회의를 통해 문제점을 확인 후 기록하며 규칙이나 표준을 잘 준수하여 진행되었는지 확인한다.

5.4 문제점 해결 방안

1) 개발 측면

각자 맡은 스프린트 백로그 진행 중 충분히 고민하고 시도했음에도 어려움이 생겼을 경우 일일 스크럼 회의 때 팀원들에게 문제에 대하여 설명한 후 시간을 정해 Discord 화상 회의를 통해 코드 리뷰 후 도움을 받도록 한다. 팀원 모두 시간적으로 여유가 없을 경우 익일 오전 내에 화상 회의를 의무적으로 진행하도록 한다.

2) 테스트 진행 시 문제 해결

테스트 진행 도중 문제가 발생할 경우 해당 부분과 관련된 개발을 맡았던 인원이 모두 모여 음성, 화상 연락을 통하여 문제점을 설명해주고 해결방안을 찾아나가는 것을 규칙으로 하며 해당 부분 개발을 맡았던 팀원이 최대한 당일 해결을 하도록 한다. 해결이 되지 않을 경우 1)의 해결 법으로 해결한다.

6 표준 및 개발 절차

6.1 개발 방법론

- 프로젝트에서 채택한 개발 방법론
  본 프로젝트에서는 팀원 간의 커뮤니케이션과 팀워크를 중점으로 프로젝트를 관리 및 변경하고 사용자의 요구사항을 만족하는 소프트웨어를 개발하는 것을 목표로 한다. 애자일 방법론(Agile)을 채택하여 짧은 기간 동안 계획과 요구 분석, 구현과 테스팅을 진행하고 빠른 상황 대처와 피드백 반영을 진행한다.
  프로젝트 진행도를 한 눈에 파악하고 기간 내에 체계적인 프로젝트 관리와 의미있는 산출물을 내기 위해 애자일 방법론 중 스크럼(Scrum) 기법과 칸반(Kanban) 기법을 병행한다.

- 개발 방법 및 절차
  1) 스프린트 단위: 스프린트(Sprint)는 1주 단위로 총 4번의 스프린트를 진행한다.
     - 1차 스프린트(10/26~11/1): 프론트・백엔드 설계, 백엔드 API 자료 확인
     - 2차 스프린트(11/2~11/6): 소프트웨어 주요 기능 구현 여부 확인, 프로토타입 산출

- 3차 스프린트(11/7~11/15): 소프트웨어 부가 기능 구현 여부 확인, 소프트웨어 테스팅
- 4차 스프린트(11/16~11/20): 소프트웨어 테스팅, 최종 완성

2) 데일리 스크럼(Daily Scrum)
팀원의 작업 진행도를 확인하기 위해 데일리 스크럼을 진행한다.
   - 진행 장소: Discord Daily Scrum 채팅 채널
   - 진행 일시: 아침 10시 ~ 낮 1시 (수업 시간을 고려)
   - 데일리 스크럼과 함께 스프린트 추적의 업무량을 작성한다.

원활한 스크럼 진행을 위해 스크럼 규칙을 규정한다. 스크럼은 스프린트 진행 기간 동안 매일 진행되며 아래의 규칙을 따른다.

> 데일리 스크럼 규칙
> 1. 지난 데일리 스크럼과 비교하여 자신이 완수한 작업
> 2. 다음 데일리 스크럼까지 자신이 완수할 작업
> 3. 현재 작업에서 문제가 되고 있는 것

3) 스프린트 추적: 제품 백로그와 스프린트 백로그를 작성하여 스프린트 동안 구현할 기능 및 산출물을 확인하고 초기 추정치를 통해 필요한 자원을 파악한다.

4) 스프린트 회고: 매주 월요일마다 스프린트 검토와 회고, 다음 스프린트 계획 회의를 진행한다.

5) 추가적인 소통: 카카오톡(KakaoTalk) 채팅방 또는 디스코드(Discord) 이용

일시	내용	비고
스프린트 기간	데일리 스크럼 진행	스프린트 진행 기간 동안 데일리 스크럼 진행 양식: '데일리 스크럼 규칙' 참고
프로젝트 회의 10/25(화) 21:00	1차 스프린트 백로그 작성 및 계획 설정	
10/26(수) - 10/31(월)	1차 스프린트 진행	
프로젝트 회의 11/1(화) 21:00	1차 스프린트 검토 및 회고 2차 스프린트 백로그 작성 및 계획 설정	
11/2(수) - 11/6(일)	2차 스프린트 진행	
프로젝트 회의 11/6(일) 21:00	2차 스프린트 검토 및 회고 3차 스프린트 백로그 작성	프로토타입 제출 마감 전 11/6(일)에 진행

	및 계획 설정	
11/7(월) - 11/14(월)	3차 스프린트 진행	
프로젝트 회의 11/15(화) 21:00	3차 스프린트 검토 및 회고 4차 스프린트 백로그 작성 및 계획 설정	
11/16(수) ~ 11/20(일)	4차 스프린트 진행	
프로젝트 회의 11/20(일) 21:00	4차 스프린트 검토 및 회고	완료보고서 제출 마감 전 11/20(일)에 진행 완료 보고서 제출 완료

7 검토 회의

7.1 검토회 일정

프로젝트 검토회는 총 5회 진행한다.

- 진행 일시
    1) 1차: 10월 16일 일요일 21:00
    2) 2차: 10월 30일 화요일 21:00
    3) 3차: 11월 6일 일요일 21:00
    4) 4차: 11월 15일 화요일 21:00
    5) 5차: 11월 20일 일요일 21:00

- 진행 장소: Discord 음성 채널(온라인)

7.2 검토회 진행 방법

검토회는 소프트웨어 생명 주기(계획 - 요구분석 - 설계 - 개발 및 테스팅)단위로 개최한다. 생명 주기마다 산출되는 결과물을 검토하여 문제 발생 범위를 예상하고 문제가 발생했을 때 오류 검출과 수정 작업을 진행한다.

단계	일시	내용
계획	10월 16일 일요일 21:00	수행계획서 검토 및 피드백
요구 분석 및 설계	10월 30일 일요일 21:00	- 요구사항 명세서 검토 - 기능 및 UI 설계 검토

개발	11월 6일 일요일 21:00	- 주요 기능에 대한 코드 리뷰 진행 - 소프트웨어 기능 피드백 - 요구사항 명세서에 작성된 기능 구현 여부 확인
개발 및 테스팅	11월 15일 화요일 21:00	- 프로토타입 구현 검토 및 피드백 - 소프트웨어 테스트 진행
완료	11월 20일 일요일 21:00	- 소프트웨어 최종 테스트 진행 - 완료 보고서 검토 - 프로젝트 회고

7.3 검토회 후속 조치

회의 종료 후 문제점 혹은 변동 상황이 있을 경우, 문제 상황 파악 후 신속하게 대처한다.

1) 변동 상황 발생의 경우
   - 디스코드 또는 카카오톡 채팅으로 변동 상황을 즉시 공유한다.
   - 변동 상황을 GitHub Commit으로 기록한다.

2) 문제 상황 발생의 경우
   - 디스코드 또는 카카오톡 채팅으로 문제 상황을 즉시 공유한다.
   - GitHub Issue에 문제 상황을 기록한다.
   - 문제 해결 시 문제 해결 방법과 해결 과정을 문서화한다.

8 개발 환경

구분			산출물	local 개발환경 / IDE / 사용 툴
SW 개발 환경	클라이언트 환경	애플리케이션	Android 앱 (API level 31)	Android Studio (Kotlin v1.6.21)
	서버 환경	웹 서버 & 웹 애플리케이션 서버	Node.js 서버 (v16.14.0) Express	Visual Studio Code
		DB 서버	MongoDB Atlas (클라우드 DB)	javascript 라이브러리 mongoose
		파일 서버	-	-
	소프트웨어 형상 관리		git 관리 이력	GitHub

프로젝트 관리 환경	진행상황 추적 및 문서화	칸반 보드	Trello
		스프린트 추적	Google Sheets
	회의 및 소통	회의 진행	Discord, 카카오톡
		회의록	Google Docs

1. SW 개발 환경

1) 클라이언트 환경
   - 일반 사용자의 정보접근성을 높인다는 기획 단계의 요구사항을 받아들여 Android 애플리케이션의 형태로 제작한다.
2) 웹 서버 & 웹 애플리케이션 서버
   - Node.js 환경의 Express 프레임워크: 클라이언트-서버 간에 json 형태로 데이터를 전송할 것을 고려하여 해당 데이터 형식을 적극적으로 사용하는 프레임워크를 채택하였다.
   - 웹 서버와 웹 애플리케이션을 통합하여 효율적으로 개발할 수 있는 이점을 활용한다.
3) DB 서버
   - MongoDB: json 형식의 Document Model을 활용한다.
   - Nosql - Document Model : 사용자별로 기입하는 정보에 차이가 있어 속성값이 null인 필드가 많아질 것으로 예상되는 경우 정보 저장에 효율적이다.
   - 기능 요구 중 DB 데이터에 접근하는 방식을 살펴보면 수정과 삭제가 빈번하지 않고 조회 성능에 무리가 없는 DB를 채택하였다.
4) 소프트웨어 형상 관리
   - 개발 인력 간 git 관리 이력을 공유하기 위해 무료 툴인 GitHub를 이용한다.
   - Organization 기능을 활용하여 작업 현황을 모든 팀원이 동일한 방식으로 (Fork 없이) 공유한다.

2. 프로젝트 관리 환경

1) 진행상황 추적 및 문서화
   - 각 작업의 진행 단계를 추적하기 위해 칸반 보드를 사용한다. 매 주 칸반보드의 형상을 기록한다. (본문 5.1의 3)참조) 칸반 보드에는 개발 작업의 마감일, 스프린트 회의 일정, 문서화 작업을 포함해 프로젝트 진행을 위한 모든 작업을 기록한다.
   - 매 스크럼 회의마다 각 담당자의 작업 현황을 스프린트 추적 시트에 반영한다. (본문 5.3의 2) 참조) 다인원 동시 접속이 가능하고 실시간으로 문서 변경사항이 공유되는 Google Sheets를 이용한다.
2) 회의 및 소통
   - 비대면 회의 진행 시 무료 온라인 음성화상채팅 툴 Discord를 이용한다.
   - 상시 온라인 메신저 카카오톡을 이용하여 연락한다.
   - 회의록은 다인원 동시 접속이 가능하고 실시간으로 문서 변경사항이 공유되며 공유 링크 내보내기가 가능한 Google Docs를 사용하여 기록한다.

9 성능 시험 방법

1) 목표

성능 시험 단계에서 앱을 모니터링하여 사용자의 애플리케이션 이용 환경을 원활하게 한다. 궁극적으로는 화면 이벤트 감지 및 반응 속도, 네트워크 통신 속도를 향상할 수 있도록 한다.

2) 성능 시험 도구

단기간의 기기 활동을 기록하여 시스템을 추적하는 여러 방법 중 Android 프로파일러 도구를 이용한 추적 방법을 채택한다. 안드로이드 앱 개발 도구인 Android Studio 버전 3.0 이상에서 지원하는 Android 프로파일러 도구를 이용하여 개발한 앱의 CPU, 메모리, 네트워크, 배터리 리소스 사용량을 추적한다.

구분	내용
CPU 프로파일러 - CPU 활동 및 트레이스 프로파일링	- UI 버벅거림 감지: 이벤트 재생 빈도가 화면 렌더링 빈도 60fps를 초과하는지 유의하여 살펴본다. - 앱 내에서 멀티 스레드를 이용하는 부분이 생기는 경우 스레드 활동 타임라인을 주시한다.
메모리 프로파일러 - 자바 힙 및 메모리 할당 프로파일링	- 끊김 현상, 멈춤, 앱 비정상 종료로 이어질 수 있는 메모리 누수 및 메모리 변동을 식별하는 목적으로 검사한다. 타임라인에 성능 문제를 일으킬 수 있는 바람직하지 않은 메모리 할당 패턴이 있는지 확인한다. - 정상적인 사용자 상호작용, 극단적인 사용자 상호작용을 가정하여 각 상황별 메모리 할당을 기록한다. 코드에서 단기간에 너무 많은 객체를 할당하거나 누출되는 객체를 할당하는 부분을 정확히 파악하도록 한다.
Network Inspector - 네트워크 트래픽 검사	- 안드로이드 앱이 외부의 서버와 통신하는 경우를 모니터링한다. (백엔드 서버의 API, 공공데이터 API 사용) - http 요청 응답 속도를 측정한다.
에너지 프로파일러 - 에너지 사용량 프로파일링	- 기기의 각 리소스의 에너지 소비량을 모니터링한다.

10 문서화

회의록

회의일시	2020년 01월 20일	작성자	홍길동
참석자	관리팀 5명 / 영업팀 5명 / 개발팀 5명		

회의안건	회의주제서술

	내용	이슈 / 비고
회의내용	1. 회의안건1 - 내용1 - 내용2	이슈나 비고 입력
	2. 회의안건2 - 내용1 - 내용2 - 내용3	이슈나 비고 입력
	3. 회의안건3 - 내용1	이슈나 비고 입력
	4. 회의안건4 - 내용1	

	내용	진행일정
결정사항		
다음회의	다음회의 날짜와 장소 입력	

<회의록 양식>

				월	10월							11월																	
				일	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
우선순위	제품 백로그	스프린트 백로그	담당자	초기 추정치	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

<스프린트 추적 시트 양식>

요구사항 정의서 (Requirment Definition)

*일반적으로는 메뉴그룹별, 화면 메뉴별로 요구사항 ID를 정의함
*요구사항의 수정, 추가, 삭제 사항을 상세히 기록하고, 요구사항 수정은 항상 프로젝트 변경관리 프로세스에 의하여 진행 / 수정 내용 앞에는 날짜를 기록함

시스템 (Application)	업무그룹	요구사항 ID	요구사항명	요청자 (요구사항 Owner)	상태	최초요청일자	최종수정일자	요청목적(선택)	기능 요구사항	프로세스 요구사항	화면 요구사항	보안 요구사항	성능 및 용량 요구사항	데이터 요구사항	기타 요구사항
모바일쇼핑	로그인	REQ_001	로그인 화면_ID/Passwd 인증, 생체인증	홍길동	수정	2020-01-02	2020-02-01		*ID패스워드 Validation 체크 후 로그인 허용 *숫자,문자,특수문자 8자이상 조합 2/1 생체인증(홍채,지문) 요구 추		*별도 정부된 디자인 참조	*패스워드 마스킹 처리 *1초당 1회이상 시도시 접속 차단 *로그인 Activity 정보 저장			SSO(Single Sign On)를 위해 XXX와 연동
	검색	REQ_002	검색화면_상단 검색어 입력 기능		최초	2020-01-02	2020-01-15								
		REQ_003	검색화면_최근 검색어 및 추천 검색어 영역		추가	2020-01-15	2020-01-15								
		~~REQ_004~~	~~검색화면_최대 조회수 영역~~		~~삭제~~	~~2020-01-02~~	~~2020-01-15~~								
		REQ_005	검색 화면_검색 결과 화면		최초								검색결과 2초 이내 노출		

<요구사항 정의서 양식>

요구사항 추적매트릭스 (요구사항 추적표, Requirement Traceability Matrix)

*요구사항과 분석/설계 산출물을 ID를 명시하여 연결함(ID를 이용하여 추적 가능하도록 함)
*통합 및 사용자 테스트를 포함한 각 분석설계 산출물에는 요구사항 ID가 항상 매핑이 되어 쌍방향 추적성이 보장되어야 함
*설계/분석 요소에 따라 필요한 컬럼을 추가하여 사용하고, 필요한 경우에는 관련 책임자를 함께 명시할 수 있음
*ID 체계는 프로젝트 초반에 표준문서에 정의하여 따름

요구사항 ID	요구사항명	화면 ID	프로그램 ID (Code명)	테이블 ID	인터페이스 ID	통합테스트 ID	사용자테스트 ID
REQ_001	로그인 화면_ID/Passwd 인증, 생체인증	SLO_001 SLO_002	Login.java	TB_003			
REQ_002	검색 화면_상단 검색어 입력 기능	S_SR_003	SearchInput.java	TB_012	IF_003		
REQ_003	검색화면_최근 검색어 및 추천 검색어 영역	S_SR_005	Search.java	TB_201 TB_303			
REQ_005	검색 화면_검색 결과 화면	S_SR_006			IF_008		

<요구사항 추적 매트릭스 양식>

No	대분류	화면번호	소분류	내용	추가내용	결과	개발자 확인	비고
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

<테스트 양식>

관리번호	발생일자	리스크 제목	리스크 내용	해결 방법	비고

<리스크 관리대장 양식>

11 유지보수

1) 진행 일시: 월 말 1회 진행

2) 진행 절차

유지보수는 다음과 같은 절차로 진행된다.

1) 현재 프로그램의 이해
프로그램 기능과 구조를 변경하기 전, 현재 소프트웨어의 상태를 파악하기 위해 프로그램의 로직을 추적하고 사용자의 요구사항과 설계 등을 상기시킨다.

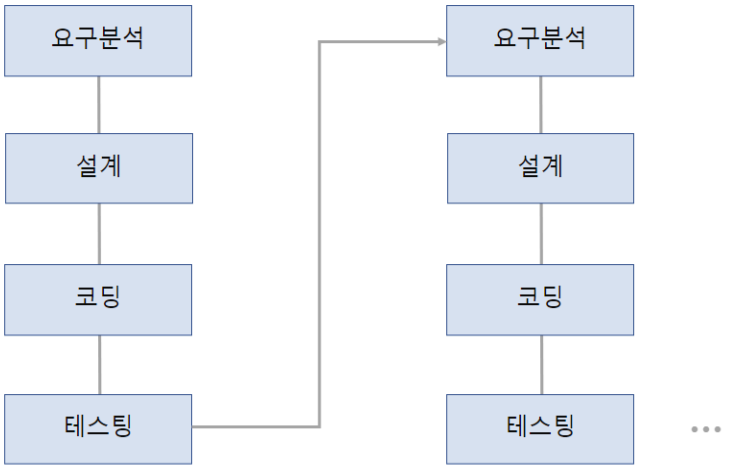
2) 변경 내용 파악 및 분석
GitHub Repository 또는 문서를 확인하여 소프트웨어에서 필요한 변경 내용을 파악하고 변경에 필요한 비용과 인력, 리스크를 분석한다.

3) 변경 영향 파악
소프트웨어 기능 간 의존 관계를 파악하여 기능 변경으로 인해 받을 영향을 파악하고 이해관계자들에게 변경 내용을 알리고 피드백을 얻는다.

4) 변경 구현과 테스팅, 설치
소프트웨어 변경을 진행하고 필요 시 소프트웨어 설계를 수정한다.

3) 진행 내용

- 유지보수 모델



<반복적 개선 모델>

유지보수 프로세스는 반복적 개선 모델을 채택한다.
본 프로젝트는 의뢰자의 요구를 확인하고 전체 생명주기 단위로 유지보수를 진행한다. 프로젝트 변경의 필요성이 생길 때마다 요구사항 분석, 설계, 구현을 반복하여 시스템을 개선한다.

- 유지보수 유형별 진행 방법

유지보수 유형	수행 내용

완전형 유지보수	1) 성능 관리: 사용자의 요구 변화에 따라 소프트웨어의 기능 변경 및 추가 작업 진행 2) 기능 보강: 의뢰자의 요구 변화에 따른 시스템의 기능 변경 및 보강 3) 코드 효율 향상 - 깃허브 이슈 및 커밋 기록 확인, GitHub Repository 코드 형상 관리 - 정기적으로 소프트웨어 점검 실시, 문제 상황이 발생했을 때 GitHub에 공유한다.
적응형 유지보수	소프트웨어가 개발 환경 이외 변경된 환경에서도 잘 실행되는지 확인한다.
수정형 유지보수	소프트웨어 설치 후 결함이 발견되면 해당 코드의 수정 작업 진행
예방형 유지보수	코드 리팩토링: 소프트웨어의 복잡성을 줄이고 코드의 구조를 단순화하여 프로그램의 이해도와 유지보수성을 향상시킨다.

- 기능의 의존관계 확인

- 형상 관리
  1) 베이스라인 정의

SW 개발 주기	베이스라인	항목
계획	계획 베이스라인	수행계획서
		의뢰자 요구 사항
		스크럼 일정 및 절차 문서
		비용 예산서
요구사항 분석	요구 베이스라인	요구사항 분석서
		유스케이스
		요구사항 정의서
		요구사항 추적 매트릭스
설계	설계 베이스라인	SW 설계 문서
		테스트 계획
구현	구현 베이스라인	원시 코드
테스트	테스트 베이스라인	단위 테스트 케이스

		테스트 케이스 결과
		사용자 매뉴얼, 설치 매뉴얼

2) 형상 관리 절차

절차	수행 내용
형상 파악	- 개발 프로세스에 따른 베이스라인 파악 - 각 베이스라인의 형상 항목 정의 - 형상 항목 확인 및 분류
형상 변경 제어	- 형상 변경 이유 파악 - 변경 영향 분석: 형상 항목에 대한 변경 파악, 관련 부서가 변경에 대한 영향 분석 및 보고 - 변경 계획 수립: 변경 계획서 작성 - 변경 계획 평가: 변경 계획에 대한 평가 진행 - 변경 추가
형상 감사	- 형상 항목 검토 - 형상 항목 확인
형상 상태 보관	형상 항목에 대한 정보 추적 및 보고

3) 형상 관리 툴

- 원시 코드: Git(분산형 버전 제어 시스템)
- 문서: Google Drive

12 설치, 인수

개발 의뢰자의 요구를 따라 프로젝트 계약에 명시된 기준에 대하여 개발 시스템의 환경에서 사용 용이성, 기능성, 성능을 테스트하는 과정을 알파테스트와 베타테스트의 단계로 나누어 진행한다.

1) 알파테스트

- 테스트 목적: 기능성, 유용성 테스트
- 시행시기: 시스템 테스트 이후~ 제품이 70 -90 % 완료 시
- 시행기간: 테스트 주기는 1~2주 동안 지속하여 발견 된 문제 수와 추가된 새 기능 수에 따라 변동
- 이해 관계자 및 참여자: 엔지니어(개발자), 품질 보증 및 제품 관리 담당자, 기술 전문가
- 알파 테스트 시행을 위한 체크리스트 항목
  1) 비즈니스 요구 사항에 맞게 설계 및 검토된 알파 테스트 대상

2) 모든 알파 테스트와 요구 사항 사이의 추적 시나리오 작성
3) 도메인 및 제품에 대한 지식이 있는 테스트 팀 인력
4) 실행을 위한 환경 설정 및 빌드
5) 도구 설정은 버그 로깅 및 테스트 관리를 위한 준비
6) 시스템 테스트는 사인 오프

2) 베타 테스트:

- 테스트 목적: 기능성, 유용성, 신뢰성, 보안 테스트

- 시행 시기: 알파 테스트 완료 직후 ~ 제품이 시장에 출시되기 전 수행

- 시행기간: 4~6주마다 1~2번의 테스트주기로 새로운 기능이 추가되거나 핵심 구성 요소가 수정된 경우에만 확장

- 제품 완성도: 90~95% 완료 될 것으로 예상 - 모든 플랫폼에서 충분히 안정적이며 모든 기능이 거의 완료된 상태

- 이해 관계자 및 참여자: 제품 관리 담당자, 품질 관리 담당자, 제품을 실제로 사용하고자하는 최종 사용자 / 실제 사용자는 참가자

- 베타 테스트 시행을 위한 체크리스트 항목
    1) 제품의 모든 구성 요소 테스트 준비.
    2) 최종 사용자에게 전달되어야하는 문서(설정, 설치, 사용, 제거 등)는 준비
    3) 제품 관리 팀은 각각의 모든 주요 기능이 양호한 작동 상태인지 검토 완료.
    4) 버그, 피드백 등을 수집하는 절차를 식별하고 검토 후 게시
    5) 알파 테스트는 결과 공개
    6) 제품의 베타 버전 출시
    7) 사용자 설명서, 알려진 문제 목록은 문서화
    8) 버그를 포착하기위한 도구, 피드백 준비, 사용 설명서 게시

3) 사용성 테스트

- 테스트 목적: 사용성 테스트, 학습 용이성을 우선순위로 둔다. 유지보수 후 사용자의 시스템 사용을 탐지할 목적으로, 수행한 테스트 데이터와 관찰결과를 분석하여 제품의 사용성 확인
- 시행 시기: (제품 완성도 100%) 시스템 배포 후, 모든 플랫폼 환경에서 실행
- 사용성 테스트를 위한 체크리스트 항목
    1) 사용성 테스트의 목적과 목표 설정
    2) 사용성의 여러 측면(학습 용이성, 오류율, 예측성, 만족) 중 상대적인 우선순위 선정
    3) 테스터가 수행할 태스크, 테스트 시나리오 작성
    4) 대표 사용자 샘플 선정
    5) 사용자 인터뷰 질문 목록 작성 및 인터뷰 양식 제작

13 참고문헌 및 부록

1-1) 울산제일일보, 새내기 유권자 투표참여정치 관심도 설문조사, 2022년10월30일 접속, http://www.ujeil.com/news/articleView.html?idxno=494

3) 정보통신산업진흥원, SW R&D 품질검증기준 개발용역 산업분야 – 홈네트워크/정보가전[001] 프로젝트 계획서 작성 가이드,2012. 11,7쪽~8쪽
3-2) “애자일 개발의 지휘자“ 스크럼 마스터의 이해, 2022년 11월 12일 접속 https://www.itworld.co.kr/news/107474

7-1) “walkthrough”, 2022년10월15일 접속, https://mj0316.tistory.com/m/entry/walkthrough

7-2) “소프트웨어 검사” 2022년10월15일 접속, https://many258.github.io/study/software-engineering-testing/

9-1) “앱 성능 측정 개요”, 안드로이드 개발자 문서, 2022년10월16일 접속, https://developer.android.com/studio/profile/measuring-performance?hl=ko

9-2) “앱 성능 프로파일링”, 안드로이드 개발자 문서, 2022년10월16일 접속, https://developer.android.com/studio/profile?hl=ko

9-3) “Android 프로파일러”, 안드로이드 개발자 문서, 2022년10월16일 접속, https://developer.android.com/studio/profile/android-profiler?hl=ko

9-4) 백중원(Leopold), “[Android] Android 네트워크 트래픽 디버깅 (by Stetho)”, Medium, 2022년10월16일 접속, https://medium.com/@joongwon/android-네트워크-트래픽-디버깅-bfce67956a56

10) 예스폼, “경영리스크 관리대장”, 2022년12월18일 접속, https://m.yesform.com/doc/form/404413

12-1) ”알파 테스트 및 베타테스트”2022.10.15접속 https://ko.myservername.com/alpha-testing-beta-testing

12-2) ”베타 테스트 [beta test] (IT용어사전, 한국정보통신기술협회)”,2022.10.16접속 https://terms.naver.com/entry.naver?docId=1919607&cid=50372&categoryId=50372

12-3) ”갤럭시폴더로 보는 알파테스트,베타테스트,필드테스트 정의”,2022.10.15접속 https://m.blog.naver.com/donrock1029/221552751125

12-4) 최은만. 『소프트웨어공학의모든것』. 생능출판, 2020