Blood Link 최종보고서

SOFTWARE ENGINEERING FINAL REPORT

-Blood Link-

**TEAM** **소공소공**

**LEADER** 컴퓨터학부 2012105038

박순렬

**Member** 컴퓨터학부 2012105091

조영래

컴퓨터학부 2014105038

박준현

컴퓨터학부 2015110453

김민지

컴퓨터학부 2015116542

이한솔

* 목차 (Contents) -

1. **요구사항 반영 분석**
   1. 기존 요구사항
      1. 기능 요구사항(Functional Requirements)
      2. 시스템 기능 요구사항 (Non-Functional Requirements)
   2. 추가 및 변경된 요구사항
2. **RISK 분석**
   1. 기 도출 Risk
   2. 발생 Risk
3. **시스템 기능**
4. **기술적으로 어려웠던 부분**
5. **참고문헌**

부록 A. **팀원 기여도**

부록 B. **회의록**

* 1. **요구사항 반영 분석**

1. 기존 요구사항

1.1.1. 기능 요구사항(Functional User Requirements)

|  |  |  |
| --- | --- | --- |
| **ID** | **요구사항** | **반영여부** |
| F-ManC-1 | 헌혈증서 등록 | 일부 반영 : 1-1 생략, 1-2 정보입력으로 대체 |
| F-ManC-2 | 헌혈증서 승인 | 일부 반영 : 유효성 확인 생략 |
| F-ManC-3 | 헌혈증서 조회 | 전체 반영 |
| F-DonC-1 | 헌혈증서 요청 | 전체 반영 |
| F-DonC-2 | 헌혈증서 전달 | 일부 반영 : 적립기능생략 |
| F-DonC-3 | 헌혈증서 펀딩 | 일부 반영 : 등급적립생략 |
| F-DonC-4 | 헌혈증서 선물 | 전체 반영 |
| F-UseC-1 | 헌혈증서 제출 | 전체 반영 |
| F-UseC-2 | 헌혈증서 접수 | 전체 반영 |
| F-UseC-3 | 헌혈증서 거부 | 전체 반영 |
| F-Manl-1 | 회원가입 | 전체 반영 |
| F-Manl-2 | 로그인 | 전체 반영 |
| F-Manl-3 | 회원정보 조회 | 전체 반영 |
| F-Manl-4 | 회원정보 수정 | 전체 반영 |
| F-Manl-5 | 회원 탈퇴 | 미반영 |

* + 1. 비기능 요구사항 (Non-Functional Requirements)

|  |  |  |
| --- | --- | --- |
| **요구사항 ID** | **Description** | **comment** |
| **N-AD-1** | **안드로이드 5.1 Lollipop 이상에서 구동** | **안드로이드 5.1 Lollipop 환경에서 개발하여 그 이상의 버전에서도 구동 가능하다** |
| **N-CD-1** | **헌혈증서 등록 시 반드시 관리자를 통한 등록절차를 거친다** | **사용자가 헌혈증서의 등록을 요청하면 관리자가 유효여부를 따진 후 승인 또는 반려한다.** |
| **N-DM-1** | **관리자가 사용자, 의료기관 정보를 관리하며 접근을 제한한다.** | **관리자가 DB에 접근하여 정보를 관리한다. 사용자와 의료기관은 웹서버를 이용하여 관리자에게 정보의 제공을 요청한다.** |
| **N-RT-2** | **헌혈증서 등록 승인 처리시간은 24시간이 넘지 않도록 한다.** | **헌혈증서 등록 승인은 신청 즉시 처리하며 반려 또는 승인까지 10분이 넘지 않는다.** |

2. 추가 및 변경 요구사항

|  |  |
| --- | --- |
| Name :ID | 헌혈증서 정보 유지: N-CD-2 |
| Description | 이 요구사항은 헌혈증서의 신뢰성을 보장하기 위한 것이다. |
| Inputs |  |
| outputs |  |
| Rationale | 이전에 등록되었던 헌혈증서를 삭제하지 않고 시스템상에서 중복된 등록 등을 조사할 수 있도록 한다. |
| Action | 헌혈증서는 등록만 가능하고 기증처의 등록, 소유자 정보 수정 외의 다른 정보에 대해서는 수정이 불가능하고, 등록된 헌혈증서는 비활성화를 이용하여 삭제를 대신한다. |
| references | - 연관된 use case ID : U-ManC, U-UseC, U-DonC  - 연관된 다른 functional/nonfunctional requirements ID : F-ManC-1, F-ManC2, N-CD-3 |

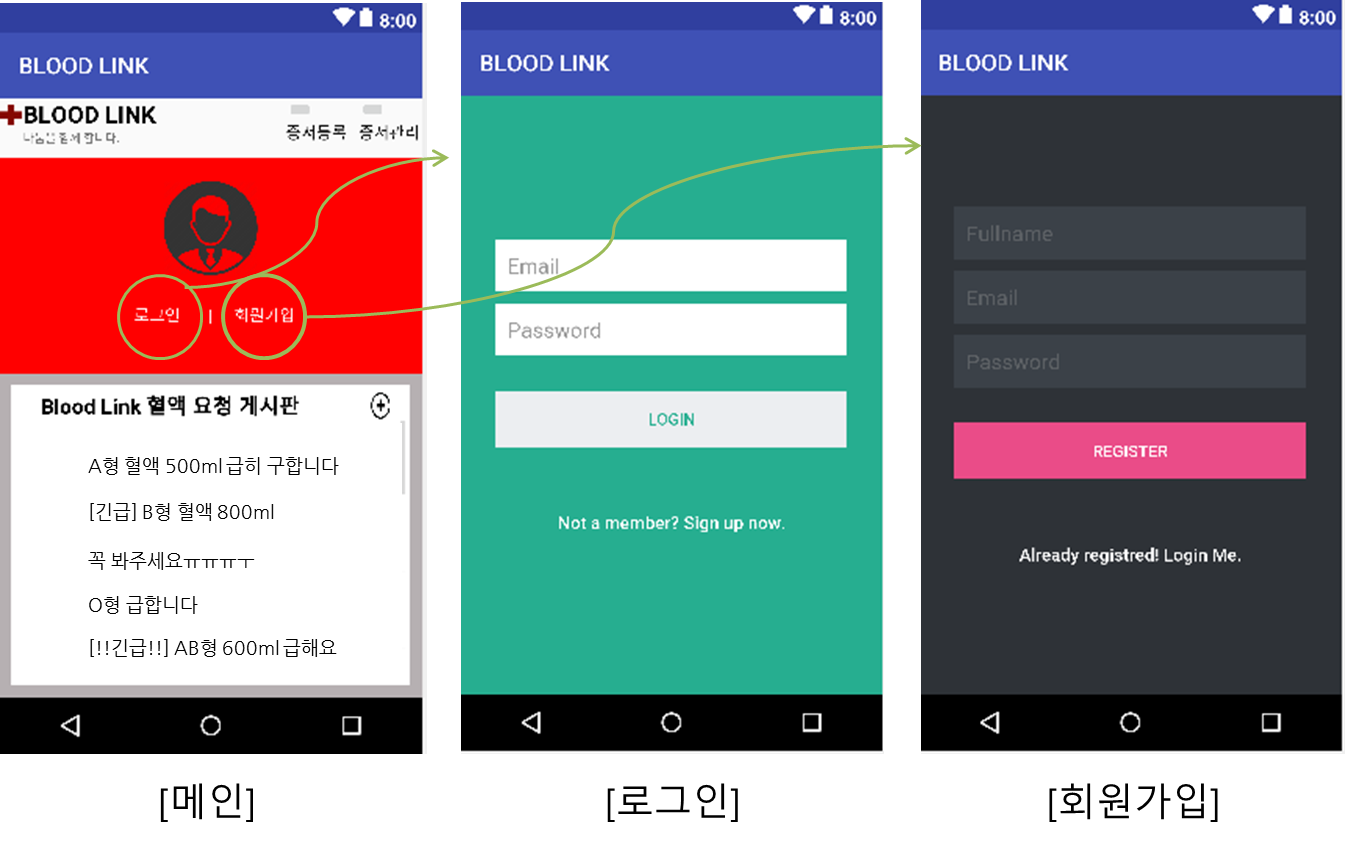
1. **Risk 분석**
   1. 기 도출 Risk

|  |  |  |
| --- | --- | --- |
| 위험(Risk) | 발생여부 | 대처 및 결과 |
| 데이터 무결성 위배 (ex. 헌혈증서) | X |  |
| 취업, 질병 등에 따른 팀원 이탈 문제 | O | 취업으로 인한 팀원이탈(조영래, 6/4~6/12)  카카오톡으로 회의를 진행, 지속적으로 소통하고 개발 내용을 공유하여 해결 |
| 헌혈, 헌혈증서 서비스에서 윤리적 문제 발생 | x |  |
| 모바일 앱, 웹 관련 개발 경험 부족 | O | 팀원의 Java App, Android App 개발 경험이 부족하여 일정 및 구현에 차질 발생  부족한 부분에 대한 학습과 협력을 통해 보완 |
| 고학년으로 구성된 팀원들의 일정 관리 | O | 팀원 대다수가 4학년으로 구성되어 취업, 타 학업 등을 수행하여 일정 관리에 난항  팀장 주도하에 짧은 시간 밀도있는 회의를 진행하고 카카오톡, github 등에서 추가 논의를 하며 일정을 보완 |
| 외부 환경에 따른 팀원의 동기 결여 | O | 학업과 다수의 프로젝트를 수행하는 과정에서 본 프로젝트에 대한 동기결여와 집중도가 낮아진 시기가 있었음  이를 극복하기 위해 팀원 교류와 회식을 통해 팀워크와 의욕을 북돋움 |
| HW/SW 상의 문제, 손상, 고장 등이 발생 | O | 안드로이드 개발을 수행하며 낮은 노트북 성능으로 인해 지속적으로 오류가 발생하여 개발에 난항을 겪음  문제의 원인을 온라인 검색을 통해 해결하고 타 기기를 사용하여 고장에 대처 |
| 의견 대립, 감정 소모 등의 팀 내 갈등 | X |  |
| 프로젝트 명세에 필요한 수치 미산정 | X |  |
| 기존 유사 서비스의 존재 (ex. 스마트헌혈) | O | 혈액관리본부의 스마트 헌혈 어플리케이션이 존재하며 유사 서비스를 제공  스마트 헌혈 어플리케이션에서 제공하는 서비스와 차별화를 두기 위해 헌혈증서에 대한 관리, 사용, 기부에 초점 |
| 미숙한 사용으로 비효율적인 자원이 발생 | O | Test와 Deploy를 위한 환경이 비효율(Heroku 상에 서버 호스팅, local database 포트포워딩, 안드로이드 가상 머신 사용 등)  제한된 자원에서 최대한 효율성을 끌어내기 위해 설계를 재조정 |
| 요구명세 과정에서 대상에 대한 막연한 산정 | X |  |
| 팀의 개발역량에 대한 수치화가 불가능 | x |  |
| 상황에 따라 팀원의 역할이 급변할 가능성 | X |  |
| 프로젝트 평가자의 팀에 대한 인식이 부정적 | x |  |

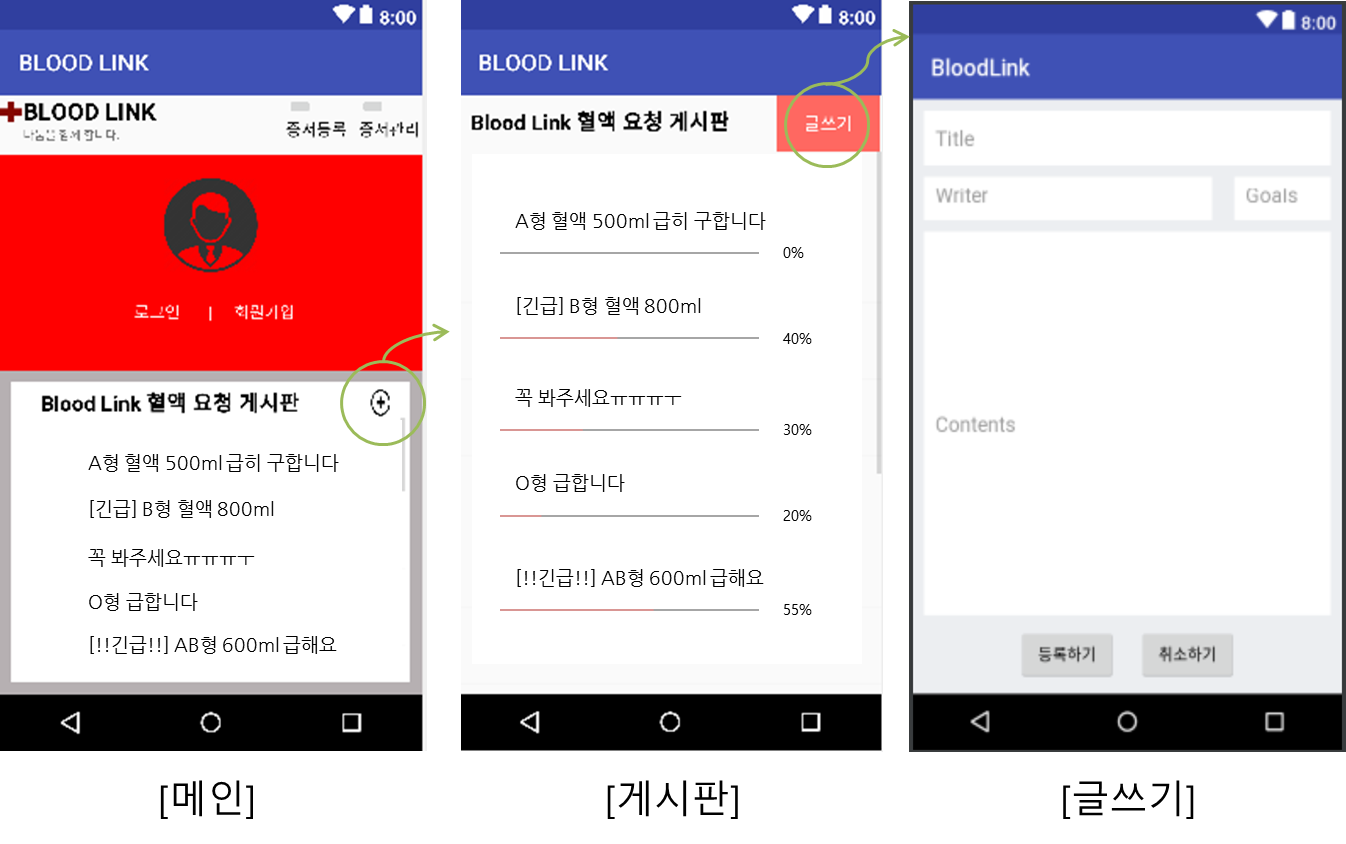
2. 발생 Risk

|  |  |  |
| --- | --- | --- |
| 위험(Risk) | 유형 | 대처 및 결과 |
| 회의 효율성 저하 | 조직, 사람 | 회의 과정에서 잡담, 지각 등의 저해 요인들로 인해 불필요하게 회의시간이 길어지고 회의집중도와 효율이 떨어지는 문제가 발생  회의 목표와 제한 시간을 정하여 집중도를 높일 수 있도록 유도하여 문제를 해결 |

1. **시스템 기능**
   * 회원가입 및 로그인



* + - 처음 어플리케이션을 켜면 위와 메인화면이 나옴.
    - 이 때 로그인 버튼을 누를 시 로그인 화면으로 전환.
    - 가입이 되어 있다면 본인의 아이디(이메일)과 비밀번호를 입력 후 LOGIN 버튼을 눌러 어플 사용.
    - 이 때 ‘BloodLink’ 서버의 DB에 해당 사용자 정보가 저장이 되어 있는 경우 로그인이 성공하고 다시 메인화면으로 돌아감.
    - 회원이 아니라면 회원가입과정 진행 가능.
    - 메인화면에서 회원가입을 버튼을 누르고 회원가입 화면에서 본인의 아이디, 비밀번호, 이름을 입력하고 REGISTER 버튼을 누르면 서버로 해당 정보가 전송이 되고 이미 등록된 회원이 아닌 경우 새로 DB에 등록.
    - 등록이 된 후 해당 페이지는 로그인 화면으로 전환이 되고 등록한 아이디와 비밀번호로 로그인을 진행 가능하게 함
  + 헌혈증서 기증 게시판

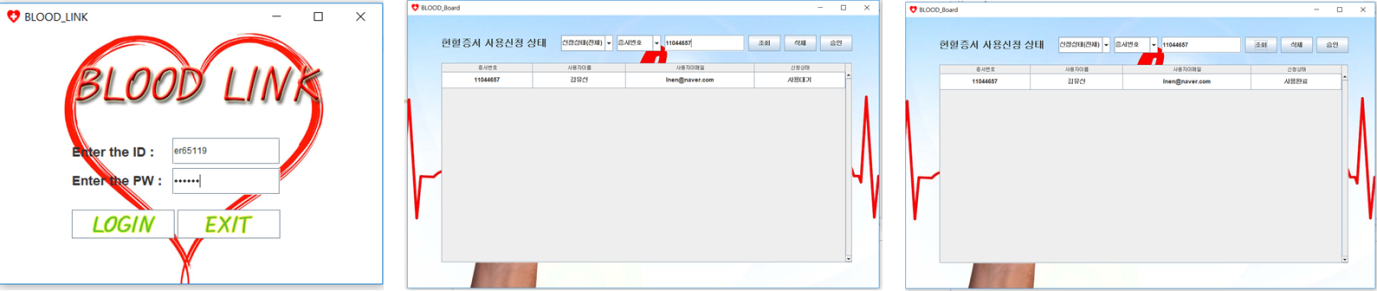


* + - 메인화면에서 보이는 게시판 옆의 ‘+’버튼을 클릭하면 게시판 페이지로 전환
    - 게시판 페이지에서 현재 등록되어 있는 게시글과 그 게시글에서 필요로 하는 혈액 달성량을 progress bar로 보여줌.
    - 이 때 100% 충족되면 해당 글은 자동으로 닫히게 되고 사용자의 눈에 보이지 않게 됨.
    - 또한, 게시글 작성자가 임의로 글을 닫을 수 있음..
    - 게시판 페이지에서 글쓰기 버튼을 누르면 게시글을 쓸 수 있는 글쓰기 화면으로 전환.
    - 제목, 작성자, 목표 혈액량, 혈액이 필요한 이유를 작성할 수 있는 페이지로 구성.
    - 해당 내용을 작성 후 ‘등록하기’버튼 클릭하면 게시판으로 돌아가고 등록된 게시글 확인 가능.
  + 증서 관리 및 등록



* + - 메인화면 상단의 증서 등록 버튼 클릭 시 사용자가 헌혈 증서를 등록할 수 있는 페이지로 전환.
    - 해당 페이지에서 증서의 정보를 입력 후 등록 버튼 클릭 시 서버로 전송이 되고 승인이 난 후 DB에 저장이 됨.
    - 등록된 증서는 증서 관리 페이지에서 조회 가능.
    - 메인에서 증서 관리 버튼 클릭 시 증서관리 페이지로 화면 전환이 되고 해당 페이지에서 사용자의 보유 증서 확인 가능.
  + 증서 사용





* + - 증서관리 페이지에서 증서 사용하기 버튼을 클릭 시 증서사용페이지로 전환.
    - 병원에서 병원관계자가 ‘BloodLink’에 로그인.
    - 로그인 정보는 서버로 넘어가고 병원 관계자 DB의 내용에 있는 경우 로그인 성공.
    - 병원에서 증서를 사용하려는 사람은 ‘BloodLink’ 어플리케이션에서 증서사용페이지를 들어가고 병원명 검색 후 사용할 증서를 선택하면 서버로 해당 정보가 넘어간다.
    - 병원에서는 해당 증서 번호를 입력하여 조회.
    - 해당 증서가 있고 사용하려는 경우 클릭 후 승인 버튼 클릭.
    - 승인이 되면 증서의 신청상태가 사용완료로 전환되고 서버의 DB에서도 사용이 완료된 증서임을 표기.
    - 사용자의 ‘BloodLink’어플리케이션에서는 해당 증서가 사라져 재사용 할 수 없도록 조치함.

1. **기술적으로 어려웠던 부분**

* Android, Java App Client와 Spring RESTful API 서버 간의 통신 인터페이스에서 HttpURLConnection을 이용하여 HTTP 통신을 구현하는 것이 필요하였는데 클라이언트 개발자의 이해부족으로 Request URL을 구성하고 Response Body를 String으로 읽을 때 어려움이 있었음
* IntelliJ 또는 Android Studio를 이용하여 Android App을 개발하는 과정에서 종종 알 수 없는 오류 및 에러(ex. AAPT2 error: check logs for details, Unsupported method: BaseConfig.getApplicationIdSuffix())가 발생하여 이를 해결하는데 난관을 겪음, Android Manifest 구성이나 Gradle 프로젝트 동기화 과정에서 충돌이 일어나거나 빌드 과정에서 오류가 발생할 때가 많아 원인을 찾을 때 개발한 코드가 아닌 프로젝트 속성을 검토해야하는 어려움이 있었음
* Spring Boot와 MySQL을 연동하는 과정에서 JPA를 통해 데이터 처리가 이루어지도록 구현하였는데 Java의 클래스 객체를 MySQL에 데이터화 할 때 클래스 변수와 테이블 속성이 Mapping되지 않는 문제가 발생하여 정상 작동하지 못함
* Spring Boot에서 사용자에 대한 Model을 정의할 때 User 클래스를 Donor(헌혈 기증자, 일반 사용자), MI(의료기관, 증서 사용처) 클래스가 상속받도록 객체를 구성하였음, CRUDRepository를 통해 DB에서 상속된 클래스에 접근하거나 추상 클래스, 추상 메서드를 통해서 상속 Controller를 구현할 때 Polymorphism이 의도한대로 이루어지지 않아 코드 중복이 필요한 문제가 있었음
* 데이터를 요청할 때 사용자와 헌혈증서 DB 테이블의 Join 관계를 구현할 필요가 있었는데 JPA를 통해서 객체의 Join관계를 표현하고자 할 때 Mapping과정에서 어려움이 발생

1. **참고문헌**

* Software Engineering 10 Edition, Sommerville, Addison-Wesley
* 난 정말 JAVA를 공부한 적이 없다구요, 윤성우
* Do it! 안드로이드 앱 프로그래밍, 정재곤
* 안드로이드 프로그래밍 정복, 김상형
* 기초부터 작품제작까지 안드로이드 프로그래밍
* Spring Project Official Document(http://spring.io)

**부록 A. 팀원 기여도**

|  |  |  |  |
| --- | --- | --- | --- |
| 팀원 | 역할 | 기여내용 | 기여도 |
| 박순렬 | 팀장, 서버 개발 | 프로젝트 총괄  프로젝트 일정, 개발, 산출물 관리  RESTful 서버 구조 구성  Web Application Logic(Business Logic) 개발  REST API Controller 구현, Spring Boot 구축 | **30%** |
| 조영래 | 안드로이드 개발 | UI/UX 개발, 안드로이드 앱 디자인  회원, 게시판 관련 기능 개발 | **10%** |
| 박준현 | JAVA 개발 | JAVA 앱 개발  의료기관 관련 기능 개발  Java Swing UI 개발 | **20%** |
| 김민지 | 안드로이드 개발 | 안드로이드 앱 개발, API 연동  헌혈증서 관련 기능 개발  API 데이터 파싱  트랜잭션, View 처리 | **20%** |
| 이한솔 | 서버 개발 | 웹서버, DB 구축  Model, CRUD 기능 개발  Spring Framework 구성, Deploy 관리 | **20%** |

**부록 B. 회의록**

<다음 장 첨부>