

# 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) -

### I. 요구사항 반영 분석

#### 1. 기존 요구사항

##### 1.1.1 기능 요구사항(Functional Requirements)

##### 1.1.2 시스템 기능 요구사항 (Non-Functional Requirements)

#### 2. 추가 및 변경된 요구사항

### II. RISK 분석

#### 1. 기 도출 Risk

#### 2. 발생 Risk

### III. 시스템 기능

### IV. 기술적으로 어려웠던 부분

### V. 참고문헌

#### 부록 A. 팀원 기여도

#### 부록 B. 회의록

## I. 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-ManI-1	회원가입	전체 반영
F-ManI-2	로그인	전체 반영
F-ManI-3	회원정보 조회	전체 반영
F-ManI-4	회원정보 수정	전체 반영
F-ManI-5	회원 탈퇴	미반영

#### 1.1.2. 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

## II. Risk 분석

### 1. 기 도출 Risk

위험(Risk)	발생 여부	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 낮아

		진 시기가 있었음 이를 극복하기 위해 팀원 교류와 회식을 통해 팀워크와 의욕을 복돋움
HW/SW 상의 문제, 손상, 고장 등 이 발생	O	안드로이드 개발을 수행하며 낮은 노트북 성능 으로 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 미산 정	X	
기존 유사 서비스의 존재 (ex. 스마 트헌혈)	O	혈액관리본부의 스마트 헌혈 어플리케이션이 존재하며 유사 서비스를 제공 스마트 헌혈 어플리케이션에서 제공하는 서비 스와 차별화를 두기 위해 헌혈증서에 대한 관 리, 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 위 해 설계를 재조정
요구명세 과정에서 대상에 대한 막 연한 산정	X	
팀의 개발역량에 대한 수치화가 불 가능	x	
상황에 따라 팀원의 역할이 급변할 가능성	X	
프로젝트 평가자의 팀에 대한 인식 이 부정적	x	

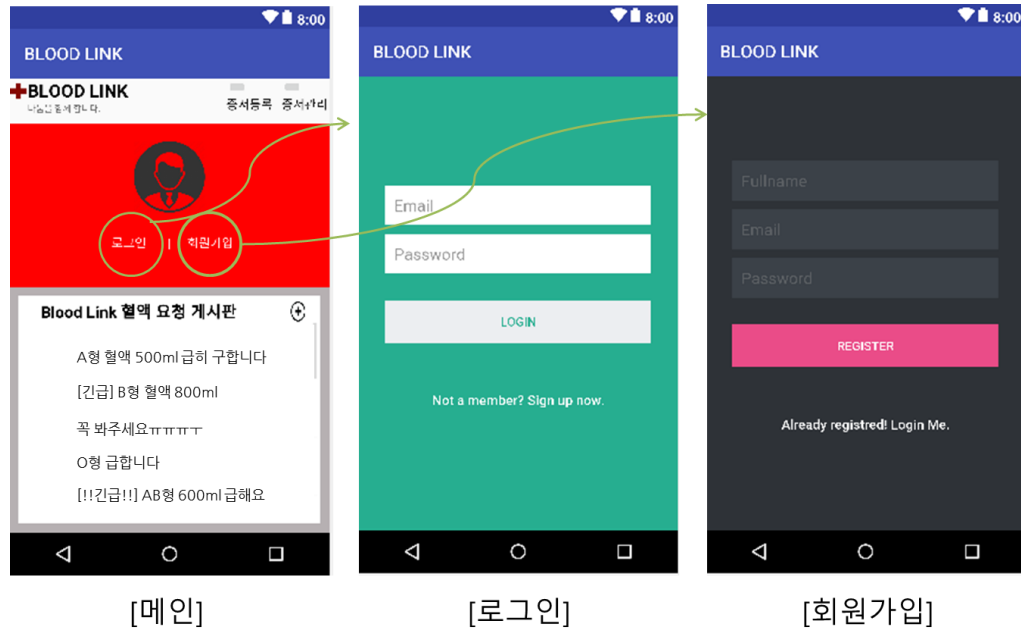
## 2. 발생 Risk

위험(Risk)	유형	대처 및 결과
회의 효율성 저하	조직, 사람	회의 과정에서 잡담, 지각 등의 저해 요인들로 인해 불필요하게 회의시간이 길어지고 회의집 중도와 효율이 떨어지는 문제가 발생

		회의 목표와 제한 시간을 정하여 집중도를 높일 수 있도록 유도하여 문제를 해결
--	--	---

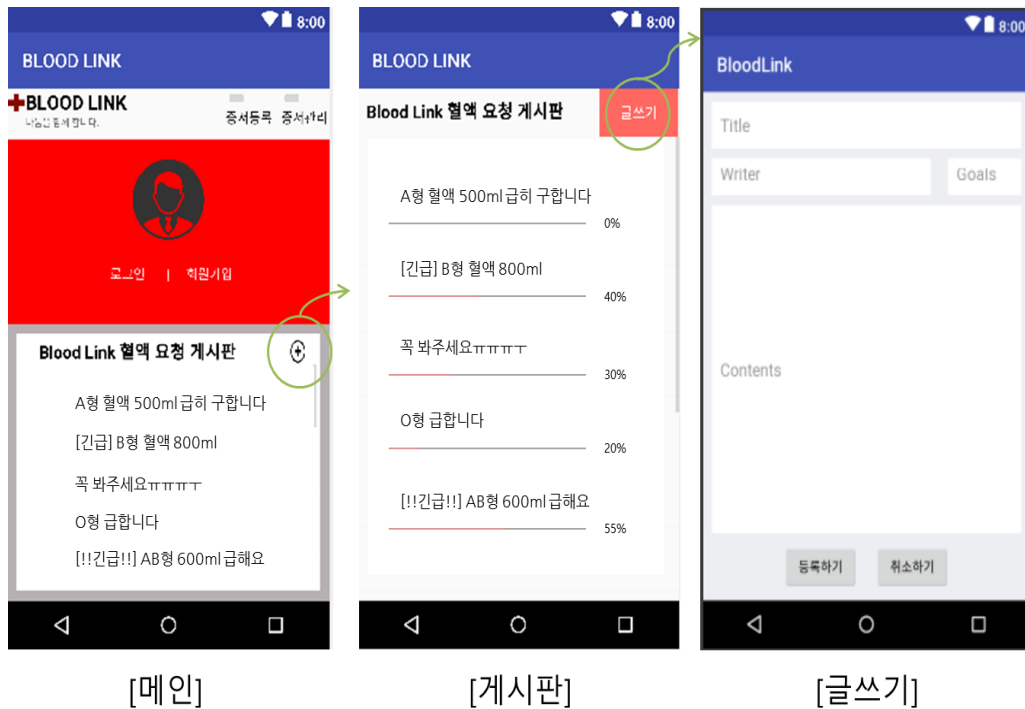
### III. 시스템 기능

#### ■ 회원가입 및 로그인



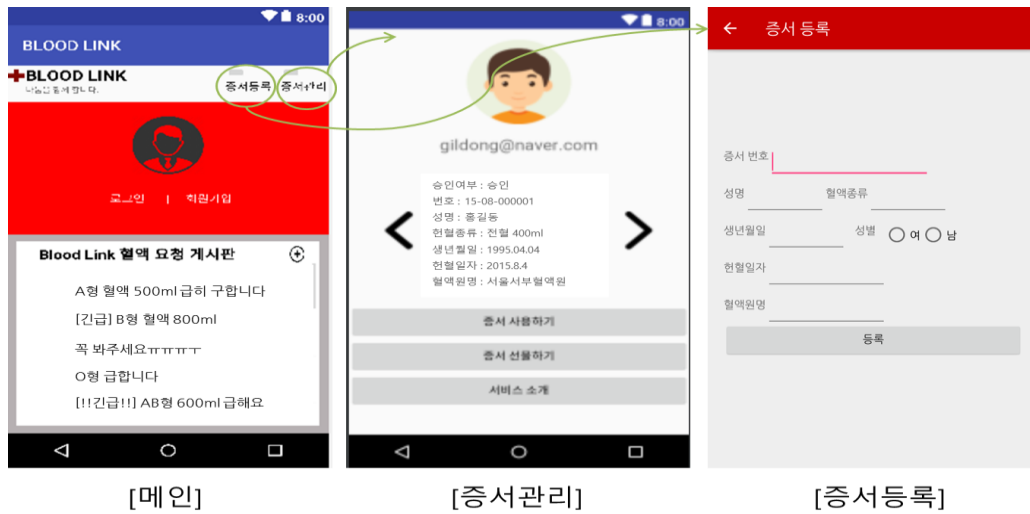
- ◆ 처음 어플리케이션을 켜면 위와 메인화면이 나옴.
- ◆ 이 때 로그인 버튼을 누를 시 로그인 화면으로 전환.
- ◆ 가입이 되어 있다면 본인의 아이디(이메일)과 비밀번호를 입력 후 LOGIN 버튼을 눌러 어플 사용.
- ◆ 이 때 'BloodLink' 서버의 DB에 해당 사용자 정보가 저장되어 있는 경우 로그인이 성공하고 다시 메인화면으로 돌아감.
- ◆ 회원이 아니라면 회원가입과정 진행 가능.
- ◆ 메인화면에서 회원가입을 버튼을 누르고 회원가입 화면에서 본인의 아이디, 비밀번호, 이름을 입력하고 REGISTER 버튼을 누르면 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



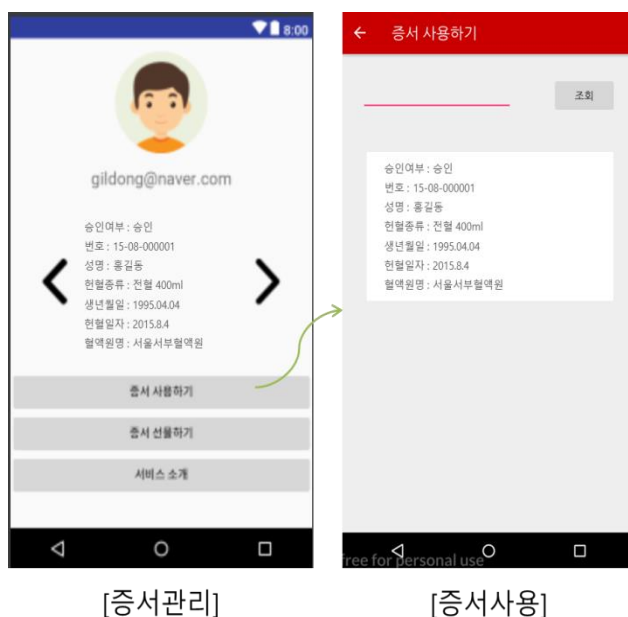
- ◆ 메인화면에서 보이는 게시판 옆의 '+'버튼을 클릭하면 게시판 페이지로 전환
- ◆ 게시판 페이지에서 현재 등록되어 있는 게시글과 그 게시글에서 필요로 하는 혈액 달성량을 progress bar로 보여줌.
- ◆ 이 때 100% 충족되면 해당 글은 자동으로 닫히게 되고 사용자의 눈에 보이지 않게 됨.
- ◆ 또한, 게시글 작성자가 임의로 글을 닫을 수 있음..
- ◆ 게시판 페이지에서 글쓰기 버튼을 누르면 게시글을 쓸 수 있는 글쓰기 화면으로 전환.
- ◆ 제목, 작성자, 목표 혈액량, 혈액이 필요한 이유를 작성할 수 있는 페이지로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

## ■ 증서 관리 및 등록

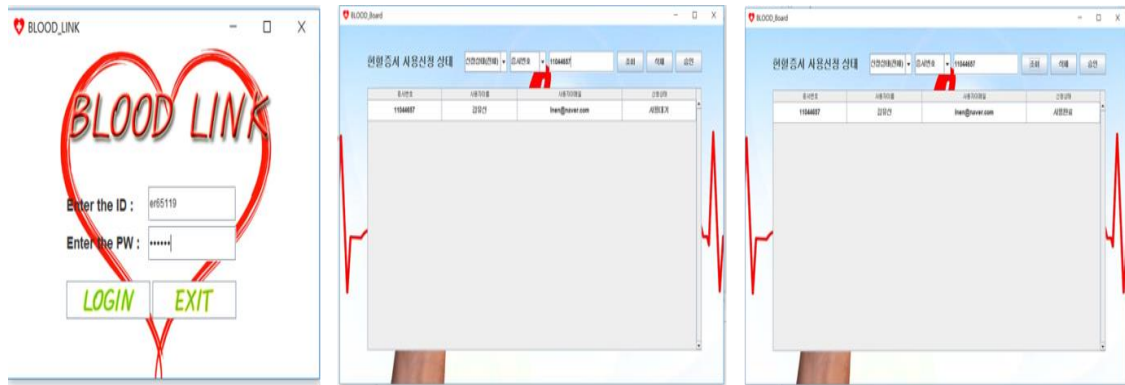


- ◆ 메인화면 상단의 증서 등록 버튼 클릭 시 사용자가 헌혈 증서를 등록할 수 있는 페이지로 전환.
- ◆ 해당 페이지에서 증서의 정보를 입력 후 등록 버튼 클릭 시 서버로 전송이 되고 승인이 난 후 DB에 저장이 됨.
- ◆ 등록된 증서는 증서 관리 페이지에서 조회 가능.
- ◆ 메인에서 증서 관리 버튼 클릭 시 증서관리 페이지로 화면 전환이 되고 해당 페이지에서 사용자의 보유 증서 확인 가능.

## ■ 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.

#### IV. 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 구현하였는

데 Java의 클래스 객체를 MySQL에 데이터화 할 때 클래스 변수와 테이블 속성이 Mapping되지 않는 문제가 발생하여 정상 작동하지 못함

- Spring Boot에서 사용자에게 대한 Model을 정의할 때 User 클래스를 Donor(헌혈 기증자, 일반 사용자), MI(의료기관, 증서 사용자) 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

## V. 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	10%

		회원, 게시판 관련 기능 개발	
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>

참여자	박순렬, 조영래, 박준현, 김민지, 이한솔
회의 일시	2018년 5월 20일 (일) 14:00
회의 장소	경북대학교 공대 9호관 418호

회의 안건	클라이언트, 서버 구현의 세부 역할 분배와 구성에 대한 논의
회의 내용	<p>요구사항에서 수정을 바라는 부분에 대한 토의</p> <ul style="list-style-type: none"> <li>- 등록된 헌혈증서를 꼭 이미지 형태로 보여주어야 하는가 <ul style="list-style-type: none"> <li>➔ 난이도에 비하여 중요한 기능이 아니므로 TextView로 정보 제공하기로 한다.</li> </ul> </li> </ul> <p>클라이언트, 서버 구현에 있어 세부적인 역할 분배</p> <ul style="list-style-type: none"> <li>- 단순히 안드로이드 어플리케이션, 자바 어플리케이션, 서버로 나뉘어져 있었던 팀원들의 역할을 구체적으로 어떤 기능을 구현할 지 정하였다.</li> <li>- SRS에 작성된 기능을 바탕으로 구현 난이도에 따라 묶어서 분배하였다.</li> </ul> <p>Spring boot 서버의 구조 논의</p> <ul style="list-style-type: none"> <li>- MVC 패턴으로 구성하되 View의 경우 자바 어플리케이션과 안드로이드 어플리케이션이 존재하므로 Model과 Controller로 구성된다.</li> <li>- 클라이언트에 REST API로 제공해주기로 하였고 거기에 맞는 대략적인 포맷을 결정하였다.</li> <li>- Heroku를 사용하여 서비스를 제공하자.</li> </ul>
회의 결론	<p>클라이언트 역할 분배:</p> <ul style="list-style-type: none"> <li>- 자바 어플리케이션 <ul style="list-style-type: none"> <li>➔ 박준현</li> </ul> </li> <li>- 안드로이드 어플리케이션 <ul style="list-style-type: none"> <li>➔ 조영래 : 마이 페이지, 로그인 페이지, 회원가입 페이지, 게시판 2개</li> <li>➔ 김민지 : 증서 등록, 증서 관리, 증서 사용하기, 증서 선물하기</li> </ul> </li> </ul> <p>서버 역할 분배:</p> <ul style="list-style-type: none"> <li>➔ 박순렬 : Controller</li> <li>➔ 이한솔 : Model, Heroku</li> </ul>

	<p>Spring boot 서버 구조:</p> <ul style="list-style-type: none"><li>- 일부 필요한 부분은 상속을 사용해서 효율적으로 구현을 하자.</li><li>- 회원가입 포맷</li><li>- [Donor 계정] [password] [사용자 이름] [사용자 연락처]</li></ul> <p>Ex) <a href="mailto:000@gmail.com">000@gmail.com</a> 1234 홍길동 010-0000-0000</p> <ul style="list-style-type: none"><li>- 증서등록 포맷</li><li>- [증서 번호] [혈액종류] [이름] [생년월일] [성별] [날짜] [혈액원명]</li></ul> <p>Ex) 15-08-000001 전혈 400ml 홍길동 1995.00.00 남 2015.00.00 서울서부혈액원</p>
차기 계획	<ul style="list-style-type: none"><li>- 대략적인 통신 포맷을 결정하였지만 구체적인 포맷을 정해야 함.(클라이언트가 어느 정도 구현이 되면 결정하기로 함)</li><li>- 클라이언트 측에서 REST API 이용하는 통신 이해할 것</li></ul>
비고	<ul style="list-style-type: none"><li>- DB서버를 어떻게 해결을 할 지 논의가 필요</li></ul>

<b>참여자</b>	박순렬, 조영래, 박준현, 김민지, 이한솔
<b>회의 일시</b>	2018년 5월 27일 (일요일) 13:00
<b>회의 장소</b>	공대 9호관 415호

<b>회의 안건</b>	<b>클라이언트, 서버 기능 수정사항 및 통신방법에 대한 논의</b>
<b>회의 내용</b>	<p>MI 검색 기준 항목에 대한 논의</p> <ul style="list-style-type: none"> <li>- 기존의 검색 기준 항목은 항목번호, 사용자이름, 사용자 이메일, 전화번호, 등록일, 사용상태</li> <li>- 검색 기준 항목의 수가 너무 불필요하게 많음</li> <li>- 증서 번호를 검색 기준 항목에 추가해야 할지에 대한 논의</li> </ul> <p>서버와 클라이언트의 통신에 관한 논의</p> <ul style="list-style-type: none"> <li>- 서버-MI 클라이언트가 정보검색을 위해 통신할 때 방법에 대한 논의</li> <li>- 서버-MI 클라이언트가 사용처리를 위해 통신할 때 방법에 대한 논의</li> <li>- 서버-MI 클라이언트가 통신 할 때 사용할 데이터 포맷 결정</li> </ul>
<b>회의 결론</b>	<p>MI 검색 기준 항목 :</p> <ul style="list-style-type: none"> <li>- 항목번호는 검색 시 활용을 위해 증서번호로 대체</li> <li>- 그 외에 사용자 이메일, 사용자이름, 사용자 이메일, 사용상태만 사용</li> <li>- 나머지 전화번호, 등록일은 불필요하다 판단하여 삭제</li> </ul> <p>서버와 클라이언트의 통신 :</p> <ul style="list-style-type: none"> <li>- 정보 검색 시 정보를 통째로 클라이언트에게 넘기고 클라이언트 내에서 정렬 및 수정 시, 개별 수정부분에 대한 파악이 어려움</li> <li>- 그러므로 정보 검색 할 때 마다 서버와 연결하여 서버 내에서 정보를 찾는 방법으로 변경</li> <li>- 정보 수정 부분 또한 마찬가지로 방식으로 변경</li> <li>- 정보 검색 시 서버 내에서의 검색 기준은 무조건 증서번호.</li> <li>- MI 클라이언트에서 사용할 데이터 포맷</li> <li>- [MI계정] [증서번호] [사용자이름] [사용자이메일] [전화번호] [등록일] [신청상태]</li> </ul> <p>EX) Med01 1191204 박준현 SKYAN@naver.com 01050 2018-05-15 사용</p>
<b>차기 계획</b>	<ul style="list-style-type: none"> <li>- MI 클라이언트 개발 시 검색 기준 수정 사항 반영</li> </ul>

---

	<ul style="list-style-type: none"><li>- 서버 개발 시 검색에 관한 부분 수정 사항 반영</li><li>- 차기 회의 때 final 보고서에 대한 부분적 논의</li></ul>
비고	<ul style="list-style-type: none"><li>- 여러 번 수정되는 기능에 대해 이해가 필요</li><li>- 아직 MI 측에서 REST api를 이용하는 통신에 대한 이해 부족</li></ul>

참여자	박순렬, 조영래, 박준현, 김민지, 이한솔
회의 일시	2018년 6월 2일 (토) 19:00
회의 장소	공대9호관 408호

회의 안건	개발 진행사항 논의, 클라이언트-서버 간 통신 방법에 대한 논의
회의 내용	<p>개발 진행사항 논의</p> <ul style="list-style-type: none"><li>- 현재 각자의 진행상황 점검</li><li>- 기본 UI가 완성된 상태의 클라이언트 중간점검</li><li>- 수정 내역에 관한 회의</li></ul> <p>클라이언트-서버 간 통신방법</p> <ul style="list-style-type: none"><li>- 서버는 Spring Framework MVC모델의 웹 서버를 채택</li><li>- RestFul API의 데이터 전송방식</li><li>- DB 사용 여부 및 적용 방법에 대한 논의</li></ul> <p>Final 보고서 작성에 대한 논의</p>
회의 결론	<p>UI에 대한 수정</p> <ul style="list-style-type: none"><li>- 기능과 관련 없는 UI에 대해 간소화</li><li>- 기능 관련 부분의 UI 강조</li></ul> <p>Rest API에 대한 경험 부족</p> <ul style="list-style-type: none"><li>- 부족한 부분 및 개념을 위주로 각자 학습</li><li>- 최근 웹 서버 프로그래밍 수업을 들은 3인이 관련 내용을 정리하여 공유</li><li>- 서버 개발 이후 프로그래밍에 적용이 쉽도록 관련 문서를 제작하기로 함</li></ul> <p>DB적용에 대한 내용</p> <ul style="list-style-type: none"><li>- DB는 MySQL을 적용</li><li>- 관리자가 웹을 통하여 관리</li><li>- 웹에서는 Spring과 연동을 위하여 hibernate를 적용</li><li>- 여러 사람이 프로젝트를 통해 이용할 수 있도록 서버와 DB를 호스팅하여 이용</li></ul> <p>Final 보고서 작성 내용</p>



---

	<ul style="list-style-type: none"><li>- 현재까지의 개발 중 수정 및 적용 된 요구사항들에 대한 확인</li><li>- Risk에 대한 복기 및 앞으로 남은 잠재적 risk 를 관리</li><li>- 요구사항 및 final 보고서 작성 내용과 관련 된 개발상황 논의</li></ul>
<b>차기 계획</b>	Risk에 대해 되돌아보고 잠재적인 risk 들을 관리 세부적인 final 보고서 내용에 대한 각자의 작업내용 공유 Restful API에 대한 학습자료 마련
<b>비고</b>	<ul style="list-style-type: none"><li>- Restful API, hibernate 연동 등에 대한 경험이 부족하여 각자의 성실한 학습이 요구됨</li></ul>

<b>참여자</b>	박순렬, 박준현, 김민지, 이한솔
<b>회의 일시</b>	2018년 6월 6일 (수요일) 14:00
<b>회의 장소</b>	공대 9호관 415호

<b>회의 안건</b>	<b>개발 중간 점검, 리스크 관리</b>
<b>회의 내용</b>	<p>개발 중간점검</p> <ul style="list-style-type: none"> <li>- 프로젝트 설계, 구조화 완료</li> <li>- 각 기능별 구현 필요 상태</li> <li>- 요구사항에 대한 일부 수정 필요</li> </ul> <p>리스크 관리</p> <ul style="list-style-type: none"> <li>- 취업으로 인한 팀원의 이탈</li> <li>- 프로젝트 외 일정들로 인해 회의, 개발 등이 상당부분 지연됨</li> <li>- 현재까지 발생한 리스크 및 앞으로 발생할 리스크에 대한 분석 필요</li> </ul> <p>개발 계획 논의</p> <ul style="list-style-type: none"> <li>- 진행중인 개발 상태 확인</li> </ul>
<b>회의 결론</b>	<ul style="list-style-type: none"> <li>● 취업, 질병 팀원 이탈 : 현재 조영래 학우가 취업 연계 인턴 활동 차 팀 활동 참여에 불참 중임. 이를 어떤 식으로 해결하는게 좋겠는가? <ul style="list-style-type: none"> <li>- 조영래 학우와 함께 있는 카카오톡 단체톡방에서 회의를 진행하거나 오늘까지 진행하였던 회의 내용을 공유하여 원격 회의를 진행하자.</li> <li>- 조영래 학우가 맡은 부분을 남겨 놓고 일단 회의를 진행한 후, 후에 다시 돌아왔을 때, 설명해주고 그의 역할을 최소로 하자. -&gt; 후에 역할 분담에서 점수를 낮추는 방법 고려</li> <li>- 그의 상세 일정을 알고 그가 참여할 수 있는 시간대를 물어보자.</li> </ul> </li> <li>● 개발 경험 부족 : 박준현 학우의 개발 경험 부족은 어떻게 해결할 수 있는가? <ul style="list-style-type: none"> <li>- 준현 학우의 개발 경험 부족이 걸림돌일 수 있지만, 그의 학습 의지가 뚜렷해 보이므로, 자바 앱 개발은 순조로움.</li> </ul> </li> <li>● 일정 관리 : 현재 일정이 많이 밀려 있음, 어떤 식으로 해결할 수 있는가? <ul style="list-style-type: none"> <li>- 기존의 예정 되어있지 않았던 조영래 학우의 이탈, 기말 시험의 빠른</li> </ul> </li> </ul>

	<p>일정 등은 일정 관리에 큰 변수, 하지만 기존의 길었던 회의 시간을 최소화하여 해결 할 수 있지 않을까?</p> <ul style="list-style-type: none"><li>● 동기 결여 : 현재 팀장 및 팀원들의 동기가 많이 결여 되어있는 것 같다<ul style="list-style-type: none"><li>- 인력 부족, 시간 부족 등으로 기존의 동기가 많이 결여되었음. 어떻게 해결할까? 회식을 통한 단합회? 각자 충분한 휴식을 취한 후 다시 동기 부여?</li></ul></li><li>● 유사 서비스 : 기존의 헌혈 어플과 서비스가 많이 겹친다.<ul style="list-style-type: none"><li>- 기존의 헌혈 어플 서비스와 팀 프로젝트에서 제공하고 하는 서비스가 상당 부분 오버랩 되어있음, 과감하게 많은 서비스를 포기하고 우리가 중점적으로 보여줄 만한 서비스들만 부각시키자. 그게 바로, 헌혈 증서 전산화, 기부 및 증서 관리이다.</li></ul></li></ul>
차기 계획	<ul style="list-style-type: none"><li>- 각 파트별 개발 내용 통합 방법 논의</li><li>- 담당 파트 개발 촉진</li></ul>
비고	<ul style="list-style-type: none"><li>- 취업 인턴 과정으로 인한 이탈자 발생(조영래)</li></ul>

참여자	박순렬, 조영래, 박준현, 김민지, 이한솔
회의 일시	2018년 6월 16일 (토) 14:00
회의 장소	공대9호관 418호

회의 안건	결과보고서, 발표 논의
회의 내용	<p>결과보고서 작성</p> <ul style="list-style-type: none"> <li>- 결과 보고서 작성 요구내용 분석</li> <li>- 작성 내용 분담 방안 논의</li> <li>- 결과보고서 작성할 내용 검토</li> </ul> <p>발표자료 작성</p> <ul style="list-style-type: none"> <li>- 발표 담당자 선정</li> <li>- 발표 내용 논의</li> <li>- 발표 테마, 구성 결정</li> </ul> <p>프로젝트 시연</p> <ul style="list-style-type: none"> <li>- 프로젝트 시연 방안 논의</li> <li>- 시연 시나리오 구성</li> <li>- 시연 환경 고려</li> </ul>
회의 결론	<p>결과보고서 작성</p> <ul style="list-style-type: none"> <li>- SRS에 명세된 요구사항들을 검토</li> </ul> <p>기능 요구사항</p> <ul style="list-style-type: none"> <li>● F-ManC-1. 헌혈증서 등록 -구현(1-1생략)(1-2정보입력으로 대체) 50%</li> <li>● F-ManC-2. 헌혈증서 승인 -유효성확인증생략(헌혈증서 양식만 확인하고 맞으면 승인) 50%</li> <li>● F-ManC-3. 헌혈증서 조회 -구현 100%</li> <li>● F-DonC-1. 헌혈증서 요청 -회원가입(우선순위에서 배제), 로그인 구현, 게시글 작성은 필수, 게시글 비활성화 구현 -&gt; ex)게시글 만료</li> <li>● F-DonC-2. 헌혈증서 전달 -구현 (적립은 보류)</li> <li>● F-DonC-3. 헌혈증서 펀딩 -구현 (등급적립은 보류)</li> </ul>

	<ul style="list-style-type: none"> <li>● F-DonC-4. 헌혈증서 선물 -구현</li> <li>● F-UseC-1. 헌혈증서 제출 -의료기관 검색은 안드로이드</li> <li>● F-UseC-2. 헌혈증서 접수 -구현</li> <li>● F-UseC-3. 헌혈증서 거부 -구현</li> <li>● F-ManI-1. 회원가입 -보류</li> <li>● F-ManI-2. 로그인 -구현</li> <li>● F-ManI-3. 회원정보 조회 -생략(보고서 상에서만 되어있고 시연은 x)</li> <li>● F-ManI-4. 회원정보 수정 -생략(보고서 상에서만 되어있고 시연은 x)</li> <li>● F-ManI-5. 회원 탈퇴 -생략(보고서 상에서만 되어있고 시연은 x)</li> </ul> <p>비기능 요구사항</p> <ul style="list-style-type: none"> <li>● N-AD-1 : 구현</li> <li>● N-CD-1 : 구현</li> <li>● N-CD-2 : 구현</li> <li>● N-CD-3 : 생략</li> <li>● N-CD-4 : 생략</li> <li>● N-DM-1 : (MI는 계정 1개, Donor 계정2개로 시연)</li> <li>● N-RT-1 : 생략</li> <li>● N-RT-2 : 구현</li> <li>● N-IF-1 : 생략</li> <li>● N-IS-2 : 생략</li> </ul> <p>- Proposal에 명시된 Risk 검토</p> <ul style="list-style-type: none"> <li>● 리스크 쓰는 사람은 proposal 에서 가져와서 했다 안했다 여부만 작성</li> <li>● 데이터 무결성 위배 : x</li> <li>● 취업 질병 팀원 이탈 : o</li> <li>● 윤리적 문제 : o</li> <li>● 개발 경험 부족 : o</li> <li>● 일정 관리 : o</li> <li>● 동기 결여 : o</li> <li>● 손상 고장 : o</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● 의견 대립 갈등 : x</li> <li>● 수치 미산정 : x</li> <li>● 유사 서비스 : o (해결 : 차별화했음)</li> <li>● 비효율적인 자원 : o</li> <li>● 팀원 역할 : x</li> <li>● 새로발생한 Risk : 회의에서의 잡담으로 효율 저하</li> </ul> <p>- 5명의 팀원이 아래와 같이 결과보고서 내용을 분담 작성하기로 함</p> <ul style="list-style-type: none"> <li>● 박순렬 : 기여도</li> <li>● 조영래 : Risk 분석, 기술적 어려움</li> <li>● 박준현 : 기능 요구사항</li> <li>● 김민지 : 시스템 기능, 참고문헌</li> <li>● 이한솔 : 비기능 요구사항</li> </ul> <p>결과보고서는 팀장이 최종 통합하여 제출</p> <p>발표자료 작성</p> <ul style="list-style-type: none"> <li>- 발표 담당자 : 박순렬</li> <li>- 발표 내용은 담당자가 구성하기로 함</li> <li>- 10분 내로 발표 및 시연이 이루어져야 하기 때문에 발표 내용은 간소화하고 시연에 집중하기로 함</li> <li>- 발표 자료 작성 또한 담당자가 수행</li> </ul> <p>프로젝트 시연</p> <ul style="list-style-type: none"> <li>- 시연 환경은 아래와 같이 구성             <ul style="list-style-type: none"> <li>● 서버 : Heroku 상에 호스팅되어 있음</li> <li>● Android App : 기증자, 수증자를 보이기 위한 안드로이드 2대</li> <li>● Java App : 의료기관을 보이며 발표 PC에 설치</li> </ul> </li> <li>- 시연 시 발표자가 의료기관, 팀원 2명이 일반 사용자 역할을 맡아 시연을 수행</li> <li>- 안드로이드 화면을 보이기 위해 웹캠을 활용하는 방안 검토</li> <li>- 시연 시나리오의 아래와 같이 구성             <ul style="list-style-type: none"> <li>● Actor : 기증자 A, 수증자 B, 의료기관 M</li> <li>● 시연 단계 A 로그인 - A 헌혈증서 목록확인 - A 헌혈증서 추가 - B 게시글 작성 - A,</li> </ul> </li> </ul>
--	--

---

	B 게시판 갱신 확인 - A 기증 - A 포인트 적립 확인 - B 헌혈증서 목록 확인 - B 게시글 만료 - M 로그인 - B 헌혈증서 사용 - M 헌혈증서 목록 확인 - M 헌혈증서 접수 - 사용 완료
차기 계획	<ul style="list-style-type: none"><li>- 개발 완료</li><li>- 결과보고서, 발표자료 제출</li><li>- 시연 리허설</li><li>- 최종 발표</li></ul>
비고	