

파수꾼 모바일 버전

부대 내 병사 휴대폰 보안 관리 어플리케이션

APP_PASOOGUNM_TEAM

일병 이승현, 일병 김동욱

목차

- 목적 및 필요성 : 이 어플리케이션은 왜 필요한가?
- 기획 의도 : 어떻게 만들고자 했는가?
- 개발한 상태 : 현재까지 어떻게 만들었는가?
- 발전 계획 : 향후 어떻게 더 개발할 수 있을까?

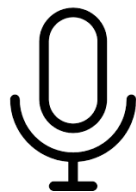
목적 & 필요성

파수꾼 Mobile Version

“

휴대폰 사용에 따라 부대 내에서 발생할 수 있는
보안 문제를 해결할 수 있는 통합 어플리케이션

”



* 파수꾼 : 현재 각 부대 국방망PC에서 사용 중인 보안 SW 프로그램

목적 & 필요성



현재 시행 중인 방안 : 보안스티커 부착

- 1) 불편함 ↑+ 비효율성 ↑+ 비효과성 ↑
- 2) 카메라에 대한 일시적·물리적 방안

기획 의도

- 각 부대의 지휘관 및 보안담당 간부들이 병사 보안 관리하는 것이 효과적이라 판단!
- 사용자(병사)와 관리자(담당 간부)의 두 다른 역할을 반영하여
기본 플랫폼 설계, 서버 및 데이터베이스 구축, 기본 기능 구현!

사용자 입장(병사)

1. 감시 상태 On / Off 유무
2. 관리자 부재 시 비밀번호를 입력하여 감시 상태를 풀 수 있는 기능(당직사관 등 타 간부)
3. 관리자와의 연결상태 실시간 표시
4. 사진 찍었을 시 빨간 화면 표시
5. 사용자가 어플리케이션 삭제 시 삭제의 권한을 관리자에게 부여하여 임의 삭제 방지

기획 의도

- 각 부대의 지휘관 및 보안담당 간부들이 병사 보안 관리하는 것이 효과적이라 판단!
- 사용자(병사)와 관리자(담당 간부)의 두 다른 역할을 반영하여
기본 플랫폼 설계, 서버 및 데이터베이스 구축, 기본 기능 구현!

관리자 입장(담당 간부(중대장 또는 보안담당 간부))

1. 관리하는 사용자들의 목록 및 보안 관련 실시간 정보(카메라 앱 구동 시간과 사진 찍은 횟수 등)
2. 사용자 감시 상태 On / Off 버튼(병사 휴가 시를 고려한 기능)
3. 사용자가 앱 삭제 시 이에 대한 내용 바로 확인
4. 사용자가 사진 찍었을 시 Push 알림으로 해당 내용 바로 확인
5. 사용자 삭제 가능(전역자/휴대폰 변경 고려)

개발 현황

1) JSP와 SQLite를 활용한 로그인/회원가입 등 기본 회원 관리 기능

PasooM

사용자로
회원가입하기

이름

아이디

관리자아이디 연결할 관리자 ID를 입력하세요:

패스워드

패스워드확인

회원가입하기

PasooM

사용자로 로그인하기

아이디

패스워드

로그인

회원가입하기

PasooM

관리자로 로그인하기

아이디

패스워드

로그인

회원가입하기

PasooM

관리자로
회원가입하기

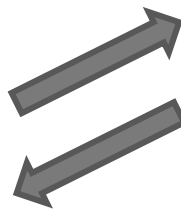
이름

아이디

패스워드

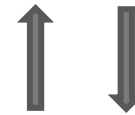
패스워드확인

회원가입하기



```
1: java.lang.String requestParam(String name) {
2:     return request.getParameter(name);
3: }
4:
5: @RequestMapping("/login")
6: public ModelAndView login(@RequestParam("id") String id,
7:                           @RequestParam("password") String password)
8:     throws Exception {
9:     ModelAndView mv = new ModelAndView();
10:    User user = userDao.findById(id);
11:    if (user == null) {
12:        mv.addObject("message", "아이디가 없습니다.");
13:        mv.setViewName("login");
14:        return mv;
15:    }
16:    if (!user.getPassword().equals(password)) {
17:        mv.addObject("message", "비밀번호가 일치하지 않습니다.");
18:        mv.setViewName("login");
19:        return mv;
20:    }
21:    mv.addObject("message", "로그인 성공!");
22:    mv.setViewName("loginSuccess");
23:    return mv;
24: }
25:
26: @RequestMapping("/register")
27: public ModelAndView register(@RequestParam("id") String id,
28:                              @RequestParam("password") String password,
29:                              @RequestParam("confirmPassword") String confirmPassword)
30:     throws Exception {
31:     ModelAndView mv = new ModelAndView();
32:    if (id.length() < 4) {
33:        mv.addObject("message", "아이디는 4자리 이상이어야 합니다.");
34:        mv.setViewName("register");
35:        return mv;
36:    }
37:    if (password.length() < 8) {
38:        mv.addObject("message", "비밀번호는 8자리 이상이어야 합니다.");
39:        mv.setViewName("register");
40:        return mv;
41:    }
42:    if (!password.equals(confirmPassword)) {
43:        mv.addObject("message", "비밀번호가 일치하지 않습니다.");
44:        mv.setViewName("register");
45:        return mv;
46:    }
47:    User user = userDao.findById(id);
48:    if (user != null) {
49:        mv.addObject("message", "이미 등록된 아이디입니다.");
50:        mv.setViewName("register");
51:        return mv;
52:    }
53:    userDao.save(new User(id, password));
54:    mv.addObject("message", "회원가입 성공!");
55:    mv.setViewName("registerSuccess");
56:    return mv;
57: }
```

Java Server Page(JSP)

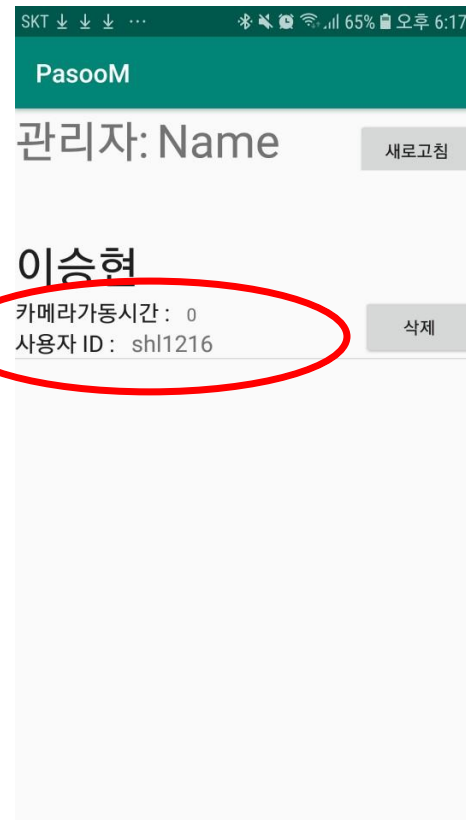
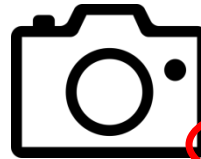
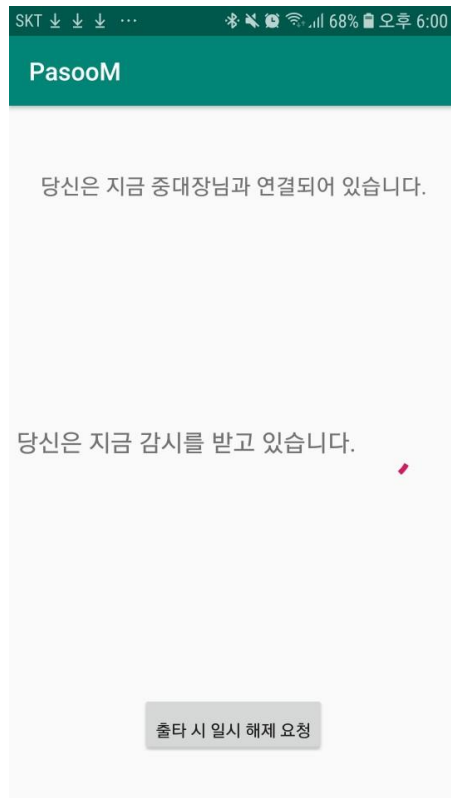


ID	PW	Name
leader	1	이민준

SQLite(Database)

개발 현황

2) 사용자의 카메라 접속 여부와 접속 시간을 관리자에게 전송



개발 현황

3) 병사의 카메라 가동 시간 확인 가능 및 명단에서 삭제 가능
(병사가 임의로 데이터베이스에 접속하여 내용 삭제 불가)



향후 발전 계획

기획했으나 아직 구현하지 못한 기능 :

- 출타 여부 반영 : 병사들 휴가 가는 등 감시가 필요 없을 시 감시 모드 해제
- 연결 상태 : 병사 및 관리자가 서버와 연결된 상태를 실시간으로 받고 표시
- 녹음, GPS 등 차단 및 알림 : 기타 보안 위험도 높은 어플리케이션 관리
- 국방망PC와 USB 연결시 차단 및 알림
- 부대 내 등록된 블루투스 기기 활용 허가

향후 발전 계획

추가적으로 해야 하거나 하고 싶은 기능 :

- 프로토타이핑 통한 예외처리 및 디버깅
- 젊은 감각으로 UI/UX 디자인
- iOS 기반 단말기에 대한 어플리케이션 적용 고민

감사합니다!