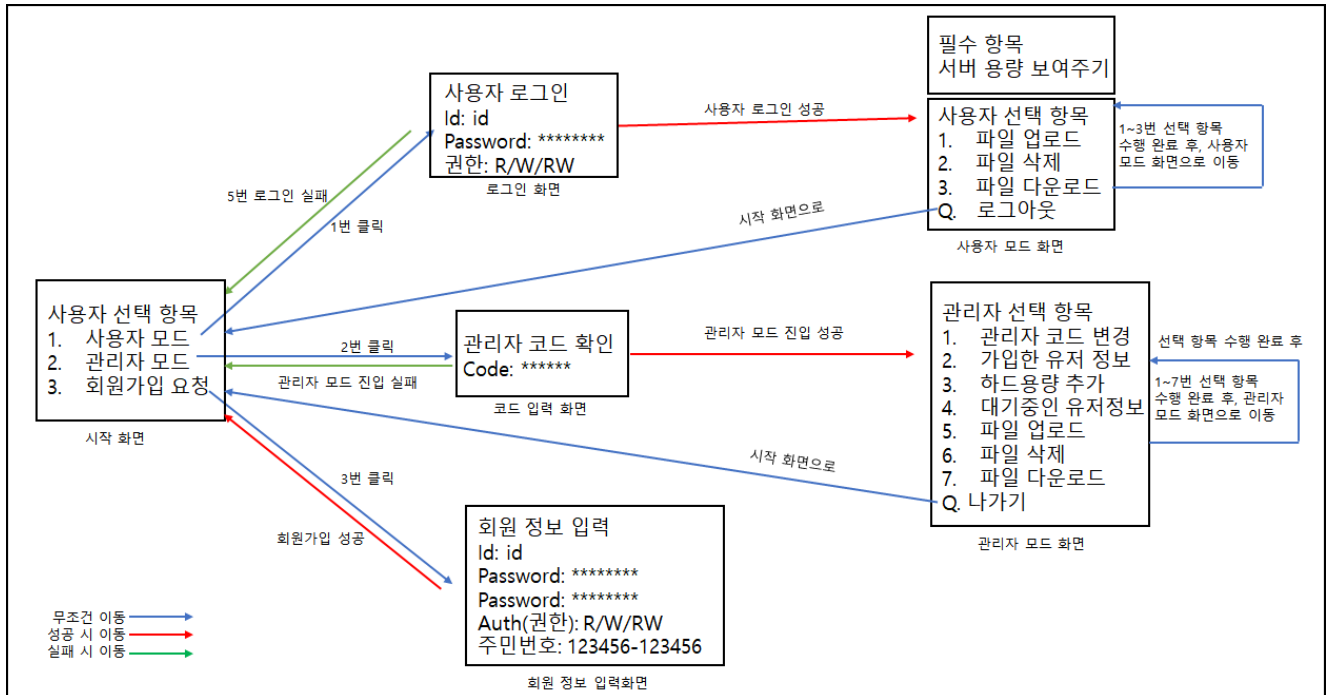


프로그램 설명서

시스템명	간단한 회원관리 프로그램	작성일	2020-02-23
단계명	-	작성자	이지혜

간단한 회원관리 프로그램



1. 프로그램 개요

- 사용자 및 파일을 관리하는 서버 프로그램 개발
- 사용자 모드 및 관리자 모드로 구성
- 그 외 회원가입 기능을 이용해 사용자 추가 가능

2. 명칭

- 회원관리 프로그램

3. 기간

- 2020년 2월 3일 ~ 2020년 2월 23일

4. 목적

- TDD(Test-Driven Development) 구현을 위한 간단한 프로그램 구현 및 테스트
- 개발자가 실무에서 접하는 업무 간접체험

목차

1. 프로그램 요구사항.....	3
2. 프로그램 흐름도	5
2-1) 사용자 흐름도.....	5
2-2) 관리자 흐름도.....	6
3. 자료구조 및 클래스.....	7
3-1) 자료구조 정의.....	7
3-2) 클래스 정의	8
4. 프로그램 구현 결과.....	9
5. 테스트.....	11
6. 문서 산출물	11

프로그램 세부 설명서

1 프로그램 요구사항

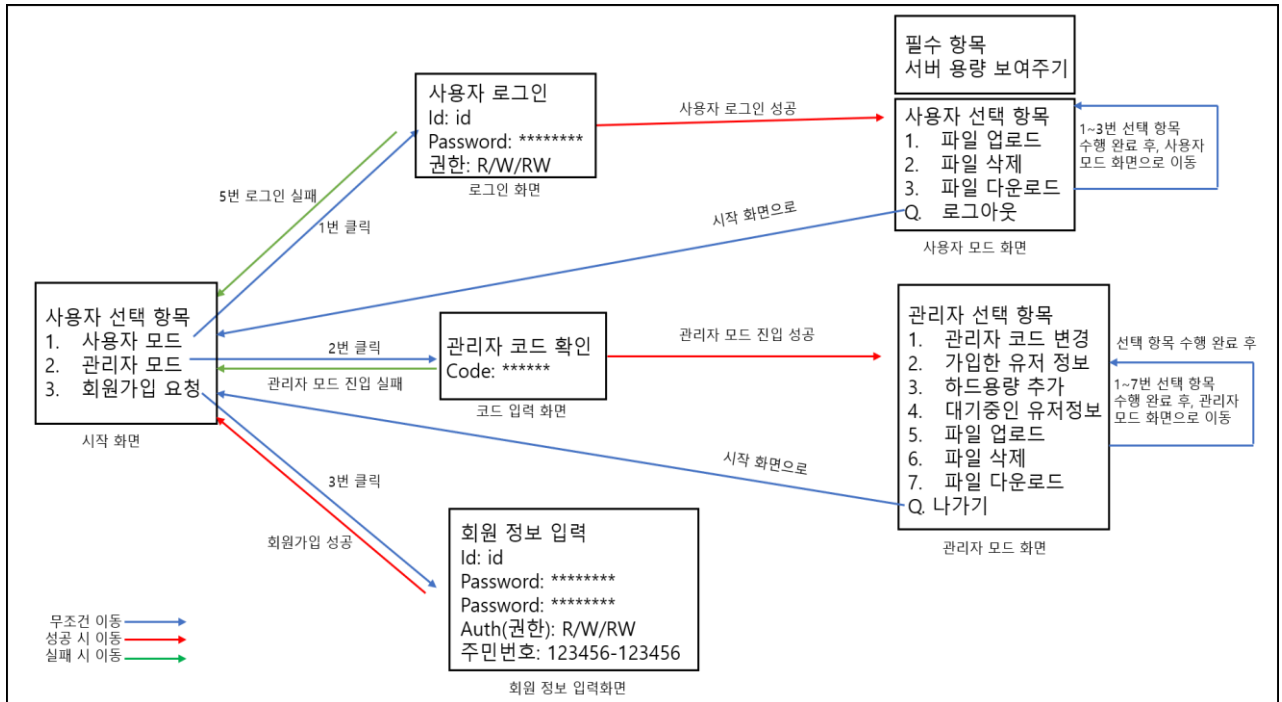


그림 1. 프로그램 개요

이 프로그램은 회원관리를 할 수 있는 프로그램으로 TDD 테스트 및 개발 업무에 필요한 지식을 쌓기 위한 목적으로 만들어졌다. 프로그램 기능은 크게 사용자 모드와 관리자모드로 나뉜다. 사용자 모드에서는 파일 관리 기능이 필요하고, 관리자 모드에서는 관리자 코드 변경/유저관리/하드용량/사용자기능이 있다. 그 외에 회원등록을 위한 회원가입 기능이 추가되었다.

- 프로그램 흐름도: 그림 1 참고
- 세부 프로젝트 범위: 표 1 참고
- 세부 프로그램 요구사항: 표 2 참고

Seq	업무	세부사항
1	개발	1) 사용자모드 구현 2) 관리자모드 구현 3) 서버 정보 업데이트 구현 4) 회원가입 구현
2	테스트	1) 기능 테스트 시트 작성 2) 테스트 수행
3	문서화	1) 프로그램 설명서 작성 2) Doxygen으로 프로젝트 문서화하기

Table 1. 프로젝트 범위

Seq	분류	요구사항	구현여부
1	로그인	1) 로그인 시 Id/Password/권한으로 사용자 확인 * R: 읽기권한, W: 쓰기권한, RW: 읽고 쓰기 권한 2) 로그인 성공 시, 성공 메시지 출력 3) 최대 5 번까지 시도 가능 4) 5 번까지 실패 -> 휴면계정	완료
		1) 로그인 성공 시, 로그인 시도 횟수 0 으로 수정 2) 로그인 실패 시, 구체적 사유 명시 X	완료
2	사용자 모드	1) 파일 업로드/파일 삭제 시, 서버 용량 체크 2) 파일 업로드/파일 삭제/파일 다운로드 시, W 혹은 RW 권한 필요	완료
3	관리자모드	1) 관리자 모드를 접속하려면 코드가 필요 2) 관리자 코드 변경/유저정보 및 관리/하드용량 추가/ 기본 유저권한 기능 사용 가능(파일 업로드, 다운로드, 삭제) 3) 하드용량추가는 총 3 회까지 가능	완료
		1) 관리자 코드 입력 값 -> '*' 로 변환 2) 관리자 코드 변경 시 -> 두 번 입력, '*' 로 변환 3) 관리자 코드 암호화(SHQ256 이상 권장) 4) 파일 업로드/파일 삭제 시, 서버 용량 체크 5) 관리자 코드 변경 시, 기존 관리자 코드 체크 6) 유저 정보 출력 시: 유저 자료구조 모두 보여주기 7) 서버 용량 추가 시 입력 값이 0 보다 클 때만	완료
4	회원가입	1) ID/Password/주민번호/신청권한 입력 2) Password: 영어 대소문자 및 숫자만 가능/8 자 이상 * 대소문자 + 숫자로 구성되어야 통과 가능 3) Password 에 주민번호 포함하지 말 것 * 주민번호를 먼저 입력 받고 password 입력 받기 4) 회원가입 완료 후, 승인 대기 유저로 이동 5) 관리자가 승인 후, 유효한 유저로 이동	완료
		1) id 입력 -> 대기유저와 유효한 유저 ID 와 동일한지 체크 2) 아이디 6 글자 이상 3) 비밀번호 -> 두 번 입력, 사용자 입력 '*' 로 변환 4) 주민번호 형식 체크 -> 000000-000000 5) 비밀번호와 주민번호는 저장 시 암호화하기(SHA256 이상 권장)	완료
5	암호화	1) 비밀번호와 주민번호는 암호화 후 저장 2) SHA256 해시 함수 이상 사용 권장	완료 SHA256 사용
6	파일	1) 파일 업로드 시, 서버 용량 확인 2) 파일 삭제 시, 서버에 보유 중인 파일 확인	완료
7	기타	1) 화면 구분 명확하게 하기 2) 뒤로가기	보류
8	심화기능	1) 시간 순으로 사용자 명령어 확인 2) 최대 10 개까지 확인 3) undo 기능 포함(hint: 메멘토 패턴 사용)	보류
		1) 관리자 기능: 서버상태 백업(hint: clone 함수 참고) 2) 시간 순으로 N 번까지 백업 가능 3) 복구 시, N 번 선택 가능	보류

Table 2. 프로그램 세부 요구사항 및 제약사항

2 프로그램 흐름도

2-1) 사용자 흐름도

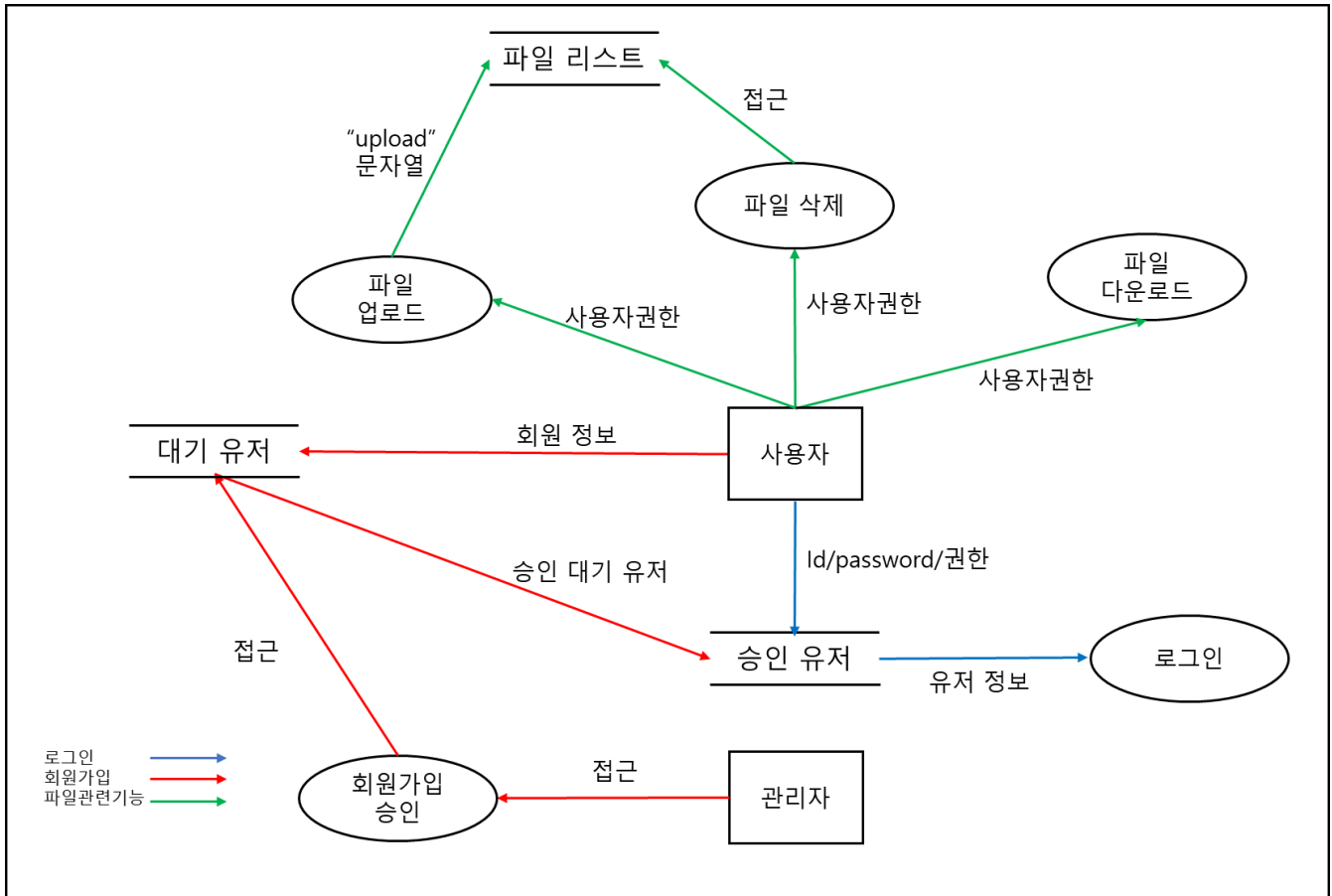


그림 2. 흐름도 by 사용자

사용자 입장에서 접근할 수 있는 기능은 회원가입, 로그인, 파일처리(업로드, 삭제, 다운로드)이다. 사용자로 인해서 발생하는 데이터는 유저 정보가 있으며 회원가입, 로그인, 파일기능 사용에 필요하다.

- 로그인
 - “ID/Password/권한” 데이터가 승인 유저 데이터 안에 있는지 확인
 - 유저 정보가 있다면 로그인 성공
- 회원가입
 - 회원 정보 입력하면 대기 유저 데이터에 추가
 - 관리자가 승인 모드에 들어가면 대기유저에 접근
 - 관리자가 회원 승인하면 대기 유저가 승인유저 데이터로 이동
 - 대기 유저는 clear
- 파일관련 기능
 - 사용자 권한 체크 -> W/RW 인 경우, 기능 사용 가능
 - 파일 업로드 시, “upload” 문자열 삽입
 - 파일 삭제 시, “upload” 문자열 한 개 삭제
 - 다운로드 시, “다운로드” 문자열 출력
- 기타사항: 그림 2 참고

2-2) 관리자 흐름도

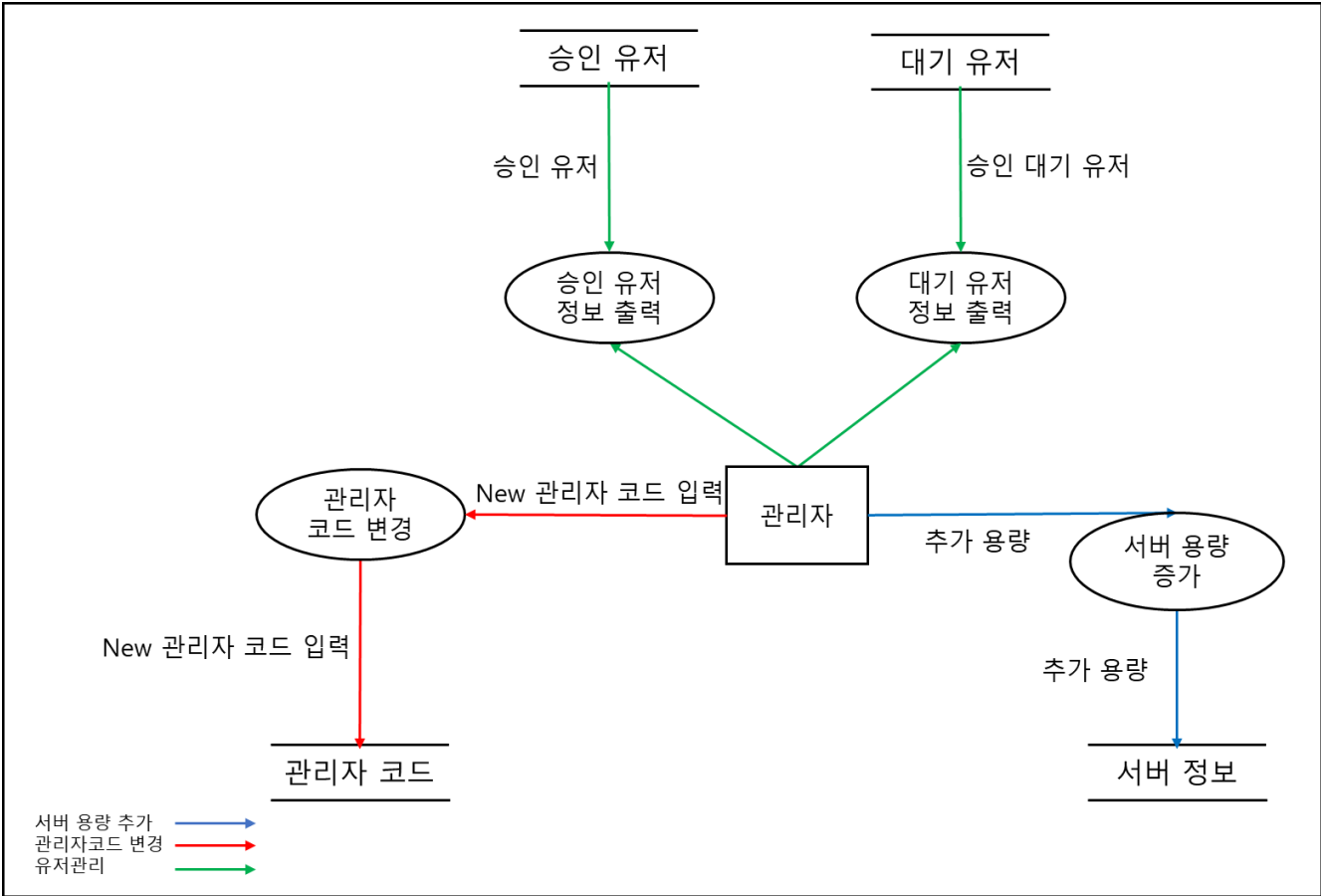


그림 3. 흐름도 by 관리자

관리자 입장에서 접근할 수 있는 기능은 회원관리, 서버관리(용량, 코드), 파일처리(업로드, 삭제, 다운로드)이다. 관리자로 인해서 발생하는 데이터는 관리자 코드, 서버 용량, 유저(승인, 대기)가 있다. 파일처리 기능은 권한 체크를 제외하고 동일하므로 제외했다. 그 외에 회원가입 승인 등 중복되는 부분 생략했다.

- 서버 용량
 - 관리자가 추가하고 싶은 용량 입력 후 용량 추가
- 관리자 코드 변경
 - 새로운 관리자 코드 입력 후 관리자 코드 변경
- 유저 관리
 - 승인 출력 시 승인 유저나 대기 유저 데이터에 접근
- 기타사항: 그림 3 참고

3 자료구조 및 클래스

3-1) 자료구조 정의

유저 형태	자료형	변수명	설명	기타
struct 사용자	string	id	사용자 아이디	-
	string	passwd	비밀번호(SHA256 으로 암호화된 후 저장)	-
	string	auth	권한(R/W/RW)	-
	string	identity	주민번호(SHA256 으로 암호화된 후 저장)	-
	unsigned int	tried	로그인 시도 횟수/기본 0	-
	bool	active	휴면 계정 여부/기본 false	-
Struct 관리자	string	code	관리자 접속 시 필요한 관리자 코드	-
class 서버	unsigned int	total_disk	서버 총 디스크 용량/기본 100 ■ getter/setter 有	-
	unsigned int	use_disk	사용중인 디스크 용량/기본 0 ■ getter/setter 有	-
	unsigned int	avail_disk	사용 가능한 디스크 용량/기본 100 ■ getter/setter 有	-
	vector<string>	files	서버에 존재하는 파일들 ■ getter/setter 有 ■ AddFile(): 파일추가 함수로 현재는 "upload" 문자열만 추가 가능 ■ DeleteFile(): 파일 삭제 함수로 현재는 맨 뒤에 있는 "upload" 문자열 삭제	-
	unsigned int	addDisk	용량 추가 횟수/기본 0 ■ getter/setter 有 ■ IncAddDisk(): 디스크 용량 수정 횟수 + 1	-

Table 3. 자료구조 정의

프로그램 요구사항 및 흐름도에서 도출한 자료구조는 3 가지로, 사용자, 관리자, 서버이다. 자세한 사항은 표 3에 정의되어 있다.

3-2) 클래스 정의

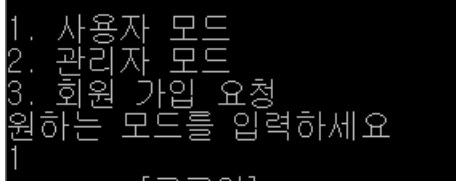
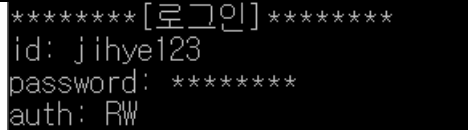
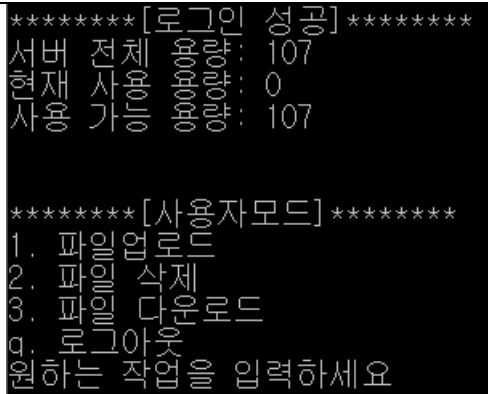
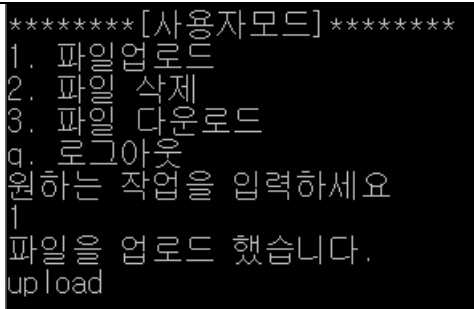
클래스 구분	기능 구분	함수	기능	기타
관리자 모드 클래스	초기 화면	initAdminMode	관리자 모드 진입 성공 -> 관리자 행위 선택	-
	관리자 코드 제어	ChangeCode	관리자 모드 접속 코드 수정	-
		InputCode	관리자 코드 입력 -> '*' 변경	-
	사용자 데이터 제어	ShowUserInfo	승인된 유저 정보 모두 출력	-
		showWaitUser	대기 중인 유저 정보 모두 출력	-
		ApproveUser	대기 중인 모든 유저 승인, 대기 유저 벡터 비우기	-
		PrintUsers	실제로 사용자 자료구조 출력하는 cout 모음	-
	서버 디스크 및 파일 제어	AddDisk	디스크 용량 추가	-
		CheckChangedDisk	디스크 추가 횟수 확인	-
		showDiskInfo	서버 디스크 정보 출력	-
		UploadUpdate	파일 업로드 후, 서버 용량 1 증가 * "upload" 문자열만 서버에 files 변수에 추가	-
		DeleteUpdate	파일 삭제 후, 서버 용량 1 감소 * "upload" 문자열만 서버에 files 변수에 삭제	-
		FileDownload	파일 다운로드 되었다는 문자열 출력	-
		PrintFiles	서버의 files 벡터 내에 있는 모든 정보 출력	-
사용자 모드 클래스	초기 화면	initUserMode	로그인 후, 사용자 기능 선택	-
	서버	showServerInfo	서버 용량 정보 출력	-
	파일 제어	CheckPermission	사용자 권한 체크 -> R이면 false	-
		FileUpload	파일 추가 기능으로, 현재 "upload" 문자열만 서버에 files 변수에 삭제 사용자 권한이 R이면 사용 불가능	-
		UploadUpdate	파일 업로드 후, 서버 용량 1 증가	-
		FileRemove	파일 삭제 기능으로, 현재 "upload" 문자열만 서버에 files 변수에 삭제 사용자 권한이 R이면 사용 불가능	-
		DeleteUpdate	파일 삭제 후, 서버 용량 1 감소	-
		FileDownload	파일 다운로드 되었다는 문자열 출력	-
		PrintFiles	서버의 files 벡터 내에 있는 모든 정보 출력	-
				-
서버 클래스	초기화면	initConsole	초기 화면: 관리자 및 사용자모드/회원가입 선택	-
	로그인	LoginUser	로그인 시도 함수: 아이디, 비밀번호 입력	-
		CheckUser	승인유저인지, 휴면계정인지 확인	-
		PlusUserTried	사용자가 로그인 틀린 횟수 체크	-
	관리자	LoginAdmin	관리자 코드 입력	-
		CheckCode	관리자 코드가 일치하는지 체크/일치하면 true 반환	-
		InputToSecret	관리자코드/비밀번호 입력 값 -> "*" 로 변환	-
	회원가입	registerUser	아이디, 비밀번호, 권한, 주민번호 입력 받기 입력 완료 시, 대기 사용자로 추가	-
		InputId	아이디 입력 받기 아이디 길이 체크	-
		isNotVaild_ID	사용 중인 아이디인지 체크(승인된 사용자와 비교)	-
		isNotWait_ID	사용 중인 아이디인지 체크(대기 사용자와 비교)	-
		isVaild_Passwd	패스워드 체크 -> "[a-zA-Z]+[0-9]+" 정규식 사용 대소문자 및 영어로 구성(최소 1 개 이상 사용) 길이 8 이상인지	-
		isNotContain_iden	비밀번호에 주민번호 포함여부 확인하기	-
		InputAuth	사용자가 원하는 권한 입력 받기 R W RW 만 올바른 입력으로 간주	-
		InputIdentity	주민번호 입력 받기	-
		isVaild_Identity	주민번호 형식 일치 확인 -> "[0-9]{6}-[0-9]{6}" 정규식 사용	-

Table 4. 클래스 정의.

프로그램 구현을 위해 사용한 클래스는 관리자 모드 클래스(ServerMode.h), 사용자 모드 클래스(UserMode.h), 서버 클래스(Server.h)이다. 각 클래스 기능은 함수별로 정리해서 표 4에 자세히 설명했다. 이 외에 암호화를 위해 KISA에서 공개한 sha256 클래스(sha256.h)를 사용했다.

4 프로그램 구현 결과

프로그램 구현 결과 시작화면, 사용자 모드 및 기능, 관리자 모드 및 기능, 회원가입 기능 등을 캡처 후, 표 5로 정리했다.

Seq	분류	설명	이미지
1	시작 화면	시작화면	
2	사용자 모드	로그인 화면	
		사용자 모드 화면	
		파일 업로드	

		파일 삭제	<p>*****[사용자모드]*****</p> <p>1. 파일업로드 2. 파일 삭제 3. 파일 다운로드 q. 로그아웃 원하는 작업을 입력하세요</p> <p>2</p> <p>파일을 삭제했습니다. 파일이 없습니다.</p>
		파일 다운로드	<p>*****[사용자모드]*****</p> <p>1. 파일업로드 2. 파일 삭제 3. 파일 다운로드 q. 로그아웃 원하는 작업을 입력하세요</p> <p>3</p> <p>파일이 다운로드 되었습니다.</p>
3	회원가입	회원가입 하기	<p>*****[회원가입]*****</p> <p>id: 6글자 이상, password: 대소문자 + 숫자, 8글자 이상 id: jihye123 주민번호를 입력하세요. ex)000000-000000 123123-123123 password: ***** password: ***** 권한 선택: R(읽기), W(쓰기), RW(읽고쓰기) RW</p>
4	관리자 모드	관리자 코드 입력	<p>1. 사용자 모드 2. 관리자 모드 3. 회원 가입 요청 원하는 모드를 입력하세요</p> <p>2</p> <p>code: *****</p>
		관리자 모드 화면	<p>*****[관리자모드]*****</p> <p>1. 관리자 코드 변경 2. 가입한 유저 정보 3. 하드 용량 추가 4. 대기 중인 유저 정보 5. 파일 업로드 6. 파일 삭제 7. 파일 다운로드 Q. 종료하기 원하는 작업을 입력하세요</p> <p>5</p>
		관리자 코드 변경	<p>관리자 확인을 위해 관리자 코드를 입력해주세요. code: *** 관리자 확인이 완료되었습니다. 바꾸고 싶은 관리자 코드를 입력해주세요. code: ***** code: *****</p>

		가입 승인된 유저 정보 확인	<pre> *****[승인된 사용자 2]***** id: jihye123 password: 2714c40e460f2e0c8a6173ee4bfc90 auth: RW identity: 6b18b6bcb1b53ce8b6b55b604698b1 tried: 0 active: 1 </pre>
		하드 용량 추가	<pre> *****[서버 용량 확인]***** 서버 전체 용량: 100 현재 사용 용량: 0 사용 가능 용량: 100 ****주의사항: 최대 3번까지 추가 가능!!!**** 추가할 디스크 용량을 입력하세요. 100 추가되었습니다. *****[서버 용량 확인]***** 서버 전체 용량: 200 현재 사용 용량: 0 사용 가능 용량: 200 </pre>
		대기 중인 유저 정보 확인	<pre> *****[대기 사용자 0]***** id: jihye123 password: 2714c40e460f2e0c8a6173ee4bfc90 auth: RW identity: 6b18b6bcb1b53ce8b6b55b604698b1 tried: 0 active: 1 모든 유저를 승인하시겠습니까? Y y </pre>

Table 5. 프로그램 구현 결과

5 테스트

테스트는 기능 중점으로 수행했으며, 올바른 입력과 올바르지 않은 입력을 넣어가며 결과를 확인했다. 리턴 값이 명확한 함수는 테스트 코드로 수행하고, void 함수나 화면 상에서 확인해야 되는 기능들은 프로그램을 수행하며 결과를 확인했다.

- “02. 프로그램 테스트시트.xlsx” 참고
- 테스트 코드는 추후 수정 예정

6 문서 산출물

프로그램을 만들면서 생긴 문서는 다음과 같다.

- 프로그램 설명서: 프로그램 개요, 흐름도, 자료구조 및 클래스 정의, 프로그램 결과 화면 포함
- 테스트시트: 프로그램 테스트 수행 및 결과 기록
- Doxygen: Doxygen 폴더에서 확인