콘솔 파일 서버 시스템 설계서

노미래

목차

[콘솔 서버 시스템 설계서 1](#_Toc46578229)

[자료구조와 클래스 2](#_Toc46578230)

[Database schema 2](#_Toc46578231)

[클래스 프로토타입 3](#_Toc46578232)

[구현 8](#_Toc46578233)

[FR 1. 로그인 8](#_Toc46578234)

[FR 2. 회원가입 9](#_Toc46578235)

[FR 3. 관리자 모드 10](#_Toc46578236)

[FR 4. 유저정보 암호화 12](#_Toc46578237)

[FR5. 파일 관련 기능 12](#_Toc46578238)

[Test cases 14](#_Toc46578239)

# 자료구조와 클래스

## Database schema

|  |  |  |
| --- | --- | --- |
| **server** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| id | INT | 서버의 unique id |
| capacity\_all | INT | 배정된 서버 하드 용량 |
| capacity\_used | INT | 사용된 서버 하드 용량 |

|  |  |  |
| --- | --- | --- |
| **user** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| id | INT | 사용자의 unique id |
| username | VARCHAR(20) | 사용자의 username(id) / UNIQUE / MIN 5 MAX 20 |
| password | VARCHAR(20) | 사용자의 password / MIN 8 MAX 20 |
| identNum | VARCHAR(13) | 사용자의 주민등록번호 / length 13 |
| authority | VARCHAR(5) | 사용자의 권한레벨 / R or W or RW |
| isdormant | BOOLEAN | 휴면계정 여부 / 기본값 False |

|  |  |  |
| --- | --- | --- |
| **pending\_user** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| id | INT | 대기중인 사용자의 unique id |
| username | VARCHAR(20) | 대기중인 사용자의 username(id) |
| password | VARCHAR(20) | 대기중인 사용자의 password |
| identNum | VARCHAR(13) | 사용자의 주민등록번호 / length 13 |
| authority | VARCHAR(5) | 대기중인 사용자의 권한레벨 |

|  |  |  |
| --- | --- | --- |
| **admin** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| id | INT | 관리자의 unique id |
| code | VARCHAR(20) | 관리자 코드/ UNIQUE / MIN 5 MAX 20 |

## 클래스 프로토타입

|  |  |  |
| --- | --- | --- |
| **ServerInfo** | | |
| **Member Variables** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| capacityAll | int | 배정된 서버 하드 용량 |
| capacityUsed | int | 사용된 서버 하드 용량 |

|  |  |  |
| --- | --- | --- |
| **UserInfo** | | |
| **Member Variables** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| username | string | 사용자의 username(id) |
| password | string | 사용자의 password |
| identNum | string | 사용자의 주민등록번호 |
| authority | string | 사용자의 권한레벨 |
| isdormant | bool | 휴면계정 여부 / 기본값 False |

|  |  |  |
| --- | --- | --- |
| **AdminInfo** | | |
| **Member Variables** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| code | string | 관리자 코드 |

|  |  |  |
| --- | --- | --- |
| **Server** | | |
| **Member Variables** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| server | ServerInfo | 서버 정보를 담고 있는 변수 |
| util | Util | 공통으로 사용되는 유틸을 담은 클래스 |
| **Methods** | | |
| **NAME** | **DESCRIPTION** | |
| run | 서버 실행. 프로그램 실행 시 가장 먼저 호출되는 함수  사용자가 선택한 기능으로 연결 | |

|  |  |  |
| --- | --- | --- |
| **Auth** | | |
| **Member Variables** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| user | UserInfo | 회원가입/로그인 시 사용할 사용자 정보 |
| **Methods** | | |
| **NAME** | **DESCRIPTION** | |
| registerUser | 사용자의 정보를 받고 검사하여 회원가입 요청 | |
| userLogin | 사용자의 id와 password를 검사하여 로그인 | |
| adminLogin | 관리자의 코드를 검사하여 로그인 | |

|  |  |  |
| --- | --- | --- |
| **UserMode** | | |
| **Member Variables** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| user | UserInfo | 사용자 정보를 담고 있는 변수 |
| server | ServerInfo | 서버 정보를 담고 있는 변수 |
| util | Util | 공통으로 사용되는 유틸을 담은 클래스 |
| fileManager | FileManager | 파일에 관련된 유틸을 담은 클래스 |
| **Methods** | | |
| **NAME** | **DESCRIPTION** | |
| run | 유저모드 시작. 유저모드 실행 시 호출되는 함수 | |
| showCapacity | 서버의 전체, 사용, 사용가능 용량을 읽어 출력 | |
| uploadFile | fileManager를 사용하여 파일 업로드 | |
| deleteFile | fileManager를 사용하여 선택된 파일 삭제 | |
| downloadFile | fileManager를 사용하여 선택된 파일 다운로드 | |

|  |  |  |
| --- | --- | --- |
| **AdminMode** | | |
| **Member Variables** | | |
| **NAME** | **TYPE** | **DESCRIPTION** |
| admin | AdminInfo | 관리자 정보를 담고 있는 변수 |
| server | ServerInfo | 서버 정보를 담고 있는 변수 |
| util | Util | 공통으로 사용되는 유틸을 담은 클래스 |
| fileManager | FileManager | 파일에 관련된 유틸을 담은 클래스 |
| **Methods** | | |
| **NAME** | **DESCRIPTION** | |
| run | 관리자모드 시작. 관리자모드 실행 시 호출되는 함수 | |
| changeAdminCode | 관리자 코드를 변경 | |
| showUserInfo | 가입한 사용자들의 정보를 조회 | |
| addCapacity | 하드 용량 추가 / 최대 3회 | |
| managePendingUser | 대기중인 사용자를 조회하고 승인 여부 결정 | |
| uploadFile | fileManager를 사용하여 파일 업로드 | |
| deleteFile | fileManager를 사용하여 선택된 파일 삭제 | |
| downloadFile | fileManager를 사용하여 선택된 파일 다운로드 | |

|  |  |
| --- | --- |
| **Util** | |
| **Methods** | |
| **NAME** | **DESCRIPTION** |
| selectMenu | 메뉴 종류를 parameter로 받아 출력하고 사용자가 선택한 메뉴를 리턴 |
| encryptString | 문자열을 parameter로 받아 암호화하여 리턴 |

|  |  |
| --- | --- |
| **FileManager** | |
| **Member Variables** | |
| server | ServerInfo |
| **Methods** | |
| **NAME** | **DESCRIPTION** |
| readCapacity | 서버에 할당된 용량과 현재 사용한 용량을 읽어 리턴 |
| selectFromFiles | 업로드 된 모든 파일의 인덱스와 이름, 용량을 출력하고 하나를 선택받아 리턴 |
| uploadFile | 사용자로부터 파일을 받아 배정된 하드에 업로드 |
| downloadFile | 사용자가 선택한 파일을 하드에서 다운로드 |
| deleteFile | 업로드 된 파일 중 사용자가 선택한 파일을 제거 |

# 구현

## FR 1. 로그인

* id와 password, 권한레벨 정보로 로그인한다.
* 로그인 실패 시 5번까지 재시도하고, 5번 실패할 경우 휴면계정으로 전환한다

|  |
| --- |
| Auth.login:      connect to the user table in the database      set loginFail as []      while true:          set username as **input**("id:")          set password as **input**("password:")          set user as findOneFromDatabase(userTable, username)          if user doesn't exist:  **print** "존재하지 않는 아이디입니다."              return false          if user.isDormant:  **print** "휴면계정입니다. 메인 화면으로 돌아갑니다."              return false            if password not equal to user.password:              if username in loginFail:                  loginFail[username]++                  if loginFail[username] >= 5:                      set user.isDormant as true                      updateToDatabase(userTable, user)  **print** "로그인 5회 실패. 휴면계정으로 전환되었습니다."  **print** "메인 화면으로 돌아갑니다."                      return false              else:                  add (username, 1) to loginFail  **print** "비밀번호 불일치. 다시 시도해주세요."              continue    *# 로그인 성공*          return true |

## FR 2. 회원가입

* id, password, 주민번호, 신청 권한을 입력한다.
* 비밀번호는 영어 대소문자 및 숫자만 가능하며 8자 이상이어야 한다.
* 비밀번호에 주민등록번호가 포함되면 안 된다.
* 정보 입력 후 회원가입 승인 요청을 보내 대기자에 등록되고, 승인여부는 관리자가 관리한다.

|  |
| --- |
| Auth.registerUser:      while true:          set username as **input**("id:")          set password as **input**("password:")          set identNum as **input**("주민등록번호:")          set authLevel as **input**("권한레벨:")          if validateUsername(username) equal to false:              continue          if validatePassword(password, identNum) equal to false:              continue          if validateLength(identNum, 13, 13) equal to false:  **print** "유효하지 않은 주민등록번호입니다."              continue          if validateAuthLevel(authLevel) equal to false:  **print** "권한레벨은 R, W, RW 중 하나여야 합니다."              continue    *# 회원가입 신청 가능한 정보*          set user as UserInfo(username, password, identNum, authLevel)          addToDatabase(pendingUser, user)  **print** "회원가입 신청이 완료되었습니다."          return true |

|  |
| --- |
| Auth.validateUsername(username):      if validateLength(username, 5, 20) equal to false:  **print** "아이디는 5자 이상, 20자 이하여야 합니다."          return false        if username in the userTable:  **print** "이미 존재하는 아이디입니다."          return false        return true |

|  |
| --- |
| Auth.validatePassword(password, identNum):      if validateLength(password, 8, 20) equal to false:  **print** "비밀번호는 8자 이상, 20자 이하여야 합니다."          return false        if password contains identNum:  **print** "비밀번호에 주민등록번호가 포함될 수 없습니다."          return false        return true |

## FR 3. 관리자 모드

* 사용자의 정보를 관리한다.
* 관리자 모드 접속 시 관리자 코드를 요구한다.
* 관리자 코드 변경, 유저 정보 열람 및 관리, 하드 용량 추가, 기본 유저권한 기능을 제공한다.
* 하드 용량 추가는 3회까지 가능하다.

|  |
| --- |
| AdminMode.run:      while true:          set selectedMenu as util.selectMenu(ADMIN)          switch selectedMenu:          case 1:  *# changeAdminCode 메소드*              set admin.code as **input**("new admin code:")              updateToDatabase(adminTable, admin)  **print** "관리자 코드가 변경되었습니다."              continue          case 2:  *# showUserInfo 메소드*              set users as findAllFromDatabase(userTable)              for user in users:  **print** user.username  **print** util.encryptString(user.password)  **print** util.encryptString(user.identNum)  **print** user.authLevel  **print** user.isDormant              continue          case 3:  *# addCapacity 메소드*              if server.capacityAll >= SERVER\_CAPACITY\_MAX:  **print** "하드 용량이 최대치로 추가된 상태입니다."                  continue              server.capacityAll += DEFAULT\_SERVER\_CAPACITY              updateToDatabase(serverTable, server)              continue          case 4:              managePendingUser()              continue          case 5:  *# uploadFile 메소드*              fileManager.uploadFile()              continue          case 6:  *# deleteFile 메소드*              fileManager.deleteFile()              continue          case 7:  *# downloadFile 메소드*              fileManager.downloadFile()              continue          case 8:  *# 메인 메뉴로 돌아가기*              return |

|  |
| --- |
| AdminMode.managePendingUser:      while true:  **print** **all** pendingUsers from pendingUserTable  **print** "관리할 대기 유저 번호를 입력해주세요."  **print** "관리자 모드로 돌아가시려면 0을 입력해주세요."          set selectedNumber as **input**()          if selectedNumber equal to 0:              return            if selectedNumber < 0 or selectedNumber > pendingUsers.length:  **print** "유효하지 않은 입력입니다."              continue            set pendingUser as pendingUsers[selectedNumber]          while true:  **print** "가입을 승인하시려면 y, 거절하시려면 n을 입력해주세요."              set hasAccepted as **input**()              switch hasAccepted:              case 'y':                  addToDatabase(userTable, pendingUser)                  deleteFromDatabase(pendingUserTable, pendingUser)  **print** "가입이 승인되었습니다."                  break              case 'n':                  deleteFromDatabase(pendingUserTable, pendingUser)    **print** "가입이 거절되었습니다."                  break              case default:  **print** "유효하지 않은 입력입니다."                  continue    *# y, n 작업 후 inner while loop 종료하고 상위 loop로 돌아감*              break |

## FR 4. 유저정보 암호화

* 유저정보 중 주민번호와 password는 관리자 열람 시 암호화하여 출력한다.

|  |
| --- |
| Util.encryptString(inputString):  *# simple XOR encryption*      set resultString as ""      for c in inputString:          resultString += c ^ ENCRYPTION\_KEY        return resultString; |

## FR5. 파일 관련 기능

* 업로드 시 파일 이름, 용량, 파일 모드를 입력받고 수용 가능한 용량인지 체크한다.
* 업로드, 삭제 시 최종 용량에 반영된다.
* 파일 삭제 시 파일 이름, 용량, 크기를 출력한다.

|  |
| --- |
| FileManager.uploadFile:      set fileName as **input**("file name:")      set fileSize as **input**("file size:")      set fileMode as **input**("file size:")      set fileLocation as **input**("file location:")      set server as findOneFromDatabase(serverTable, CURR\_SERVER\_INDEX)      set capacityAvail as (server.capacityAll - capacityUsed)      if fileSize > capacityAvail:  **print** "파일이 현재 서버의 수용 가능 용량보다 큽니다."          return false        set file as readFile(fileLocation, fileName)      writeFile(FILE\_UPLOAD\_LOCATION, file)      set server.capacityUsed += fileSize      updateToDatabase(server)  **print** "파일이 업로드 되었습니다."      return true |

|  |
| --- |
| FileManager.deleteFile:      set selectedNumber as selectFromFiles()      if selectedNumber equal to 0:          return true        set file as findFileFromDisk(selectedNumber)      removeFile(file)      set server.capacityUsed -= file.size      updateToDatabase(server)  **print** "선택하신 파일이 삭제 되었습니다."      return true |

|  |
| --- |
| FileManager.downloadFile:      set selectedNumber as selectFromFiles()      if selectedNumber equal to 0:          return true        set file as findFileFromDisk(selectedNumber)      writeFile(FILE\_DOWNLOAD\_LOCATION, file)  **print** "선택하신 파일이 다운로드 되었습니다."      return true |

|  |
| --- |
| FileManager.selectFromFiles:      set files as findAllFilesFromDisk()      for file in files:  **print** file.index  **print** file.name  **print** file.size    **print** "삭제할 파일 번호를 입력해주세요."  **print** "전 화면으로 돌아가시려면 0을 입력해주세요."      set result as **input**()      return result |

# Test cases

엑셀 파일 “Test cases.xlsx”를 참조한다.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **테스트** | **테스트 절차** | **관련 FR** |
| 1 | 로그인 성공 | 1. 사용자 모드를 선택한다. 2. 서버에 등록된 id/password를 입력한다. 3. 사용자 모드에 접속되는지 확인한다. | FR 1 |
| 2 | 로그인 실패: 존재하지 않는 아이디 | 1. 사용자 모드를 선택한다. 2. 서버에 등록되지 않은 id를 입력한다. 3. 안내문과 메인으로 돌아가지는지 확인한다. | FR 1 |
| 3 | 로그인 실패: 비밀번호 불일치 1회 | 1. 사용자 모드를 선택한다. 2. 서버에 등록된 id/틀린 password를 입력한다. 3. 안내문/다시 시도되는지 확인한다. | FR 1 |
| 4 | 로그인 실패: 비밀번호 불일치 5회 | 1. 사용자 모드를 선택한다. 2. 서버에 등록된 id/틀린 password를 5회 입력한다. 3. 휴면계정 안내문/메인으로 돌아가는지 확인한다. | FR 1 |
| 5 | 로그인 실패: 휴면계정 | 1. 사용자 모드를 선택한다. 2. 휴면계정으로 등록된 id/password를 입력한다. 3. 휴면계정 안내문/메인으로 돌아가는지 확인한다. | FR 1 |
| 6 | 회원가입 요청 성공 | 1. 회원가입 요청을 선택한다. 2. 올바른 id, password, 주민번호, 권한을 입력한다. 3. 안내문/메인으로 돌아가는지 확인한다. 4. DB pending-user에 등록되었는지 확인한다. | FR 2 |
| 7 | 회원가입 요청 실패: id 길이 | 1. 회원가입 요청을 선택한다. 2. 잘못된 길이의 id를 입력한다. 3. 안내문/다시 시도되는지 확인한다. | FR 2 |
| 8 | 회원가입 요청 실패: id 중복 | 1. 회원가입 요청을 선택한다. 2. DB user/pending-user에 등록된 아이디를 입력한다. 3. 안내문/다시 시도되는지 확인한다. | FR 2 |
| 9 | 회원가입 요청 실패: password 길이 | 1. 회원가입 요청을 선택한다. 2. 잘못된 길이의 password를 입력한다. 3. 안내문/다시 시도되는지 확인한다. | FR 2 |
| 10 | 회원가입 요청 실패: password 주민번호 | 1. 회원가입 요청을 선택한다. 2. 주민번호를 포함한 password를 입력한다. 3. 안내문/다시 시도되는지 확인한다. | FR 2 |
| 11 | 회원가입 요청 실패: 주민번호 길이 | 1. 회원가입 요청을 선택한다. 2. 잘못된 길이의 주민번호를 입력한다. 3. 안내문/다시 시도되는지 확인한다. | FR 2 |
| 12 | 회원가입 요청 실패: 권한레벨 | 1. 회원가입 요청을 선택한다. 2. R, W, RW가 아닌 권한레벨을 입력한다. 3. 안내문/다시 시도되는지 확인한다. | FR 2 |
| 13 | 관리자 모드: 돌아가기 | 1. 관리자 모드를 선택한다. 2. 메인으로 돌아가기(8)을 선택한다. 3. 메인으로 돌아가는지 확인한다. | FR 3 |
| 14 | 관리자 모드: 관리자 코드 변경 | 1. 관리자 모드를 선택한다. 2. 관리자 코드 변경(1)을 선택한다. 3. 새 관리자 코드를 입력한다. 4. 안내문/관리자 모드로 돌아가는지 확인한다. 5. DB admin에 업데이트 되었는지 확인한다. | FR 3 |
| 15 | 관리자 모드: 유저 정보 보기 | 1. 관리자 모드를 선택한다. 2. 가입한 유저 정보(2)를 선택한다. 3. DB user의 정보가 출력되는지 확인한다. 4. 관리자 모드로 돌아가는지 확인한다. | FR 3 |
| 16 | 관리자 모드: 하드 용량 추가: 성공 | 1. 관리자 모드를 선택한다. 2. 하드 용량 추가(3)를 선택한다. 3. 성공 안내문/관리자 모드로 돌아가는지 확인한다. 4. DB server의 업데이트된 capacityAll을 확인한다. | FR 3 |
| 17 | 관리자 모드: 하드 용량 추가: 실패 | 1. 하드 용량이 3회 추가된 상태로 세팅한다. 2. 관리자 모드를 선택한다. 3. 하드 용량 추가(3)를 선택한다. 4. 실패 안내문/관리자 모드로 돌아가는지 확인한다. 5. DB server의 업데이트되지 않은 capacityAll을 확인한다. | FR 3 |
| 18 | 관리자 모드: 대기 유저 관리: 돌아가기 | 1. 관리자 모드를 선택한다. 2. 대기중인 유저 정보(4)를 선택한다. 3. 돌아가기(0)을 입력한다. 4. 관리자 모드로 돌아가는지 확인한다. | FR 3 |
| 19 | 관리자 모드: 대기 유저 관리: 승인 | 1. 관리자 모드를 선택한다. 2. 대기중인 유저 정보(4)를 선택한다. 3. 등록된 유저 중 임의의 번호를 입력한다. 4. y를 입력한다. 5. 승인 안내문/해당 기능이 다시 실행되는지 확인한다. 6. 업데이트된 DB pending-user/user 테이블을 확인한다. | FR 3 |
| 20 | 관리자 모드: 대기 유저 관리: 거절 | 1. 관리자 모드를 선택한다. 2. 대기중인 유저 정보(4)를 선택한다. 3. 등록된 유저 중 임의의 번호를 입력한다. 4. n을 입력한다. 5. 거절 안내문/해당 기능이 다시 실행되는지 확인한다. 6. 업데이트된 DB pending-user 테이블을 확인한다. 7. 추가되지 않은 DB user 테이블을 확인한다. | FR 3 |
| 21 | 관리자 모드: 대기 유저 관리: 유저번호 실패 | 1. 관리자 모드를 선택한다. 2. 대기중인 유저 정보(4)를 선택한다. 3. 등록된 유저 번호가 아닌 임의의 번호를 입력한다. 4. 실패 안내문/해당 기능이 다시 실행되는지 확인한다. | FR 3 |
| 22 | 관리자 모드: 대기 유저 관리: 승인여부 실패 | 1. 관리자 모드를 선택한다. 2. 대기중인 유저 정보(4)를 선택한다. 3. 등록된 유저 중 임의의 번호를 입력한다. 4. y/n이 아닌 임의의 문자를 입력한다. 5. 실패 안내문/다시 시도되는지 확인한다. | FR 3 |
| 23 | 관리자 모드: 파일 업로드 성공 | 1. 관리자 모드를 선택한다. 2. 파일 업로드(5)를 선택한다. 3. FileManager.uploadFile이 실행되는지 확인한다. 4. 성공 안내문/관리자 모드로 돌아가는지 확인한다. 5. 서버에서 업로드된 파일을 확인한다. | FR 3 |
| 24 | 관리자 모드: 파일 업로드 실패 | 1. 관리자 모드를 선택한다. 2. 파일 업로드(5)를 선택한다. 3. FileManager.uploadFile이 실행되는지 확인한다. 4. 실패 안내문/관리자 모드로 돌아가는지 확인한다. 6. 서버에 파일이 업로드되지 않은 것을 확인한다. | FR 3 |
| 25 | 관리자 모드: 파일 삭제 성공 | 1. 관리자 모드를 선택한다. 2. 파일 삭제(6)을 선택한다. 3. FileManager.deleteFile이 실행되는지 확인한다. 4. 성공 안내문/관리자 모드로 돌아가는지 확인한다. 5. 서버에서 파일이 삭제되지 않은 것을 확인한다. | FR 3 |
| 26 | 관리자 모드: 파일 삭제 실패 | 1. 관리자 모드를 선택한다. 2. 파일 삭제(6)을 선택한다. 3. FileManager.deleteFile이 실행되는지 확인한다. 4. 실패 안내문/관리자 모드로 돌아가는지 확인한다. 4. 서버에서 삭제되지 않은 파일을 확인한다. | FR 3 |
| 27 | 관리자 모드: 파일 다운로드 성공 | 1. 관리자 모드를 선택한다. 2. 파일 다운로드(7)을 선택한다. 3. FileManager.downloadFile이 실행되는지 확인한다. 4. 성공 안내문/관리자 모드로 돌아가는지 확인한다. 5. 다운로드 폴더에 다운로드된 파일을 확인한다. | FR 3 |
| 28 | 관리자 모드: 파일 다운로드 실패 | 1. 관리자 모드를 선택한다. 2. 파일 다운로드(7)을 선택한다. 3. FileManager.downloadFile이 실행되는지 확인한다. 4. 실패 안내문/관리자 모드로 돌아가는지 확인한다. 5. 다운로드 폴더에 다운로드 되지 않은 것을 확인한다. | FR 3 |
| 29 | 유저정보 암호화 | 1. Util.encryptString에 임의의 문자열을 전달한다. 2. 문자열이 암호화되어 반환되는지 확인한다. | FR 4 |
| 30 | 유저정보 암호화: 관리자 모드에서 | 1. 관리자 모드를 선택한다. 2. 가입한 유저 정보(2)를 선택한다. 3. password, 주민번호가 암호화되어 출력되는지 확인한다. | FR 4 |
| 31 | 파일 매니저: 파일 업로드: 성공 | 1. FileManager.uploadFile을 호출한다. 2. 안내문과 리턴값을 확인한다. 3. 서버에 업로드된 파일을 확인한다. 4. DB server에 업데이트된 capacityUsed를 확인한다. | FR 5 |
| 32 | 파일 매니저: 파일 업로드: 실패 | 1. FileManager.uploadFile에 수용가능 용량보다 큰 파일을 전달한다. 2. 실패 안내문과 리턴값을 확인한다. 3. 서버에 파일이 업로드되지 않은 것을 확인한다. 4. DB server가 업데이트되지 않은 것을 확인한다. | FR 5 |
| 33 | 파일 매니저: 파일 삭제 | 1. FileManager.deleteFile을 호출한다. 2. 업로드된 파일 중 임의의 번호를 입력한다. 3. 성공 안내문과 리턴값을 확인한다. 4. 서버에서 파일이 삭제된 것을 확인한다. 5. DB server에 업데이트된 capacityUsed를 확인한다. | FR 5 |
| 34 | 파일 매니저: 파일 다운로드: 취소 | 1. FileManager.downloadFile을 호출한다. 2. 돌아가기(0)을 입력한다. 3. 메소드 종료와 리턴값을 확인한다. | FR 5 |
| 35 | 파일 매니저: 파일 다운로드: 성공 | 1. FileManager.downloadFile을 호출한다. 2. 업로드된 파일 중 임의의 번호를 입력한다. 3. 성공 안내문과 리턴값을 확인한다. 4. 다운로드 폴더에 다운로드된 파일을 확인한다. | FR 5 |
| 36 | 파일 매니저: 파일 출력과 선택: 돌아가기 | 1. FileManager.selectFromFiles를 호출한다. 2. 업로드된 파일들이 출력되는지 확인한다. 3. 돌아가기(0)을 입력한다. 4. 메소드 종료와 리턴값을 확인한다. | FR 5 |
| 37 | 파일 매니저: 파일 출력과 선택: 성공 | 1. FileManager.selectFromFiles를 호출한다. 2. 업로드된 파일들이 출력되는지 확인한다. 3. 업로드된 파일 중 임의의 번호를 입력한다. 4. 메소드 종료와 리턴값을 확인한다. | FR 5 |