

개발 실패 보고서

제출일 : 2022년 12월 24일

팀명	우연의보라		
참여인원	정철우, 고연재, 이보라, 박의용		
프로젝트 소개			
프로젝트 명	QT활용한 도서관 관리 프로그램		
활동일시	12/19(월) ~ 12/24(토)	장소	광주인력개발원 공학 1관 드론융합실
주요 주제	pyQT와 QT디자이너를 이용한 도서관 관리 프로그램 제작		
개발목적	클래스를 개인적으로 학습하여 활용 pygame 모듈을 활용		
개인별 임무분담	정철우	도서조회, 대여 및 반납 파트	- 도서 기본 정보 검색 - 도서 위치 검색 - 대여 가능 여부 (대여일,반납일,대여여부 확인) - 로그인 이후 도서관 내 도서 대여 or 반납 - 도서 반납 I/O 본인 회원정보 통해서 확인
	고연재	로그인 및 회원가입,	- 회원 가입시 회원정보 입력 I/O - 필수 입력 정보 누락 체크 - 로그인 시도시 회원 정보 일치 여부
	박의용	로그인, 메인화면 파트	- 도서 추천 - 도서관 시설 정보 - 도서관 공지 및 이벤트 - 페이지 이동
	이보라	코로나 이슈.	
개발환경	VScode / Pycharm / window OS		
구현기능	<ul style="list-style-type: none">- Python 기본 함수들의 응용- 사용자 지정 함수의 사용- 클래스 사용		

필수

본인이 작성한 코드는 다른 팀원에게 정확하게 설명해 줄 수 있도록 할 것.

설명 때 이해하지 못한건 대충 넘어가지 말고 바로바로 피드백 할 것.

학습 일정표

분류	항목		12/19			12/20			12/21			12/22			12/23			
학습	구현	회의	구조도 작성/일정표/계획서															
		개별 코드 작성				개별 코딩 작업												
		디버깅							디버깅				디버깅					
		코드 취합										코드 취합						
		제출																

개인 일정

정철우	특이 사항 없음
고연재	특이 사항 없음
박익용	특이 사항 없음
이보라	코로나 양성판정 12/19~

요구분석서

1.메인 화면

-도서 추천(임의의 도서 추천, 이미지 포함, 각 도서관 별 상이)

-도서관 시설 정보(각 도서관 별 상이)

-도서관 공지 및 이벤트(각 도서관 별 상이)

-페이지 이동(도서 조회, 대여 및 반납, 회원 가입)

2.로그인 및 회원가입

-회원가입시 회원정보 파일에 저장(이름, 거주지, ID, PW, 연락처,등등)

-회원가입시 필수 정보 누락시 가입 안되게 처리

-로그인시 회원정보 파일에 있는 정보와 일치해야 한다.(ID, PW)

3.도서 조회

-도서의 기본 정보(작가,출판사..등)

-도서의 위치(본인 도서관 어디에 소장되어 있는지 , 아니면 타 도서관에 있는지)

-대여가능 정보(대여 여부, 반납 예정일자)

	<p>4.대여 및 반납</p> <ul style="list-style-type: none"> -대여 서비스는 로그인이 필요하다. -현재 본인의 도서관에 소장되어 있는 도서중에서 대여 가능 -다른 회원이 대여중인 도서는 대여 불가능 -반납은 도서명 또는 등록번호로 반납가능 -반납시 대여리스트에서 반납 처리 한다. <p>*4개의 비슷한 프로그램을 만들어야 하는 만큼 각각의 페이지를 클래스화 하여 분업을 잘하십시오.</p> <p>*파일을 잘 설계하십시오.</p>
<p>파트설명</p>	<p>정철우</p> <p>-책 대여</p> <pre>def borrow booksEvent(self): if self.loginsignal ==False: QtWidgets.QMessageBox.warning(self.select, '알림', "로그인 후 이용해 주세요") return if self.data[self.index_list[self.selectrow]][11] != "O": print("대여불가") QtWidgets.QMessageBox.warning(self.select, '알림', "이미 대여된 도서입니다.") elif self.data[self.index_list[self.selectrow]][2] != self.libraryName: QtWidgets.QMessageBox.warning(self.select, '알림', "해당 도서관의 도서가 아닙니다.") elif self.UserInfo[3] == "" or self.UserInfo[3] == None or self.UserInfo[3] == " " : self.checking.lb_bookname.setText(self.data[self.index_list[sel f.selectrow]][5]) self.checking.showModal3() else: QtWidgets.QMessageBox.warning(self.select, '알림', "이미 대여한 도서가 있습니다.")</pre> <p>책 대여 파트로 로그인 시그널을 받아서 True 일 경우에만 작동하도록 맨처음 조건문으로 확인해서 F면 바로 멈추도록 함. 그리고 대여 상태인지 현재 도서관의 도서인지 확인하는 조건문으로 체크함.</p> <p>-책 반납</p> <pre>def return booksEvent(self): if self.UserInfo == None:</pre>

```

        QtWidgets.QMessageBox.warning(self.select, '알림',
        "로그인 후 이용해주세요")
        return
        if self.data[self.index_list[self.selectrow]][2] !=
        self.libraryName:
            QtWidgets.QMessageBox.warning(self.select, '알림', "해당
            도서관의 도서가 아닙니다.")
        elif self.UserInfo[3] !=
        self.data[self.index_list[self.selectrow]][4]:
            QtWidgets.QMessageBox.warning(self.select, '알림',
            "본인이 대여한 도서가 아닙니다.")
        elif self.data[self.index_list[self.selectrow]][11] == "X"
        and self.data[self.index_list[self.selectrow]][
            2] == self.libraryName and self.UserInfo[3] ==
        self.data[self.index_list[self.selectrow]][4]:
            QtWidgets.QMessageBox.warning(self.select, '알림', "반납
            완료")
            self.data[self.index_list[self.selectrow]][11] = "O"
            self.data[self.index_list[self.selectrow]][12] = "-"
            self.data[self.index_list[self.selectrow]][13] = "-"
            self.UserInfo[3] = ''
            self.find_UserInfo_book = None
            self.tableWidget.clear()
            df = pd.read_csv('join55.csv')
            count = 0
            for i in df["ID"]:
                count += 1
                if i == self.UserInfo[0]:
                    tind = count - 1
                    df.iat[tind, 3] = ''
                    df.to_csv("join55.csv", index=None)
                    # print("반납완료")
            else:
                QtWidgets.QMessageBox.warning(self.select, '알림',
                "대여한 도서가 아닙니다")
                # print("xxxxxxx")

```

반납 파트도 대여파트와 같이 로그인시에만 작동하도록 함.
반납 또한 본인이 빌린건지 빌렸던 도서관의 책인지 확인 후에 반납이 되도록 구성

추가로 반납은 대여하는 방식 그대로 반납 할 수도 있고.
따로 반납구간을 만들어서 본인이 빌린 책만 표시하게 해놓음.

```

def return_booksEvent 2(self):
    if self.loginsignal == False or self.UserInfo == None:
        QtWidgets.QMessageBox.warning(self, '알림', "반납은
        로그인을 해주세요")
    return
    if self.find_UserInfo_book[2] != self.libraryName:
        QtWidgets.QMessageBox.warning(self, '알림', "해당 도서관의

```

```

    도서가 아닙니다.")
    else:
        QtWidgets.QMessageBox.warning(self, '알림', "반납 완료")
        self.data[self.find_UserInfo_book[0]][11] = "O"
        self.data[self.find_UserInfo_book[0]][12] = "-"
        self.data[self.find_UserInfo_book[0]][13] = "-"
        self.UserInfo[3] = ' '
        self.find_UserInfo_book = None

        df = pd.read_csv('join55.csv')
        tind = -5
        count = 0
        for i in df["ID"]:
            count += 1
            if i == self.UserInfo[0]:
                tind = count - 1
            if tind < 0 :
                return
            df.iat[tind, 3] = ' '
            df.to_csv("join55.csv", index=None)
            print("반납완료")

```

한마디로 반납은 두가지 방식으로 가능하다.

책을 검색후에 본인이 빌린 책을 직접 골라서 반납이 가능하고,
두번째는 로그인 정보를 통해서 본인이 빌린 책을 바로 나타내서 반납이
가능하다.

-책 대여 체크 후 데이터 변경

```

def checked(self):
    if self.checking.borrowsignal == None:
        return
    if self.checking.borrowsignal:
        now = QDate.currentDate()
        to = QDate.currentDate().addDays(7)
        self.data[self.index_list[self.selectrow]][12] =
now.toString(Qt.ISODate)
        self.data[self.index_list[self.selectrow]][13] =
to.toString(Qt.ISODate)
        self.data[self.index_list[self.selectrow]][11] = "X"
        self.UserInfo[3] =
self.data[self.index_list[self.selectrow]][4]
        df = pd.read_csv('join55.csv')
        count = 0
        for i in df["ID"]:
            count += 1
            print(i)
            print(self.UserInfo)
            if str(i) == str(self.UserInfo[0]):
                tind = count - 1

```

```
df.iat[tind, 3] =
self.data[self.index_list[self.selectrow]][4]
df.to_csv("join55.csv", index=None)
print("책빌림")
self.checking.borrowsignal = False
```

책 대여시 확인창을 한번 더 띄운후에 결정하도록 했고,
마지막까지 확인 후에 회원정보 리스트에 책 등록번호를 추가함으로써
로그인시 본인이 어떤 책을 빌렸는지 확인 가능함.

-책 검색

```
def searching_bookname(self):
    if self.search_name.text() == "" or self.search_name.text()
    == None :
        QtWidgets.QMessageBox.warning(self, "검색오류", "키워드를
        입력해주세요")
        return
    else:
        # self.all_title = list(List_flow)
        self.fined = []
        # self.all_title = ['0', '연번', ' 소장도서관명', '
        자료실명', ' 등록번호', ' 도서명', ' 저자', ' 출판사', ' 출판연도', '
        청구기호', ' 데이터기준일자', '대여일자', '반납일자', '대여가능여부']
        self.all_title = list(self.alldata3.columns)

        self.text = self.search_name.text()
        # test = "저자"
        self.list = [" "]
        self.index_list = []
        for i in self.data:
            if i[5] == None :
                continue
            if str(self.text) in str(i[5]):
                self.index_list.append(i[0])
        self.showlist()
```

아무것도 입력하지 않을 경우 모든 책을 표시하기에는 너무 크기 때문에
알림창으로 키워드를 입력하라고 멈추게 함.

키워드 입력시엔 책 제목에 해당 키워드가 있는것만 책 제목이 아닌 총
데이터리스트의 인덱스(행순서)를 인덱스 리스트에 추가해서 인덱스
리스트를 통해서 총 데이터에서 필요한 데이터만 검색해서 표시하도록 함.

ps. 원래는 로그인 정보와 데이터 리스트를 메인에서 다 관리 하는 형태로
하려다가 로그인 파트에서 시간이 생각보다 오래 걸려서 해당 작업은 그냥
책데이터는 **search** 파트에서 선언해서 사용하고 회원데이터는 필요한
곳에서 그냥 바로 불러서 사용하는 방향으로 처리함.

		그리고 마지막 메인에서 종료하기 누를시에 지금까지 변경되고 사용되었었던 책 데이터를 저장함. (이게 23만권이다 보니까 시간이 오래 걸려서 프로그램을 완전히 종료하는 시점에서만 한번 동작하도록 함.
	고연재	<p>class Login_Page(Qmainwindow, form_class) 로그인 페이지 전체를 아우르는 클래스</p> <p>def __init__(self) 각 페이지에 들어가는 버튼의 기능을 입혀주는 함수</p> <p>def main(self) 로그인 페이지. 아이디와 비밀번호를 입력하고 메인 화면으로 넘어간다.</p> <p>def join(self) 회원가입 페이지. 회원가입 공란에 기능을 입혀주는 함수 아이디, 비밀번호, 비밀번호 확인, 이름, 전화번호,아이디 중복확인으로 구성돼있다.</p> <p>def MoveMainPage(self) 부모 클래스인 메인함수로 이동하는 식이 담겨있다.</p> <p>def join_issue(self) def join(self)에 들어갈 여러가지 추가 기능을 담고있다. 필수 기재사항 관리, 비밀번호와 비밀번호 확인 일치 여부, 회원가입 성공여부가 여기서 결정된다.</p> <p>def Id_inspection(self) def join(self)에 있는 아이디 중복확인 들어갈 기능을 담고있다. csv에 저장된 회원정보를 읽어서 같은 아이디가 있는지 없는지 검사한다.</p> <p>def get_through(self) csv에 저장된 회원정보를 토대로 def main(self)의 로그인 가능 여부를 판별해준다. csv에 기입한 아이디와 비밀번호가 있을 시 메세지 박스가 뜨면서 메인으로 이동하게 해준다. 아이디와 비밀번호가 다를 시 메인으로 이동하지 못 하게 막는다.</p> <p>def backmain(self) 로그인이 완료 되면 메인으로 이동하게 도와주는 기능을 담고 있다.</p>
	박의용	<pre> class MainWindow(QMainWindow, Ui_Form): 메인화면 구성 (도서관이름, 도서관시설현황, 도서추천, 공지이벤트) def showmain(self): 로그인 버튼 클릭시 로그인 화면 이동 </pre>

		<pre>def showsub(self): 도서 검색 및 도서 대여 반납 이동 def logout(self): 로그아웃 이동 def singa(self): 신가도서관이동 def chumdan(self): 첨단도서관이동 def flower(self): 이야기꽃도서관 def jangduk(self): 장덕도서관 def exit(self): 종료</pre>
	이보라	철우야 잘하자^^ 라고 김연수가 씀
소 감	정철우	<p>초반에 클래스간 데이터 공유부분에서 조금 막히긴 했지만 그외엔 위젯 사용법 등 내용을 몰라서 라기보다 사용해보지 않은 것들이라 막히는 부분을 제외하고는 무난했다고 생각함. 다만 데이터를 연동하면서 로그인신호나 로그인한 회원데이터를 연동할 때 서로 수정을 하다 보니까 자꾸 방식이 달라져서 에러가 자주 발생함. 그리고 csv파일 자체에 none값으로 들어간 데이터들이 있어서 프로그램이 돌다가 에러창도 없이 멈추는것때문에 하루 이상은 걸린듯 하다. import 개념을 좀더 이해해보려고 혼자 파일 3개 쓰면서 해봤는데 처음엔 이게 뭔가 했지만 그래도 이 짓 때문에 이해하는데 수월했고이번 프로젝트로 클래스 자체에 대한 이해는 좀 더 상승했다고 본다.</p>
	고연재	<p>가장 많은걸 배운 프로젝트였다. 교수님들 말처럼 이제부터 진짜 시작이라는 생각이 든다. 클래스간에 연결이 어려우면서도 자유도가 이렇게나 높구나라는 생각도 동시에 들었다. 계속 공부해보고 부지런히 써보면서 클래스의 활용법을 계속 알아가야겠다는 생각이 지속적으로 들었다.</p> <p>각 클래스간 연동이 되는지 확인하고 로그인 상태를 유지토록 하는게 가장 어려웠다. 팀장의 도움으로 어느정도는 이해했지만 앞으로 더 써보고 연습해봐야 될 것 같다.</p> <p>qt와 qt designer도 흥미로웠다. 개발원에 들어오기 전부터 우리의 삶에서 떼 수 없는 관계가 돼버린 프로그램들이 어떻게 구동이 되는지 항상 궁금했는데 이런 것들을 활용해서 되는거라는걸 처음 알게 됐다.</p> <p>정말 값진 5일이였다. 이제 정말 내가 하고 있는 것의 방향성이 보인다. 다음 주부터는 git을 활용한다. 이 모든 것들이 코딩의 기본이라는 걸 잊지 말고 허투루 공부하는 일이 없도록 해야겠다.</p>
	박의용	<p>PYQT, QT DESIGNER, PYTHON CLASS 종합 선물세트 과제였다. 가장 많이 한 고민은 ‘어떻게 하면 각자 팀원들이 만든 코드를 취합할 수 있을까?’였다. 팀장이 아닌 팀원이었지만, ‘내가 만약 팀장이라면, 내가 만약 혼자서 이 프로젝트를 진행했다면 어떻게 하였을까’ 라는 마음가짐을 가지고 임했고, 최대한 노력하였다. 사실 어렸을 때부터</p>

		책을 좋아했고, 마침 도서관 관련 프로젝트라서 더욱 열심히 했다. 결과물이 어떨든 최선을 다했고, 팀원들끼리의 코드 취합이 어떻게든 되서 뿌듯했다. 다음 프로젝트 때는 이번에 배운 것들을 토대로, 더욱 정진하고 싶은 마음이 들었다.
	이보라	
실패 사유	한명의 부재로 시간이 부족했다기 보다 본인이 맡은 파트를 본인 힘으로 최대한 해보도록 하려고 바로바로 알려주지 않고 시간 주는걸 반복하다 보니 제시간에 끝내지 못했다. 막히면 바로바로 알려주는것보다 본인이 맡은 파트에 대해서 최대한 생각해보도록 하는게 더 좋다고 판단해서 이렇게 진행하게 되었는데 결국 마감기한을 지키지 못함.	