

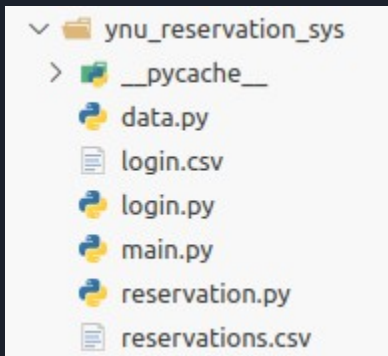


Hello 회의실 예약 시스템

DR-01428 서연우

프로그램 소개

1. 목적: 회의실 예약 시스템 구현
2. 파일 구조



3. 주요 기능: 예약, 조회, 수정, 취소, 연장
4. 사용자 인터페이스: Tkinter를 활용한 GUI 로그인 /회원가입 시스템
5. 데이터는 login.csv, reservations.csv 파일로부터 불러옴



모듈 소개

1. main.py

: 메인 프로그램, 로그인과 회원가입, 터미널 출력

2. login.py

: 로그인 관련 기능, 사용자 인증 및 등록

3. reservation.py

: 예약 관련 기능, 예약 생성, 조회, 수정, 취소, 연장

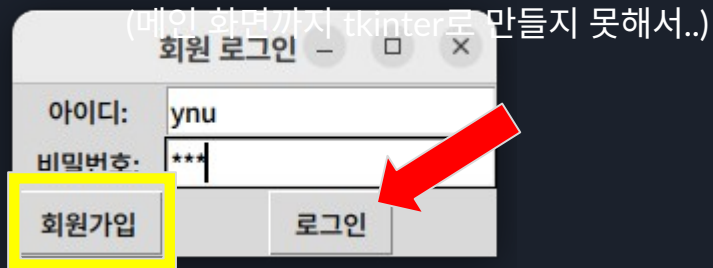
4. data.py

: 데이터 로드 및 저장, CSV 파일 사용

[1단계] 로그인 및 회원가입

로그인하면 '터미널 창'으로 넘어간 후 예약 시스템 시작

(메인 화면까지 tkinter로 만들지 못해서..)

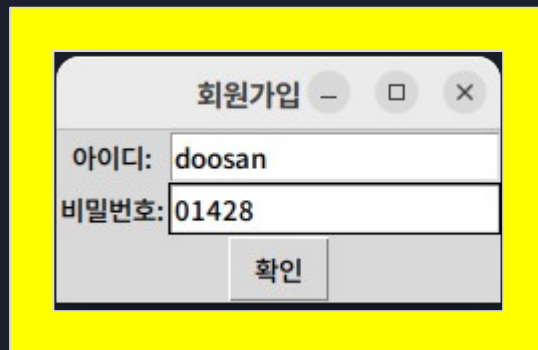


회원 로그인

아이디: ynu

비밀번호: ***

회원가입 로그인

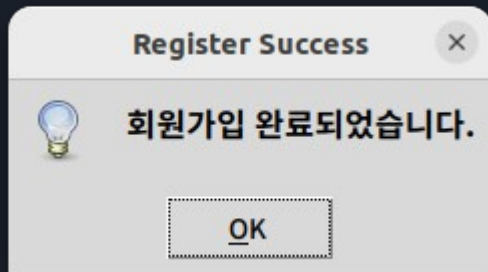


회원가입

아이디: doosan

비밀번호: 01428

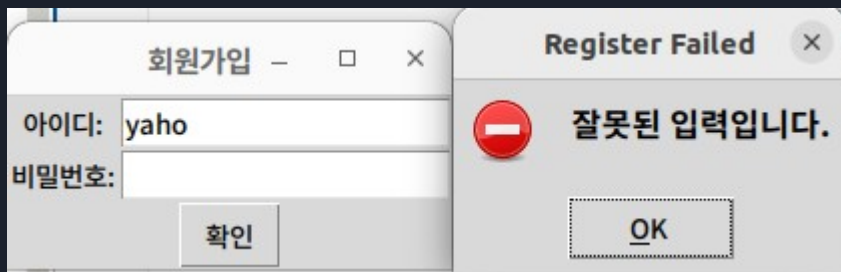
확인



Register Success

회원가입 완료되었습니다.

OK



회원가입

아이디: yaho

비밀번호:

확인

Register Failed

잘못된 입력입니다.

OK

다 입력하지 않고 빈칸인 상태에서 '확인'을 누른 경우, 경고 메시지 출력

[2단계] 메인 메뉴

최종 업데이트: 2024-08-02

=====

== Hello 회의실 예약 프로그램 ==

=====

- | | |
|-------|-------|
| 1. 예약 | 4. 취소 |
| 2. 조회 | 5. 연장 |
| 3. 수정 | 0. 종료 |

선택:

```
choice = int(input("선택: "))
```

```
# csv 파일 불러오기
```

```
reservations = load_reservations()
```

```
if choice == 1:
```

```
    make_reservation(reservations)
```

```
elif choice == 2:
```

```
    view_reservations(reservations)
```

```
elif choice == 3:
```

```
    modify_reservation(reservations)
```

```
elif choice == 4:
```

```
    cancel_reservation(reservations)
```

```
elif choice == 5:
```

```
    extend_reservation(reservations)
```

```
elif choice == 0:
```

```
    save_reservations(reservations)
```

```
    print("이용해주셔서 감사합니다.")
```

```
    break
```

```
# 맞게 입력할 때까지 while문 반복
```

```
else:
```

```
    print("잘못된 선택입니다. 다시 선택해주세요.")
```

reservations.csv를 불러오고, 번호를 선택하면 reservation.py의 해당 함수를 실행하도록 설계

[3단계] 조회 기능

reservations.csv 데이터 출력

선택: 2

예약 번호: 1, 회의실 번호: 1, 시작 시간: 2024-08-01 08:00, 끝 시간: 2024-08-01 10:00, 참석 인원: 12

예약 번호: 2, 회의실 번호: 2, 시작 시간: 2024-08-01 18:00, 끝 시간: 2024-08-01 19:00, 참석 인원: 8

예약 번호: 3, 회의실 번호: 1, 시작 시간: 2024-08-05 09:00, 끝 시간: 2024-08-05 10:00, 참석 인원: 10

예약 번호: 4, 회의실 번호: 2, 시작 시간: 2024-08-06 09:00, 끝 시간: 2024-08-06 10:00, 참석 인원: 7

예약 번호: 5, 회의실 번호: 3, 시작 시간: 2024-08-07 09:00, 끝 시간: 2024-08-07 10:00, 참석 인원: 3

예약 번호: 6, 회의실 번호: 1, 시작 시간: 2024-08-17 12:00, 끝 시간: 2024-08-17 13:45, 참석 인원: 15

예약 번호: 7, 회의실 번호: 2, 시작 시간: 2024-09-26 17:00, 끝 시간: 2024-09-26 20:25, 참석 인원: 9

조회

```
def view_reservations(reservations):
```

```
    for index, reservation in enumerate(reservations):
```

```
        print(f"예약 번호: {index + 1}, 회의실 번호: {reservation[0]}, 시작 시간: {reservation[1]}, 끝 시간: {reservation[2]}, 참석 인원: {reservation[3]}")
```

[3단계] 예약 기능

선택: 1

예약 시작 시간 (YYYY-MM-DD HH:MM): 2024-08-19 13:00

예약 끝 시간 (YYYY-MM-DD HH:MM): 2024-08-19 15:00

참석 인원: 8

[2번] '중 회의실'이 예약되었습니다.

예약 시작 시간, 끝 시간, 참석 인원을
사용자에게 입력 받고, 명수에 맞게 회의실
자동 배정

예약 시간이 현재 시간 이후인지,
예약 끝 시간이 시작 시간 이후인지,
예약 시간이 겹치는지 체크

```
def make_reservation(reservations):
    while True:
        start_time_str = input("예약 시작 시간 (YYYY-MM-DD HH:MM): ")
        start_time = datetime.strptime(start_time_str, "%Y-%m-%d %H:%M")
        if start_time > datetime.now():
            break
        else:
            print("예약 시작 시간은 현재 시간 이후여야 합니다.\n")

    while True:
        end_time_str = input("예약 끝 시간 (YYYY-MM-DD HH:MM): ")
        end_time = datetime.strptime(end_time_str, "%Y-%m-%d %H:%M")
        if end_time > start_time:
            break
        else:
            print("예약 끝 시간은 시작 시간 이후여야 합니다.\n")

    for res in reservations:
        res_start = datetime.strptime(res[1], "%Y-%m-%d %H:%M")
        res_end = datetime.strptime(res[2], "%Y-%m-%d %H:%M")
        if (start_time < res_end and end_time > res_start):
            print("예약 시간이 겹칩니다. 다른 시간으로 시도해 주세요.")
            return

    attendees = int(input("참석 인원: "))
    room = assign_room(attendees)

    reservations.append([list(room.keys())[0], start_time_str, end_time_str, attendees])
    print(f"[{list(room.keys())[0]}번] '{list(room.values())[0]}'이 예약되었습니다.")
    save_reservations(reservations)
```

[4단계] 수정 기능

예약과 코드구조 거의 동일 ->

새 예약 시간이 현재 시간 이후인지,
새 예약 끝 시간이 시작 시간 이후인지,
새 예약 시간이 겹치는지 체크

```
if 0 <= index < len(reservations):
```

```
else:
```

```
    print("잘못된 번호입니다.")
```

차이점!

수정하는 과정이므로, 사용자가 입력한 예약 번호가 예약 리스트 내에서 유효한 범위에 속하는지 확인 필요

선택: 3

예약 번호: 1, 회의실 번호: 1, 시작 시간: 2024-08-01 08:00, 끝 시간: 2024-08-01 10:00, 참석 인원: 12

예약 번호: 2, 회의실 번호: 2, 시작 시간: 2024-08-01 18:00, 끝 시간: 2024-08-01 19:00, 참석 인원: 8

예약 번호: 3, 회의실 번호: 1, 시작 시간: 2024-08-05 09:00, 끝 시간: 2024-08-05 10:00, 참석 인원: 10

예약 번호: 4, 회의실 번호: 2, 시작 시간: 2024-08-06 09:00, 끝 시간: 2024-08-06 10:00, 참석 인원: 7

예약 번호: 5, 회의실 번호: 3, 시작 시간: 2024-08-07 09:00, 끝 시간: 2024-08-07 10:00, 참석 인원: 3

예약 번호: 6, 회의실 번호: 1, 시작 시간: 2024-08-17 12:00, 끝 시간: 2024-08-17 13:45, 참석 인원: 15

예약 번호: 7, 회의실 번호: 2, 시작 시간: 2024-09-26 17:00, 끝 시간: 2024-09-26 20:25, 참석 인원: 9

예약 번호: 8, 회의실 번호: 2, 시작 시간: 2024-08-08 09:00, 끝 시간: 2024-08-08 11:00, 참석 인원: 8

수정할 예약 번호: 8

새 예약 시작 시간 (YYYY-MM-DD HH:MM): 2024-08-13 09:00

새 예약 끝 시간 (YYYY-MM-DD HH:MM): 2024-08-13 11:00

새 참석 인원: 8

예약이 수정되었습니다.

== Hello 회의실 예약 프로그램 ==

- | | |
|-------|-------|
| 1. 예약 | 4. 취소 |
| 2. 조회 | 5. 연장 |
| 3. 수정 | 0. 종료 |

선택: 2

예약 번호: 1, 회의실 번호: 1, 시작 시간: 2024-08-01 08:00, 끝 시간: 2024-08-01 10:00, 참석 인원: 12

예약 번호: 2, 회의실 번호: 2, 시작 시간: 2024-08-01 18:00, 끝 시간: 2024-08-01 19:00, 참석 인원: 8

예약 번호: 3, 회의실 번호: 1, 시작 시간: 2024-08-05 09:00, 끝 시간: 2024-08-05 10:00, 참석 인원: 10

예약 번호: 4, 회의실 번호: 2, 시작 시간: 2024-08-06 09:00, 끝 시간: 2024-08-06 10:00, 참석 인원: 7

예약 번호: 5, 회의실 번호: 3, 시작 시간: 2024-08-07 09:00, 끝 시간: 2024-08-07 10:00, 참석 인원: 3

예약 번호: 6, 회의실 번호: 1, 시작 시간: 2024-08-17 12:00, 끝 시간: 2024-08-17 13:45, 참석 인원: 15

예약 번호: 7, 회의실 번호: 2, 시작 시간: 2024-09-26 17:00, 끝 시간: 2024-09-26 20:25, 참석 인원: 9

예약 번호: 8, 회의실 번호: 2, 시작 시간: 2024-08-13 09:00, 끝 시간: 2024-08-13 11:00, 참석 인원: 8

[5단계] 취소 기능

```
index = int(input("취소할 예약 번호: ")) - 1
if 0 <= index < len(reservations):
    reservations.pop(index) # 선택한 인덱스 요소 제거
```

입력한 예약 번호의 인덱스 값을 저장하여
조건문을 통해 예약 목록의 길이보다 작은지 확인 후,
조건이 참이라면 해당 인덱스에 위치한 예약 리스트에서 제거

선택: 4

예약 번호: 1, 회의실 번호: 1, 시작 시간: 2024-08-01 08:00, 끝 시간: 2024-08-01 10:00, 참석 인원: 12
예약 번호: 2, 회의실 번호: 2, 시작 시간: 2024-08-01 18:00, 끝 시간: 2024-08-01 19:00, 참석 인원: 8
예약 번호: 3, 회의실 번호: 1, 시작 시간: 2024-08-05 09:00, 끝 시간: 2024-08-05 10:00, 참석 인원: 10
예약 번호: 4, 회의실 번호: 2, 시작 시간: 2024-08-06 09:00, 끝 시간: 2024-08-06 10:00, 참석 인원: 7
예약 번호: 5, 회의실 번호: 3, 시작 시간: 2024-08-07 09:00, 끝 시간: 2024-08-07 10:00, 참석 인원: 3
예약 번호: 6, 회의실 번호: 1, 시작 시간: 2024-08-17 12:00, 끝 시간: 2024-08-17 13:45, 참석 인원: 15
예약 번호: 7, 회의실 번호: 2, 시작 시간: 2024-09-26 17:00, 끝 시간: 2024-09-26 20:25, 참석 인원: 9
예약 번호: 8, 회의실 번호: 2, 시작 시간: 2024-08-13 09:00, 끝 시간: 2024-08-13 11:00, 참석 인원: 8
취소할 예약 번호: 8

예약이 취소되었습니다.

== Hello 회의실 예약 프로그램 ==

- ```

1. 예약 4. 취소
2. 조회 5. 연장
3. 수정 0. 종료

```

선택: 2

예약 번호: 1, 회의실 번호: 1, 시작 시간: 2024-08-01 08:00, 끝 시간: 2024-08-01 10:00, 참석 인원: 12  
예약 번호: 2, 회의실 번호: 2, 시작 시간: 2024-08-01 18:00, 끝 시간: 2024-08-01 19:00, 참석 인원: 8  
예약 번호: 3, 회의실 번호: 1, 시작 시간: 2024-08-05 09:00, 끝 시간: 2024-08-05 10:00, 참석 인원: 10  
예약 번호: 4, 회의실 번호: 2, 시작 시간: 2024-08-06 09:00, 끝 시간: 2024-08-06 10:00, 참석 인원: 7  
예약 번호: 5, 회의실 번호: 3, 시작 시간: 2024-08-07 09:00, 끝 시간: 2024-08-07 10:00, 참석 인원: 3  
예약 번호: 6, 회의실 번호: 1, 시작 시간: 2024-08-17 12:00, 끝 시간: 2024-08-17 13:45, 참석 인원: 15  
예약 번호: 7, 회의실 번호: 2, 시작 시간: 2024-09-26 17:00, 끝 시간: 2024-09-26 20:25, 참석 인원: 9

## [5단계] 연장 기능

```
old_time =
datetime.strptime(reservations[index][2],
"%Y-%m-%d %H:%M") # 끝 시간에 추가
new_time =
old_time + timedelta(minutes=int(extra_time))
```

선택한 예약 번호의 끝 시간(reservations[index][2])를  
문자열로 가져와 datetime 객체로 변환하고, 사용자가  
입력한 연장 시간을 분 단위로 더해 새로운 끝 시간 계산

선택: 5

예약 번호: 1, 회의실 번호: 1, 시작 시간: 2024-08-01 08:00, 끝 시간: 2024-08-01 10:00, 참석 인원: 12  
예약 번호: 2, 회의실 번호: 2, 시작 시간: 2024-08-01 18:00, 끝 시간: 2024-08-01 19:00, 참석 인원: 8  
예약 번호: 3, 회의실 번호: 1, 시작 시간: 2024-08-05 09:00, 끝 시간: 2024-08-05 10:00, 참석 인원: 10  
예약 번호: 4, 회의실 번호: 2, 시작 시간: 2024-08-06 09:00, 끝 시간: 2024-08-06 10:00, 참석 인원: 7  
예약 번호: 5, 회의실 번호: 3, 시작 시간: 2024-08-07 09:00, 끝 시간: 2024-08-07 10:00, 참석 인원: 3  
예약 번호: 6, 회의실 번호: 1, 시작 시간: 2024-08-17 12:00, 끝 시간: 2024-08-17 13:45, 참석 인원: 15  
예약 번호: 7, 회의실 번호: 2, 시작 시간: 2024-09-26 17:00, 끝 시간: 2024-09-26 20:25, 참석 인원: 9

연장할 예약 번호: 7

연장할 시간 (분 단위): 30

예약이 연장되었습니다.

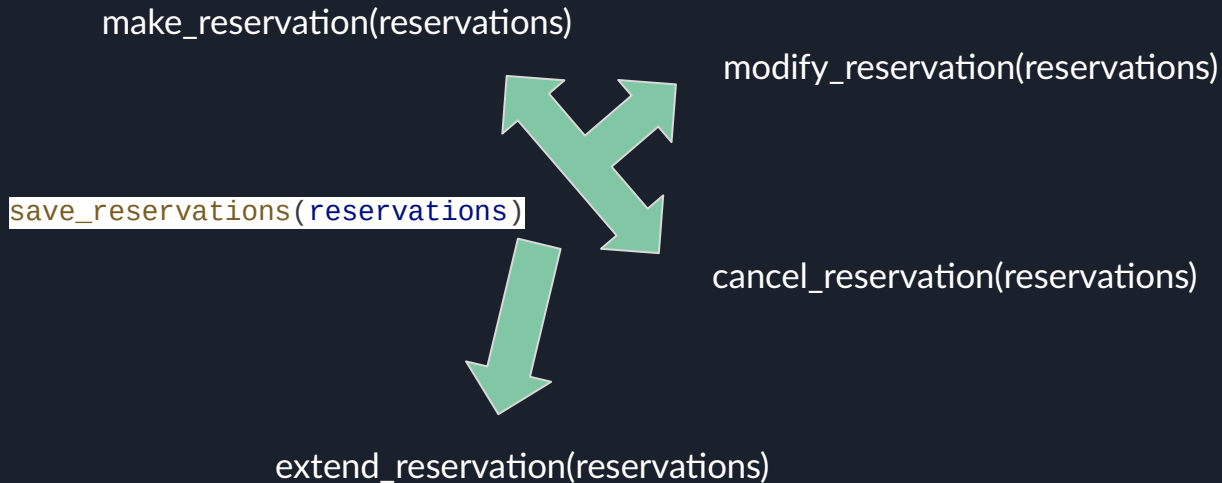
== Hello 회의실 예약 프로그램 ==

- ```
*****  
1. 예약      4. 취소  
2. 조회      5. 연장  
3. 수정      0. 종료  
*****
```

선택: 2

예약 번호: 1, 회의실 번호: 1, 시작 시간: 2024-08-01 08:00, 끝 시간: 2024-08-01 10:00, 참석 인원: 12
예약 번호: 2, 회의실 번호: 2, 시작 시간: 2024-08-01 18:00, 끝 시간: 2024-08-01 19:00, 참석 인원: 8
예약 번호: 3, 회의실 번호: 1, 시작 시간: 2024-08-05 09:00, 끝 시간: 2024-08-05 10:00, 참석 인원: 10
예약 번호: 4, 회의실 번호: 2, 시작 시간: 2024-08-06 09:00, 끝 시간: 2024-08-06 10:00, 참석 인원: 7
예약 번호: 5, 회의실 번호: 3, 시작 시간: 2024-08-07 09:00, 끝 시간: 2024-08-07 10:00, 참석 인원: 3
예약 번호: 6, 회의실 번호: 1, 시작 시간: 2024-08-17 12:00, 끝 시간: 2024-08-17 13:45, 참석 인원: 15
예약 번호: 7, 회의실 번호: 2, 시작 시간: 2024-09-26 17:00, 끝 시간: 2024-09-26 20:55, 참석 인원: 9

[6단계] 저장 구조 (reservation.py)



변화를 주는 함수들 마지막 라인에 항상 data.py의
`save_reservations(reservations)` 함수 불러와서 바로 바로 저장

코드 실행 영상

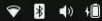
현재 활동

Visual Studio Code

8월 2일 06:52

0:00

EN



파일 편집 선택 영역 보기 이동 실행 터미널 도움말

main.py x reservation.py login.py data.py

회의실 > main.py > terminal

```
1 '''
2 파일명: 회의실 (main.py, login.py, reservation.py, data.py)
3 프로그램: Hello 회의실 예약 시스템
4 작성자: 서연우
5 작성일: 2024-08-02
6
7 test할 경우 아이디/비번 예시: user1/pass1
8 '''
9 from datetime import *
10 import tkinter as tk
11 from tkinter import messagebox
12
13 from login import load_login_data, authenticate, register_user
14 from reservation import *
15 from data import load_reservations
16
17 '''
18 로그인
19 '''
20
21 def login_app():
22     global username_entry, password_entry, root
23     root = tk.Tk() # 창 생성
24     root.title("회원 로그인")
25
26     tk.Label(root, text="아이디:").grid(row=0, column=0)
27     username_entry = tk.Entry(root) # 값 입력받기
28     username_entry.grid(row=0, column=1)
29
30     tk.Label(root, text="비밀번호:").grid(row=1, column=0)
31     password_entry = tk.Entry(root, show="*") # 비밀번호 숨기게
32     password_entry.grid(row=1, column=1)
33
34     # 로그인 버튼
35     login_button = tk.Button(root, text="로그인", command=try_login)
36     login_button.grid(row=2, column=1)
37
38     # 회원가입 버튼
39     register_button = tk.Button(root, text="회원가입", command=register_app)
40     register_button.grid(row=2, column=0)
41
42     root.mainloop() # 이벤트 메시지 루프
43
44 def try_login():
45     username = username_entry.get() # 입력값 가져오기
46     password = password_entry.get()
47     users = load_login_data() # csv 파일 불러오기
48
```

Python 디버거: 현재 파일 (python_work)

줄 122, 열 26 공백: 4 UTF-8 LF Python 3.10.12 64-bit



감사합니다