

항공권 예매 & 조회 사이트

Java 팀 프로젝트 발표

3조 - 김용현 노유정 오은지



Contents.

1. 팀원별 담당 역할

2. 프로젝트 개요

3. 개발 환경 및 사용 도구

4. 와이어 프레임 & 플로우 차트

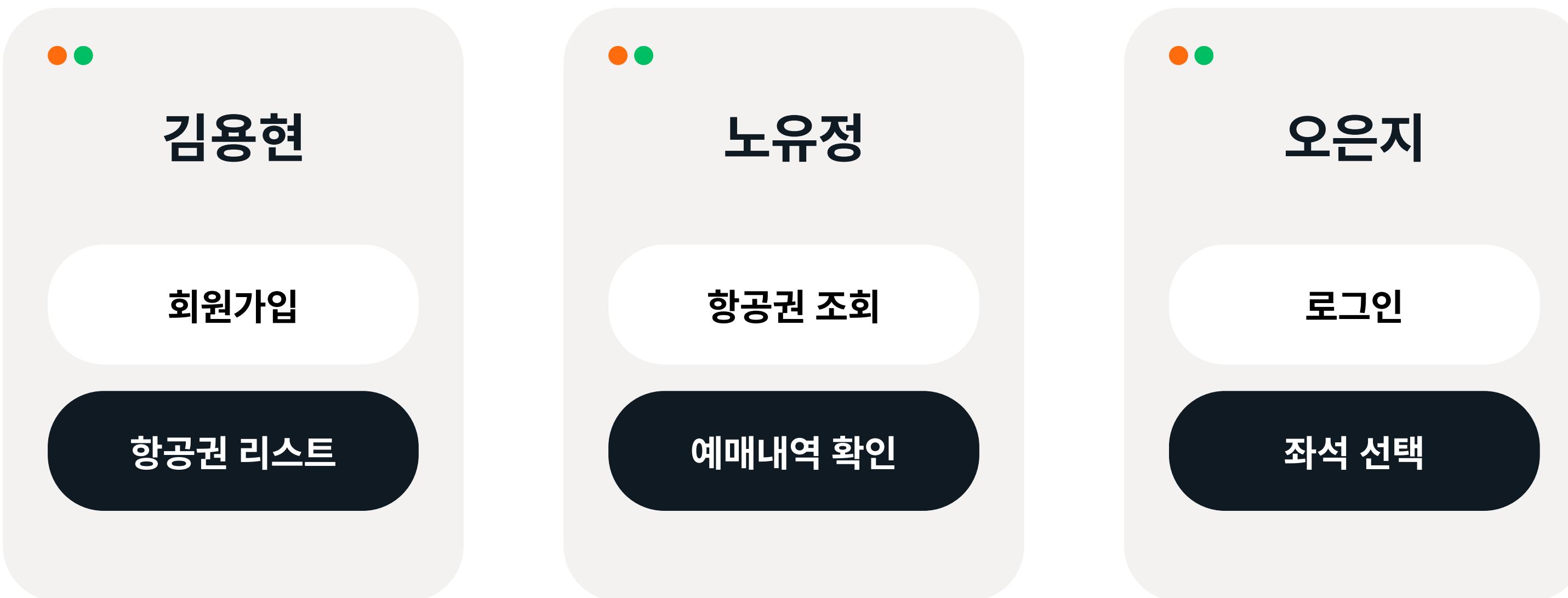
5. 화면별 기능 설명

6. 프로젝트 시연

7. 느낀점

팀원별 담당 역할

주제 선정 -> 레퍼런스 서치 -> 제작할 화면 리스트업 -> 플로우 차트 -> 와이어프레임
-> 각자 구현할 화면 담당 선정 -> 각자 코딩 + 코드 취합 + 테스트



프로젝트 개요

- 주제는 항공권 예매를 할 수 있는 사이트입니다.
- 기능은 회원가입, 로그인후 항공권을 선택하여 유저 정보 및 항공권 정보를 추출 하여 예매 내역 확인과 다운로드가 가능합니다.

주제

회원 가입한 유저에게 항공권 **예매 + 조회하는 서비스** 제공

기능

로그인/로그아웃/목적지, 날짜, 인원수 입력 후 조회
항공권 리스트/좌석 선택/예매내역 확인, 다운로드

특징

유저 정보 관리, 유저의 예매 정보 관리

개발환경 & 사용도구

개발 도구 JDK17, Eclipse
개발 언어 java
협업 도구 notion, github



The screenshot shows a GitHub repository named "three". The repository is public and has 131 commits from user "oille12". The commits are listed in reverse chronological order, showing updates to files like "flight_booking_0430" through "flight_booking_0508", "Seat.java", and "README.md". The repository also contains a ZIP file and a JAR file. The README contains a note about a flight booking system. The repository has 1 watching and 0 forks. It includes sections for About, Releases, Packages, Contributors, Languages, and Suggested workflows.

Commits

- oille12 Add files via upload · 2729f5a · 34 minutes ago · 131 Commits
- flight_booking_0430 · Update and rename UserInfo.ja... · 3 hours ago
- flight_booking_0503 · Rename menuBar.java to menu... · 3 hours ago
- flight_booking_0503_2 · Update and rename UserInfo.ja... · 3 hours ago
- flight_booking_0504 · Update and rename UserInfo.ja... · 3 hours ago
- flight_booking_0504_2 · Update and rename UserInfo.ja... · 3 hours ago
- flight_booking_0505 · Add files via upload · 2 days ago
- flight_booking_0507 · Update Seat.java · 1 hour ago
- flight_booking_0508 · Add files via upload · 34 minutes ago
- logo · Rename tway.png.png to tway.p... · 3 hours ago
- project_3 · Add files via upload · last week
- README.md · Create README.md · 3 hours ago
- flight_booking_0508.zip · Add files via upload · 35 minutes ago
- jdatepicker-1.3.2.jar · Add files via upload · last week

README

3조 java프로젝트 [항공권 예매 시스템]

Languages

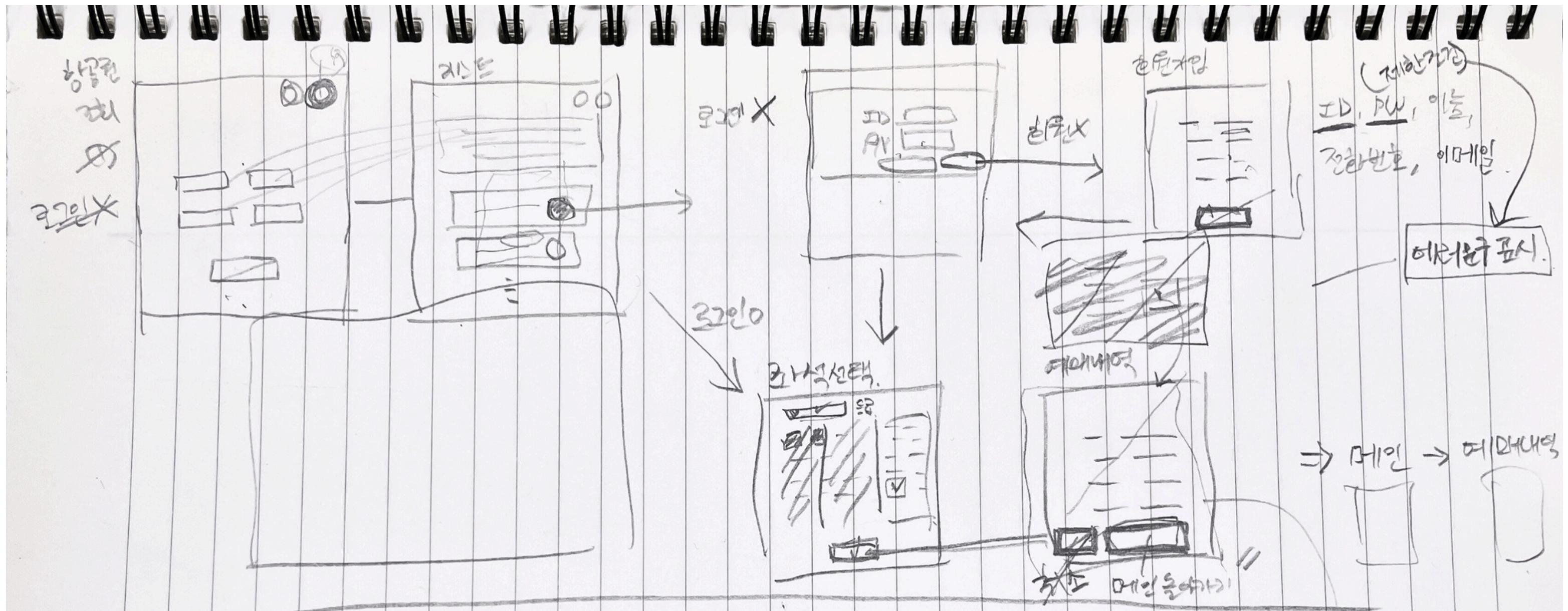
Java 100.0%

Suggested workflows

Based on your tech stack

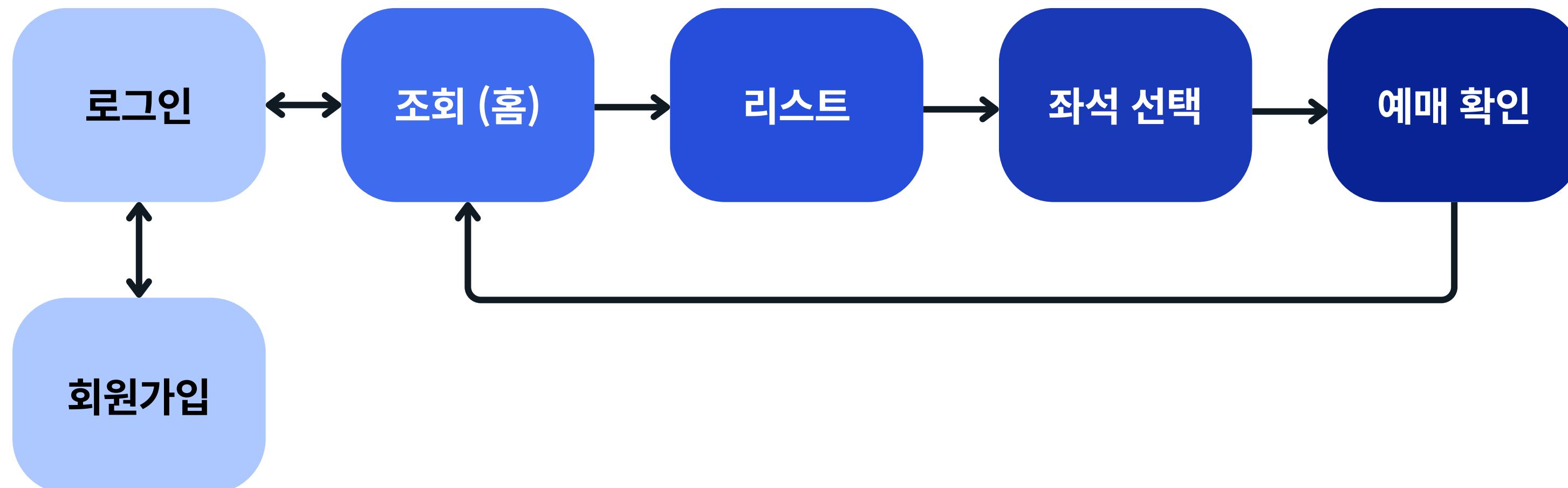
와이어 프레임

러프한 와이어프레임 스케치 후 흐름과 구현할 화면 구조 이해



플로우 차트

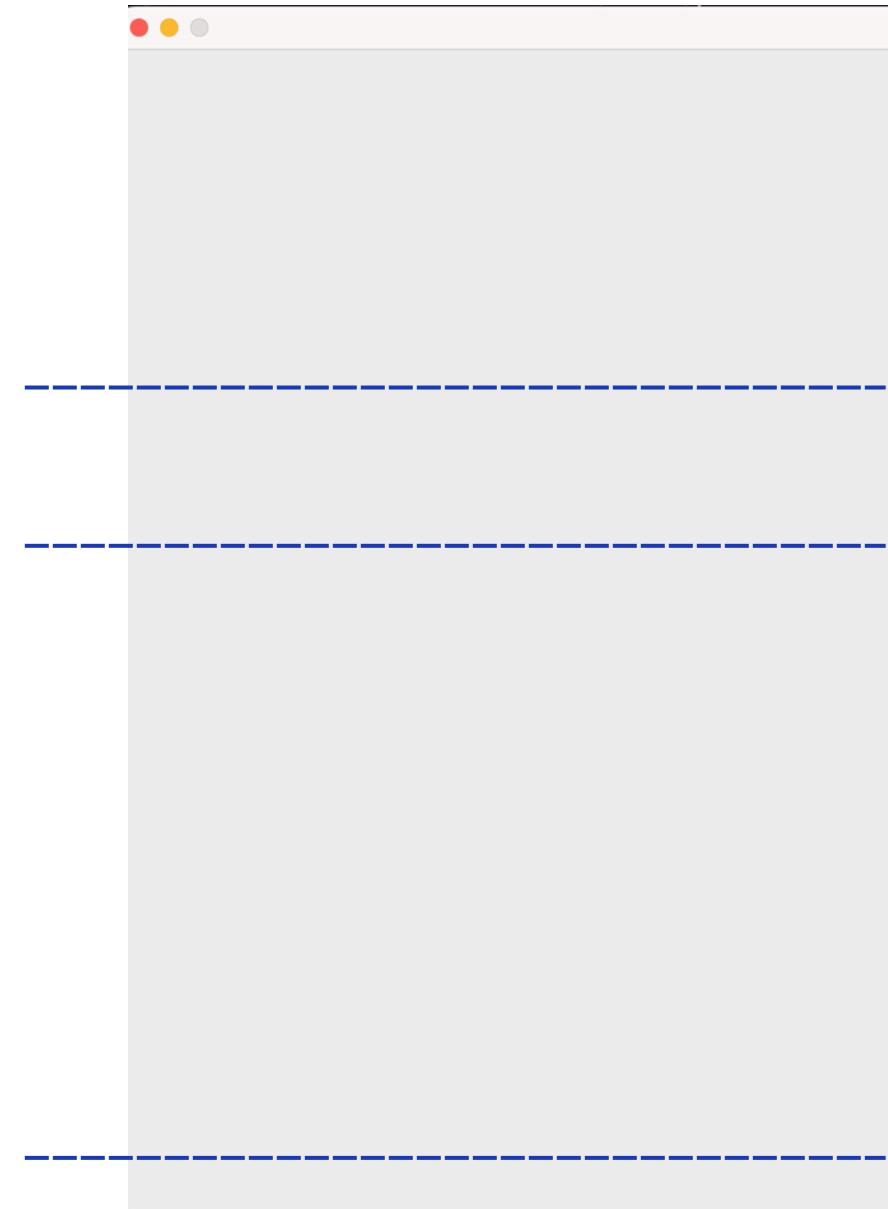
플로우 차트로 시스템의 흐름을 정리하여 작업 중 혼선 방지



화면별 기능 설명

1. 로그인

로고 이미지



아이디 / 비밀번호 입력창

Flight booking

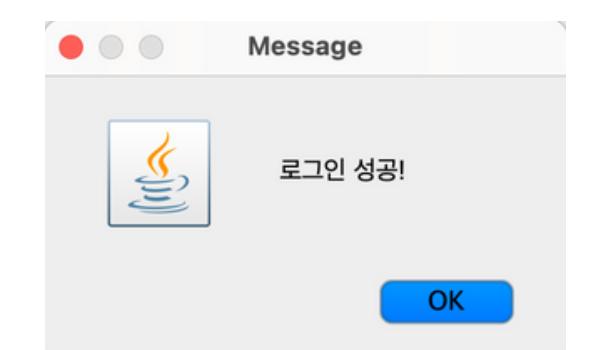
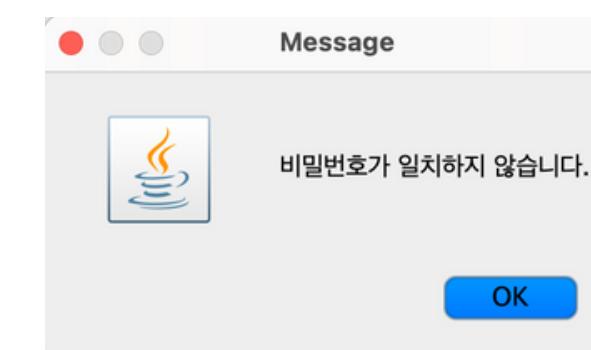
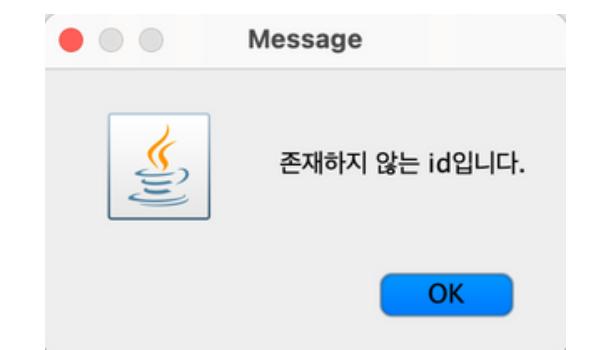
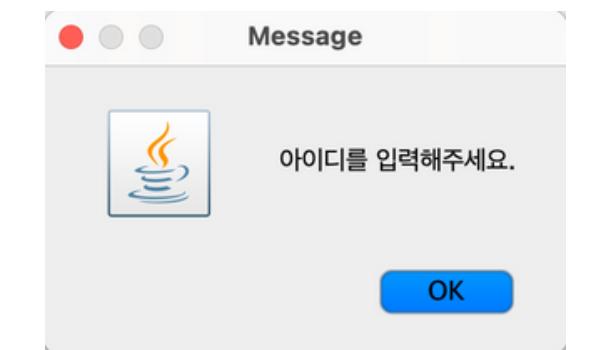
ID

Password

Login

Join

로그인 / 회원가입 버튼



1. 로그인

```
btnLogin.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        // 아이디 입력이 비었을 때
        if (idTf.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null, "아이디를 입력해주세요.");
        } else if (MembershipChecker.isIdExists(idTf.getText())) { // member.txt 파일 읽어서 확인
            // member.txt 파일에 존재하는 아이디일 때
            if (pwPf.getPassword().length == 0) {
                // 비밀번호가 비었을 때
                JOptionPane.showMessageDialog(null, "비밀번호를 입력해주세요.");
            } else {
                // 비밀번호가 입력되었을 때
                String enteredPassword = new String(pwPf.getPassword());
                String storedPassword = getPasswordForId(idTf.getText()); // member.txt 파일에서 해당 아이디의 비밀번호 가져오기
                if (storedPassword != null && MembershipChecker.isPasswordCorrect(enteredPassword, storedPassword)) {
                    // members.txt에서 id, name, phoneNumber 정보를 userInfo에 저장하기
                    extractMembers("members.txt", idTf.getText());

                    // membersReservation.txt에 저장된 id가 있는지 확인해서 있으면 예약정보 저장
                    extractMembersReservation(userInfo, "membersReservation.txt", idTf.getText());

                    // 입력한 비밀번호가 member.txt에 존재하는 해싱 비밀번호와 일치할 때
                    JOptionPane.showMessageDialog(null, "로그인 성공!");
                    setVisible(false);
                    idTf.setText("");
                    pwPf.setText("");
                    // !! ReservationPanel 화면으로 넘어가는 코드를 여기에 추가 !!
                    new ReservationPanel(userInfo);

                } else {
                    // 입력한 비밀번호가 member.txt에 존재하는 id에 맞는 비밀번호와 일치하지 않을 때
                    JOptionPane.showMessageDialog(null, "비밀번호가 일치하지 않습니다.");
                }
            }
        } else {
            // 존재하지 않는 아이디일 때
            JOptionPane.showMessageDialog(null, "존재하지 않는 id입니다.");
        }
    }
});
```

2. 회원가입

The screenshot shows a member registration form titled "회원가입". It includes fields for Name, ID, Password, and Phone Number, and a dropdown Combobox for area code. Buttons for "가입하기" (Join) and "취소" (Cancel) are at the bottom.

가입 정보 입력 (Input Fields)

지역번호 Combobox (Area Code Combobox)

가입하기 / 취소 버튼 (Join/Cancel Buttons)

아이디 유효성검사 (ID Validity Check)

입력칸이 빈칸일시 (When Input Field is Empty)

The slide illustrates the validation process for the ID field. It shows three states: 1. The user has entered the ID "홍길동". A message box appears stating "이미 존재하는 아이디입니다!" (The ID already exists!). 2. The user has cleared the ID field. A message box appears stating "입력이 유효하지 않습니다!" (The input is invalid!).

2. 회원가입

텍스트 파일 생성

```
private static void saveMemberToFile(Member member) {
    FileWriter writer = null;
    BufferedWriter bw = null;

    try {
        writer = new FileWriter("members.txt", true);
        bw = new BufferedWriter(writer);

        bw.write(member.toString());
        bw.newLine();

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if(bw != null) {
                bw.close();
            }
            if(writer != null) {
                writer.close();
            }
        } catch (Exception e2) {
            // TODO: handle exception
        }
    }
}
```

아이디 중복 체크

```
private static boolean isIdDuplicated(String id) {
    FileReader reader = null;
    BufferedReader br = null;
    // 아이디 중복 체크
    try {
        reader = new FileReader("members.txt");
        br = new BufferedReader(reader);

        String line;
        while ((line = br.readLine()) != null) {
            if (line.contains("아이디: " + id + ",")) {
                return true;
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if(br != null) {
                br.close();
            }
            if(reader != null) {
                reader.close();
            }
        } catch (Exception e2) {
            // TODO: handle exception
        }
    }
    return false;
}
```

가입버튼 취소버튼 리스너

```
//가입하기 취소 버튼 기능
ActionListener btnAl = new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String name = nameTf.getText();
        String id = idTf.getText();
        String password = new String(passwordPf.getPassword());
        String phoneNumber = phoneCb.getSelectedItem().toString() + phoneTf.getText();

        switch (e.getActionCommand()) {

            case "가입하기":
                if (!isValidInput(name, id, password)) {
                    JOptionPane.showMessageDialog(null, "입력이 유효하지 않습니다!");
                    return;
                }
                if (isIdDuplicated(id)) {
                    JOptionPane.showMessageDialog(null, "이미 존재하는 아이디입니다!");
                    return;
                }
                String hashedPassword = hashPassword(password);
                Member member = new Member(name, id, hashedPassword, phoneNumber);
                saveMemberToFile(member);
                JOptionPane.showMessageDialog(null, "가입 완료!");
                setVisible(false);
                loginForm.setVisible(true);
                break;

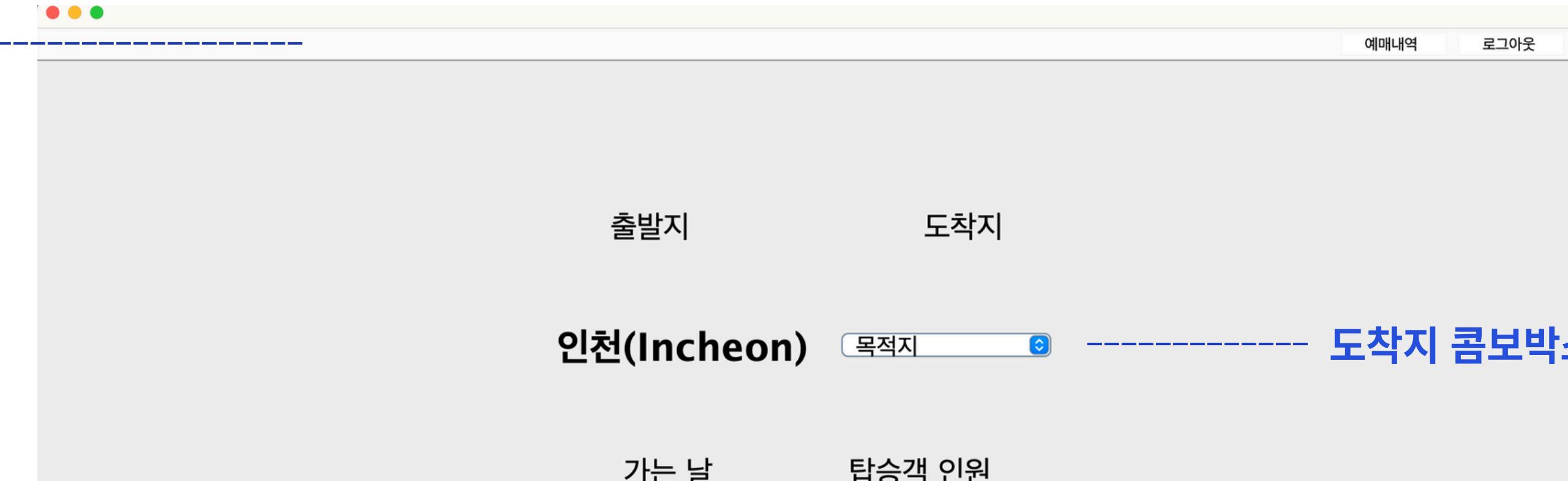
            case "취소":
                setVisible(false); // Join 화면 숨기기
                loginForm.setVisible(true); // LoginForm 화면 보이기
                break;
        }
    }
};
```

아이디, 이름 유효성 검사

```
private static boolean isValidInput(String name, String id, String password) {
    // 입력값 유효성 검사
    if (name.isEmpty() || id.isEmpty() || password.isEmpty()) {
        return false;
    }
    if (!id.matches("[a-zA-Z][a-zA-Z0-9]*")) {
        return false;
    }
    return true;
}
```

3. 항공권 조회

상단 메뉴바 -----

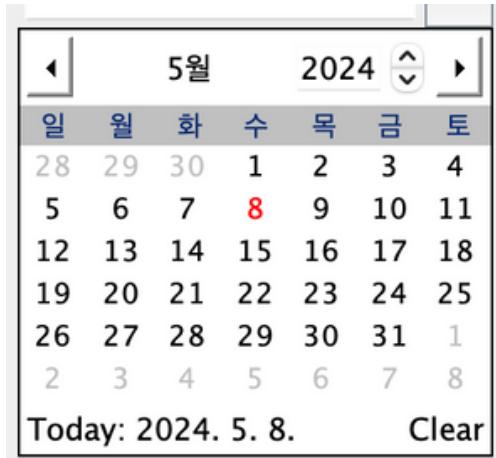


✓목적지

김포
청주
김해
대구
제주

도착지 콤보박스

출발 날짜 달력 -----



✓0
1
2
3

인원 선택 콤보박스

항공권 조회 버튼 -----

항공권 조회하기

3. 항공권 조회

```
// 출발지 고정, 목적지
JLabel start = new JLabel("인천(Incheon)");
start.setFont(new Font("맑은 고딕", Font.BOLD, 35));
start.setPreferredSize(new Dimension(250, 35));

String[] destList = {"목적지", "김포", "청주", "김해", "대구", "제주"};
JComboBox<String> dest = new JComboBox <>(destList);
dest.setPreferredSize(new Dimension(200, 38));
dest.setFont(new Font("맑은 고딕", Font.PLAIN, 22));
dest.setBackground(Color.white);

ActionListener alDest = new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        destination = (String)dest.getSelectedItem();
        System.out.println("목적지 : " + destination);
    }
};

dest.addActionListener(alDest);

// 가는 날, 인원
JLabel jl3 = new JLabel("가는 날");
jl3.setFont(new Font("맑은 고딕", Font.PLAIN, 28));
jl3.setPreferredSize(new Dimension(200, 40));

JLabel jl4 = new JLabel("탑승객 인원");
jl4.setFont(new Font("맑은 고딕", Font.PLAIN, 28));
```

```
// 날짜 선택
UtilDateModel model = new UtilDateModel();
JDatePanelImpl datePanel = new JDatePanelImpl(model);
JDatePickerImpl datePicker = new JDatePickerImpl(datePanel);
date = model.getYear() + "-" + (model.getMonth() + 1) + "-" + model.getDay();

// 인원 선택
Integer[] num = {0, 1, 2, 3};
JComboBox<Integer> people = new JComboBox <>(num);
people.setPreferredSize(new Dimension(200, 30));
people.setBackground(Color.white);
people.setSelectedIndex(0);

people.setFont(new Font("맑은 고딕", Font.PLAIN, 22));

ActionListener alPeople = new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        countPeople = (Integer)people.getSelectedItem();
        System.out.println("인원 수 : " + countPeople);
    }
};
```

```
// 조회하기 버튼 클릭 시 항공권 조회 페이지 이동
check.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // 항공권 조회 정보
        String selectedDest = (String) dest.getSelectedItem();
        int selectedPeople = people.getSelectedIndex();
        if(!selectedDest.equals("목적지") && !(selectedPeople == 0)) {
            ReservationInfo tempReser = new ReservationInfo(date, destination, countPeople);

            new FlightList(userInfo, tempReser);
            setVisible(false);
        } else {
            if (selectedDest.equals("목적지")) {
                // 유효하지 않은 목적지인 경우
                JOptionPane.showMessageDialog(null, "목적지를 선택해주세요.");
            } else if ((selectedPeople == 0))
                // 인원이 0인 경우
                JOptionPane.showMessageDialog(null, "인원을 선택해주세요.");
        }
    }
});
```

3. 항공권 조회 - 트러블 슈팅

문제. 날짜를 선택했을 시, 날짜의 정보가 바로 뜨는 달력을 만들지 못 했습니다.

접근법. 연도, 월, 일이 모두 선택이 가능한 달력을 찾기 위해 검색을 후 달력과 관련된 코드들도 여러가지 찾아 직접 만들어봐야겠다 생각했습니다.

해결법. 원하는 달력의 코드를 찾았지만 직접 만든 것이 아니라 외부 라이브러리를 추가하여 사용한다는 것을 알게 되었습니다.

JDatePicker라는 외부 라이브러리를 Build Path를 통해 추가하여 달력을 사용했습니다.

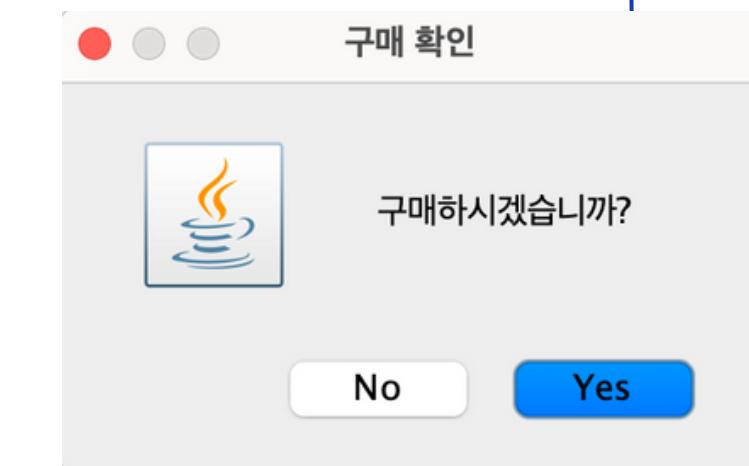
4. 항공권 리스트

조회한 항공권
리스트 테이블

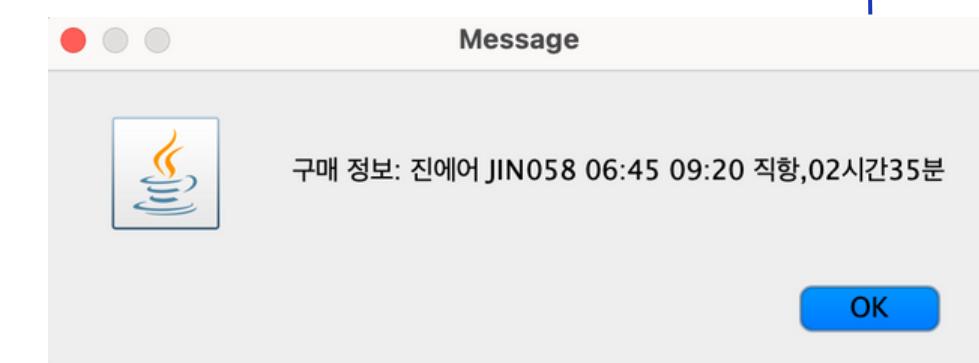
출발지: 인천 도착지 : 김해 날짜 : 2024-5-8

항공사명	항공명	출발시간	도착시간	소요시간
진에어	JIN058	06:45	09:20	직항,02시간35분
이스타항공	JIN058	07:05	09:50	직항,02시간45분
진에어	JIN871	07:25	10:00	직항,02시간35분
티웨이항공	TWA873	07:45	10:15	직항,02시간30분
에어서울	SEO037	07:50	10:30	직항,02시간40분
아시아나항공	ANA758	09:00	11:20	직항,02시간20분
진에어	JIN657	09:45	12:15	직항,02시간30분
이스타항공	EST187	15:10	17:30	직항,02시간20분
에어부산	BS848	15:50	18:10	직항,02시간20분
대한항공	KAL123	16:25	18:45	직항,02시간20분

cell 클릭시



재확인



4. 항공권 리스트

```
table.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        int row = table.rowAtPoint(e.getPoint());
        if (row >= 0) {
            int option = JOptionPane.showConfirmDialog(table, "구매하시겠습니까?", "구매 확인", JOptionPane.YES_NO_OPTION);
            if (option == JOptionPane.YES_OPTION) {
                // "예"를 클릭한 경우
                String rowData = ""; // 클릭한 행의 정보를 저장할 문자열
                for (int i = 0; i < table.getColumnCount(); i++) {
                    rowData += table.getValueAt(row, i) + " "; // 각 열의 데이터를 연결
                }
                JOptionPane.showMessageDialog(table, "구매 정보: " + rowData);

                // 항공사, 항공명 정보 ReservationInfo에 저장하기
                tempReser.setAirline(data[row][0]);
                tempReser.setAirplane(data[row][1]);

                new Seat(userInfo, tempReser);
                setVisible(false);

            } else if (option == JOptionPane.NO_OPTION) {
                // "아니오"를 클릭한 경우
                JOptionPane.showMessageDialog(table, "구매가 취소되었습니다.");
            }
        }
    }
});
```

5. 항공권 선택 - 트러블 슈팅 1

문제. 테이블에 Image파일 삽입하였을때 이미지가 저장된 주소값이 나옴

접근법. TableCellRenderer를 사용

해결법. TableCellRenderer를 사용하여 ImageIcon 객체를 생성해서 이미지 파일 경로를 설정 JLabel에 붙힌 다음 Jtable에 원하는 cell에 TableCellRenderer를 설정하였으나 똑같이 이미지 주소값이 나타남 해결하지 못함.

5. 좌석 선택

상단 메뉴바

좌석 선택

예매내역 로그아웃

Flight booking

Front

<input type="checkbox"/> A1	<input type="checkbox"/> A2	<input type="checkbox"/> A3	<input type="checkbox"/> A4	<input type="checkbox"/> A5	<input type="checkbox"/> A6	<input type="checkbox"/> A7	<input type="checkbox"/> A8	<input type="checkbox"/> A9
<input type="checkbox"/> B1	<input type="checkbox"/> B2	<input type="checkbox"/> B3	<input type="checkbox"/> B4	<input type="checkbox"/> B5	<input type="checkbox"/> B6	<input type="checkbox"/> B7	<input type="checkbox"/> B8	<input type="checkbox"/> B9
<input type="checkbox"/> C1	<input type="checkbox"/> C2	<input type="checkbox"/> C3	<input type="checkbox"/> C4	<input type="checkbox"/> C5	<input type="checkbox"/> C6	<input type="checkbox"/> C7	<input type="checkbox"/> C8	<input type="checkbox"/> C9
<input type="checkbox"/> D1	<input type="checkbox"/> D2	<input type="checkbox"/> D3	<input type="checkbox"/> D4	<input type="checkbox"/> D5	<input type="checkbox"/> D6	<input type="checkbox"/> D7	<input type="checkbox"/> D8	<input type="checkbox"/> D9
<input type="checkbox"/> E1	<input type="checkbox"/> E2	<input type="checkbox"/> E3	<input type="checkbox"/> E4	<input type="checkbox"/> E5	<input type="checkbox"/> E6	<input type="checkbox"/> E7	<input type="checkbox"/> E8	<input type="checkbox"/> E9
<input type="checkbox"/> F1	<input type="checkbox"/> F2	<input type="checkbox"/> F3	<input type="checkbox"/> F4	<input type="checkbox"/> F5	<input type="checkbox"/> F6	<input type="checkbox"/> F7	<input type="checkbox"/> F8	<input type="checkbox"/> F9
<input type="checkbox"/> G1	<input type="checkbox"/> G2	<input type="checkbox"/> G3	<input type="checkbox"/> G4	<input type="checkbox"/> G5	<input type="checkbox"/> G6	<input type="checkbox"/> G7	<input type="checkbox"/> G8	<input type="checkbox"/> G9
<input type="checkbox"/> H1	<input type="checkbox"/> H2	<input type="checkbox"/> H3	<input type="checkbox"/> H4	<input type="checkbox"/> H5	<input type="checkbox"/> H6	<input type="checkbox"/> H7	<input type="checkbox"/> H8	<input type="checkbox"/> H9
<input type="checkbox"/> I1	<input type="checkbox"/> I2	<input type="checkbox"/> I3	<input type="checkbox"/> I4	<input type="checkbox"/> I5	<input type="checkbox"/> I6	<input type="checkbox"/> I7	<input type="checkbox"/> I8	<input type="checkbox"/> I9
<input type="checkbox"/> J1	<input type="checkbox"/> J2	<input type="checkbox"/> J3	<input type="checkbox"/> J4	<input type="checkbox"/> J5	<input type="checkbox"/> J6	<input type="checkbox"/> J7	<input type="checkbox"/> J8	<input type="checkbox"/> J9

좌석 등급

등급 선택

✓ 등급 선택
이코노미
비즈니스
퍼스트

선택한 좌석정보

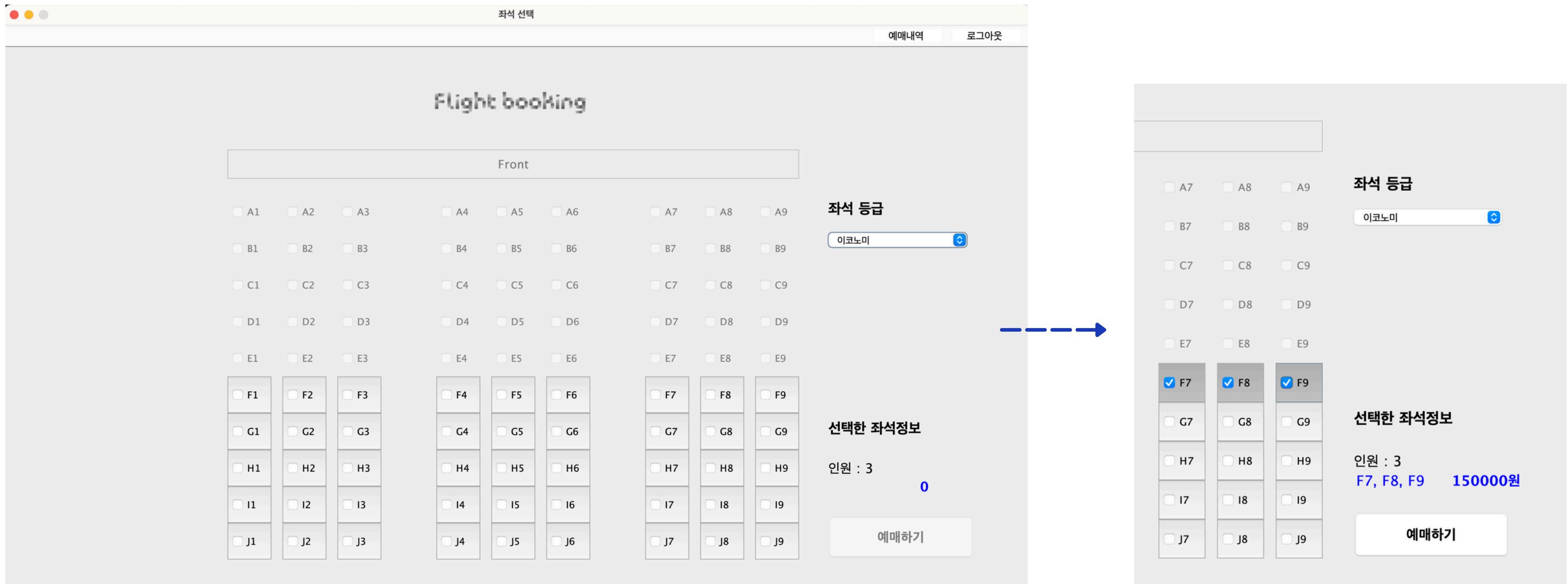
인원 : 3

선택한 좌석 정보

예매하기

예매 버튼

5. 좌석 선택



5. 좌석 선택

```

selectedSeats = new String[tempReser.getCountPeople()];
final int[] selectCount = { 0 };

for (int i = 0; i < 10; i++) {
    for (int j = 1; j <= 9; j++) {
        seatBtn[i][j].addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                JCheckBox checkBox = (JCheckBox) e.getItem();
                String seatName = checkBox.getText();
                if (e.getStateChange() == ItemEvent.SELECTED) {
                    System.out.println(seatName + "이(가) 선택되었습니다.");
                    if (selectCount[0] >= tempReser.getCountPeople()) {
                        JOptionPane.showMessageDialog(null, "인원을 초과하였습니다.", "오류",
                            JOptionPane.ERROR_MESSAGE);
                        checkBox.setSelected(false);
                        return;
                    } else {
                        selectedSeats[selectCount[0]++] = seatName;
                        int price = calculatePrice(seatName);
                        totalPrice += price;
                        seatPriceCont.setText(totalPrice + "원");
                        // 선택된 좌석 정보 업데이트
                        updateSelectedSeats(selectedSeatCont);
                    }
                } else if (e.getStateChange() == ItemEvent.DESELECTED) {
                    System.out.println(seatName + "이(가) 선택 해제되었습니다.");
                    int price = calculatePrice(seatName);
                    totalPrice -= price;
                    seatPriceCont.setText(totalPrice + "원");
                    for (int k = 0; k < selectCount[0]; k++) {
                        if (selectedSeats[k] != null && selectedSeats[k].equals(seatName)) {
                            selectedSeats[k] = null;
                            selectCount[0]--;
                        }
                    }
                    // 선택된 좌석 정보 업데이트
                    updateSelectedSeats(selectedSeatCont);
                }
                // 배열의 마지막 요소가 채워져야 예매하기 버튼 활성화
                if(selectedSeats[tempReser.getCountPeople()-1] != null) {
                    bookingBtn.setEnabled(true);
                }
            });
        });
    }
}

```

```

// 선택한 좌석 배열 출력 메서드
private void updateSelectedSeats(JLabel selectedSeatCont) {
    StringBuilder sb = new StringBuilder();
    for (String seat : selectedSeats) {
        if (seat != null) {
            sb.append(seat).append(", ");
        }
    }
    if (sb.length() > 0) {
        sb.setLength(sb.length() - 2);
    }
    selectedSeatCont.setText(sb.toString());
}

```

5. 좌석 선택 - 트러블 슈팅 1

문제. 처음에는 좌석을 단순 JButton으로 제작했다. 하지만 1개 이상의 선택한 좌석번호를 담아서 출력하는 작업을 하려고 보니, 버튼을 선택 또는 해제된 상태에 대한 이벤트를 적용할 수 없었다. 다중 선택이 되어야 하는 한계를 느꼈다.

접근법. 다중 선택의 기본이 되며 선택/해제가 되는 상태를 setSelected()로 판별할 수 있는 JCheckBox로 고쳐야 겠다고 판단했다.

해결법. JButton을 JCheckBox로 수정 후 선택이 된 상태에 대한 이벤트와 해제된 상태에 대한 이벤트를 쉽게 설정할 수 있었고, 실시간으로 선택한 좌석을 배열에 담아 출력하거나 해제한 좌석을 제거하는 작업이 잘 작동되었다.

5. 좌석 선택 - 트러블 슈팅 2

문제. 좌석이 선택/해제 가능한 체크박스라서 선택하면 가격이 증가, 해제하면 감소해야 했는데 단순 증가가 아닌 누적이 되는 등 가격 업데이트가 적용되지 않았다.

접근법. 기존에는 좌석버튼의 아이템리스너 안의 선택되었는지/해제되었는지 확인하는 조건문 안에서 선택되면 선택된 좌석의 등급별로 금액을 세팅했었는데, 여기서 가격은 미리 설정해 놓은게 아니라서 증가/감소가 아니라 누적이 되었다.
따라서 가격을 따로 세팅해놓고 원래 코드를 작성한 위치에서는 가격을 불러와서 증가/감소를 시키기로 했다.

해결법. 별도의 가격세팅 메서드를 만들어서 설정해놓은 좌석구역별 금액을 설정해놓고, 선택한 좌석의 구역을 알기위해 선택된 좌석이름을 인자로 받도록 했다.
그럼 원래 작성한 부분에서는 해당 메서드에 좌석 이름을 담아 보내고 해당 가격을 증가하거나 감소시킨 최종 가격을 정수로 받고 JLabel에 텍스트로 담아 출력했다.

6. 예매 내역 확인

The screenshot shows a flight booking confirmation page with three blue dashed boxes highlighting specific sections:

- 상단 메뉴바** (Top Menu Bar) points to the top right corner where there are three colored dots (red, yellow, green) and a "로그아웃" (Logout) button.
- 예매한 정보 확인** (Check Booking Information) points to the middle section of the page, which displays flight details in a grid format.
- 다운로드 / 메인으로
이동 버튼** (Download / Main Page Move Button) points to the bottom right corner, where there are two buttons: "다운로드" (Download) and "메인화면" (Main Page).

이름	오은지
전화번호	01012341234
항공사	항공편
 에어서울	SEO037
출발지	도착지
인천	대구
가는 날	좌석 번호
2024-5-7	C7 C8

6. 예매 내역 확인

```
// 다운로드, 메인화면 - 항공권조회
jpCheck = new JPanel();
JButton download = new JButton("다운로드");
download.setPreferredSize(new Dimension(200, 35));

JButton main = new JButton("메인화면");
main.setPreferredSize(new Dimension(200, 35));

jpCheck.add(download);
jpCheck.add(main);

// 다운로드
download.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String reservationInfo
            = "이름 : " + userInfo.getName() + "\n전화번호 : " + userInfo.getNumber() +
            "\n항공사 : " + userInfo.getAirline() + "\n항공편 : " + userInfo.getAirplane() +
            "\n날짜 : " + userInfo.getDate() + "\n출발지 : 인천" + "\n도착지 : " + userInfo.getDestination() +
            "\n좌석 : " + Arrays.toString(userInfo.getSeats());

        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setFileFilter(new FileNameExtensionFilter("Text files", "txt")); // .txt 파일만 필터링
        int userSelection = fileChooser.showSaveDialog(null);

        if (userSelection == JFileChooser.APPROVE_OPTION) {
            File fileToSave = fileChooser.getSelectedFile();
            if (!fileToSave.getAbsolutePath().endsWith(".txt")) {
                // 파일의 확장자가 .txt가 아닌 경우 확장자를 .txt로 추가
                fileToSave = new File(fileToSave.getAbsolutePath() + ".txt");
            }
            try (PrintWriter writer = new PrintWriter(new FileWriter(fileToSave))) {
                writer.println(reservationInfo); // 예매내역을 텍스트 파일에 쓰기
                System.out.println("예매내역이 저장되었습니다.");
            } catch (IOException e2) {
                e2.printStackTrace();
                System.err.println("저장 실패");
            }
        }
    }
});
jpCheck.setLayout(null);
jpCheck.setBounds(10, 10, 300, 300);
frame.add(jpCheck, "Center");
frame.setVisible(true);
}
```

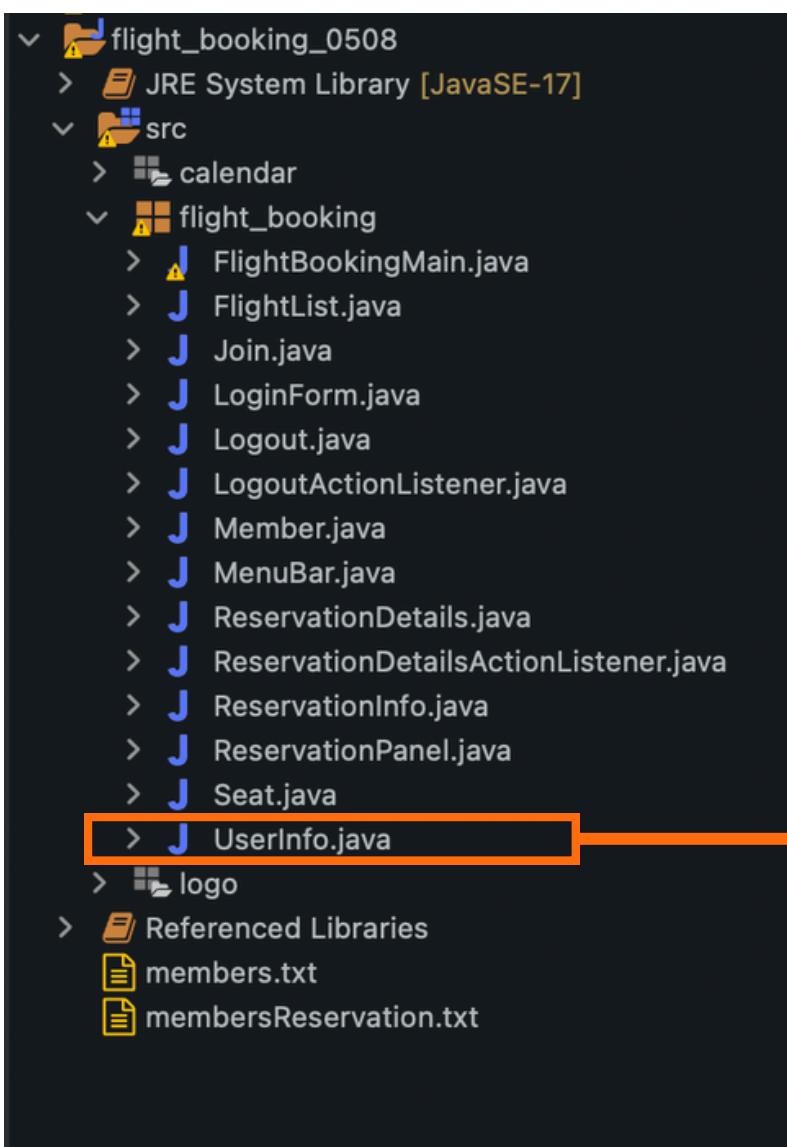
```
// 메인화면 버튼 클릭 시 메인화면으로 이동
main.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // 메인화면 - 항공권 조회 화면
        new ReservationPanel(userInfo);
        setVisible(false);
    }
});

// 항공사 이미지 경로 찾기
public String airlineImg (String airline) {
    String airlineName = "";

    if (airline != null) {
        if (airline.equals("진에어")) {
            airlineName = "jinair";
        } else if (airline.equals("이스타항공")) {
            airlineName = "easta";
        } else if (airline.equals("티웨이항공")) {
            airlineName = "tway";
        } else if (airline.equals("에어서울")) {
            airlineName = "airseoul";
        } else if (airline.equals("아시아나항공")) {
            airlineName = "ana";
        } else if (airline.equals("에어부산")) {
            airlineName = "airbusan";
        } else if (airline.equals("대한항공")) {
            airlineName = "kal";
        }
    }
    return "src/logo/" + airlineName + ".png";
}
```

6. 예매 내역 확인

클래스 리스트



유저정보 관리 클래스

```

J *UserInfo.java ×
1 package flight_booking;
2
3 import java.util.Arrays;
4
5 public class UserInfo {
6     private String id;
7     private String name;
8     private String number;
9     private String airline;
10    private String airplane;
11    private String destination;
12    private String date;
13    private String[] Seats;
14
15
16    public UserInfo () { }
17
18    public UserInfo (String id, String name, String number) {
19        this.id = id;
20        this.name = name;
21        this.number = number;
22    }
23
24    public String getName() {
25        return name;
26    }
27
28    public void setName(String name) {
29        this.name = name;
30    }
31
32
33
34
35    @Override
36    public String toString() {
37        return "아이디: " + id + ", 이름: " + name + ", 전화번호: " + number
38            + ", 항공사: " + airline + ", 항공편: " + airplane
39            + ", 도착지: " + destination + ", 날짜: " + date
40            + ", 좌석 번호: " + Arrays.toString(Seats);
41    }
42
43

```

6. 예매 내역 확인 - 트러블 슈팅

문제. 좌석 선택 시, 순서 상관없이 무작위로 선택했을 때 정렬된 상태가 아닌 선택한 순서 그대로 출력이 됐고 예매 내역이 있는 사용자가 재로그인 후 예매 내역을 바로 확인 했을 때 대괄호도 함께 출력이 됐습니다.

접근법. 예매 내역에서 텍스트로 저장되기 전에 정렬을 하여 좌석 번호들을 순차적으로 저장하게 코드를 작성해야겠다 생각했습니다.

해결법. 예매 내역에서는 sort를 이용해 정렬을 하여 저장을 했고, 로그인했을 시 좌석 정보를 가져올 때 대괄호가 포함된 요소를 substring으로 잘라내어 정보를 가져오게 코드를 작성했습니다.

프로젝트 시연

느낀점 & 배운점



김용현

처음으로 프로젝트에 참여하면서 java 공부가 부족한것을 느꼈고.
열심히는 했으나 수업시간때 배운것을 제대로 활용하지 못하게 너무 아쉽다.



노유정

객체를 사용하는 방법이나 메서드를 만들고 사용하는 방법 등 확실하게 알아갈 수 있어서 좋았으나 코드를 간결하게 만들고 싶었는데 완벽하게 정리를 하지 못하게 아쉽다.



오은지

자료구조 선언시 어떤 자료구조가 적합한지 조금 더 파악 후 결정해야 함을 느꼈다.

감사합니다

Thank you

3조 - 김용현 노유정 오은지