

Start.

- 팀 프로젝트명 – JAVA / SWING + Oracle을 활용한 도서관리 ERP 프로그램
- 개발자 : 김민국(팀장)외 2명
- 주요 역할 및 담당 : 도서검색기능 및 분류 검색, 중복 체크, 추가, 수정, 삭제 기능 등

INDEX

➤ 1. 주제 선정 및 배경

➤ 2. 계획

➤ 3. 설계

➤ 4. 화면 구현



01. 주제 선정 및 배경

1. 주제 선정 및 배경

✓ 주 제 : 관리자, 이용자를 위한 도서관리 프로그램 구현

✓ 선정 배경 :

- 학교나 회사와 같은 여러 기관에서 일하는 사서나 관리자들에게 활용되는 시스템
- 도서 공간에 소장된 도서의 명세를 기록하여 관리하기 위해 사용하는 시스템을 말한다
- 도서가 비치된 위치나 도서명을 별도로 표시하면 쉽게 찾을 수 있고 도서관 이용자들이 반납한 후 관리자들은 정확한 자리에 다시 보관하여 이후 다른 이용자들이 이용하는 데 불편함 없이 관리할 수 있다

1. 2 프로젝트의 기능 및 특징

✓ 기능 및 특징

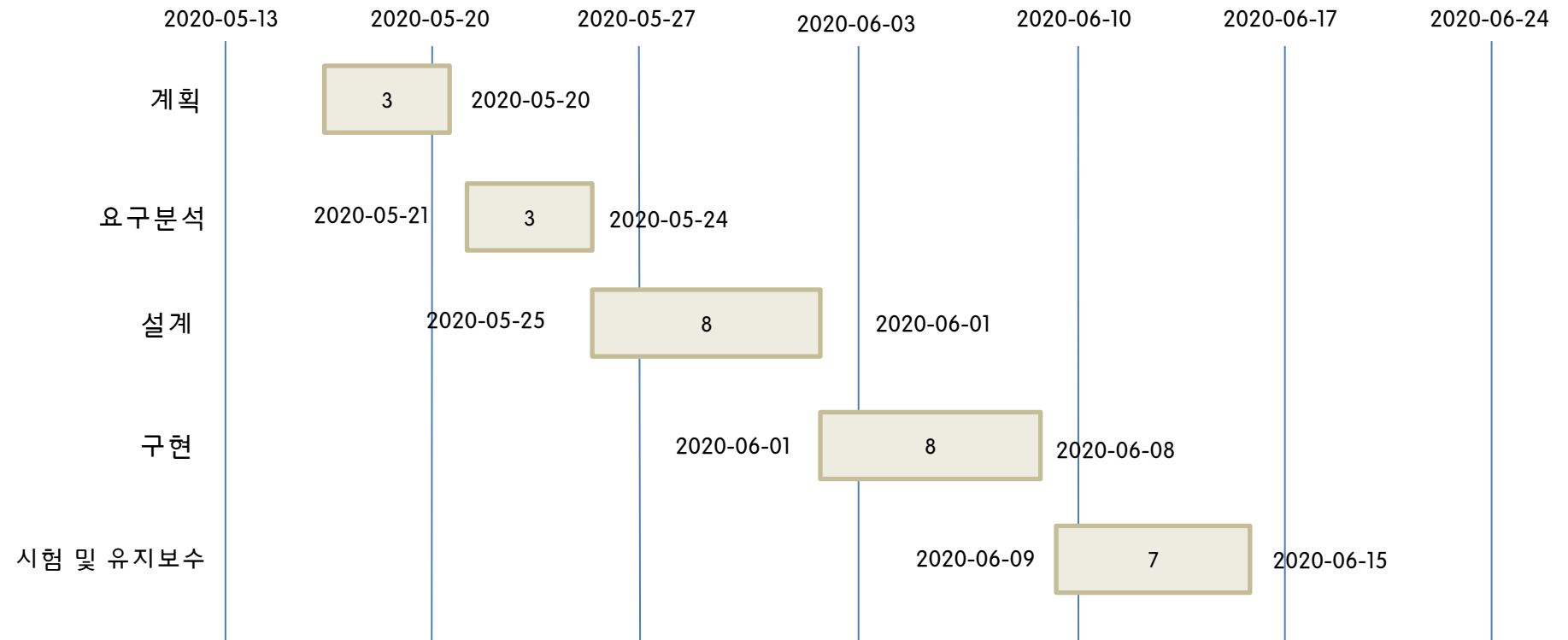
- 이용자는 별도의 회원 가입 및 로그인 없이 도서검색 가능
- 관리자는 로그인을 통하여 이용자는 접근할 수 없도록 관리
- 도서코드 등록시 중복여부 체크 기능
- 층별로 도서 재고량 파악하는 기능
- 도서명, 저자명 등 별도의 검색 기능
- 도서 데이터베이스를 엑셀 파일로 저장 기능

02. 계획

2. 1 일정 계획

총 개발 기간	2020.05.18 ~ 2020.06.15 (약 1개월)	
소 작업	도서관리 프로그램	
소 작업별 개발 기간	소작업	개발 기간
	계획	5월 18일 ~ 5월 20일
	요구 분석	5월 21일 ~ 5월 24일
	설계	5월 25일 ~ 6월 01일
	구현	6월 01일 ~ 6월 08일
	시험 및 유지보수	6월 09일 ~ 6월 15일
개발 순서	계획 → 요구 분석 → 설계 → 구현 → 시험 및 유지 보수	
필요 자원	개발용 PC, 개발 공간	

2. 1 일정 계획 (간트 차트)



03. 설계

3.1 데이터 베이스 설계

개체	ABOOKMANAGE				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	도서코드	code	varchar2	60	Not Null + Unique
속성	도서명	bookname	varchar2	400	Unique
	저자	author	varchar2	30	Not Null
	출판사	editorial	varchar2	40	Not Null
	출판일	publish_day	date	40	Not Null
	책 상태	condition	varchar2	50	Not Null
	장르	kind	varchar2	20	Not Null
	재고	remain	number	30	Not Null
	층	category	number	20	Not Null

3. 2 화면 설계

관리 화면
(SQL MODE)

관리자 메모장
- 업무 중 메모가능(구현예정)

저장

도서명으로 검색

검색

100%를 기준으로
상태 정도를 보여줄 예정

저자

코드

중복
체크

책 이름

출판사

출판일

책 상태

장르

Remain

category

채팅 메시지

1층

ConditionAvg

Stock

필드 초기화

도서코드 등록 시
중복여부 체크 기능

DB

총수 별 조회기능

도서관 내 도서 상태의
평균을 나타냄

재고량

작성 중이었던 필드 리셋

데이터 추가

수정

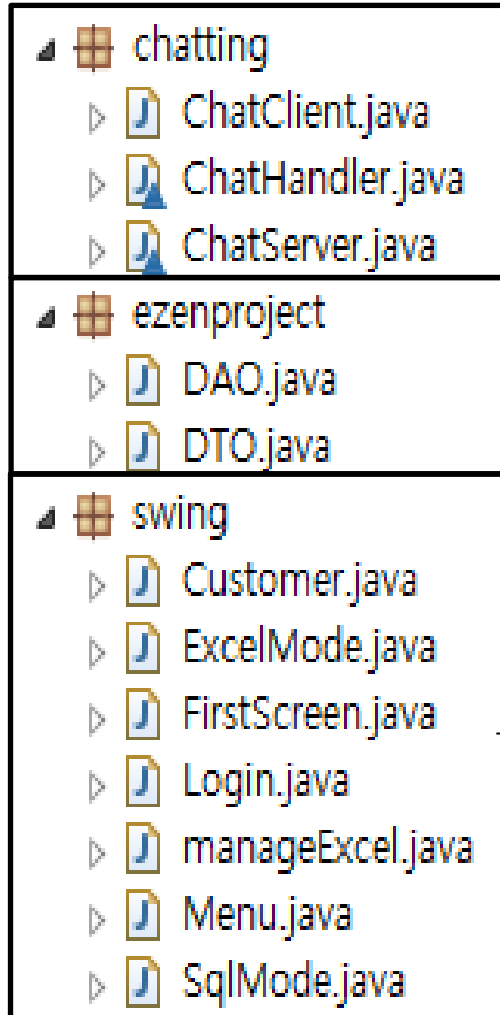
삭제

뒤로 가기

종료

04. 화면 구현

4.1 패키지 설명



채팅창의 화면 구현과 처리를 나타낸 부분

- DAO : 도서의 데이터베이스 연결 및 쿼리문을 다룬 부분
- DTO : 도서의 DTO 부분

- Customer : 프로그램 이용자가 도서 검색만 가능한 부분
- ExcelMode : 도서관리의 엑셀 모드 화면
- FirstScreen : 도서관리 프로그램 메인 화면
- Login : 관리자 로그인 화면
- manageExcel : 엑셀 파일로 보내는 기능 구현
- Menu : 도서관리 프로그램의 메뉴 화면
- SqlMode : 도서관리의 Sql모드 화면

4. 2 메인 화면



4. 3 주요 로직 (데이터베이스 연동 - Oracle)

```
public class DAO {  
  
    String driver = "oracle.jdbc.driver.OracleDriver";  
    String url = "jdbc:oracle:thin:@localhost:1521:xe";  
    String userid = "System";  
    String passwd = "비밀번호";  
  
    public DAO( ) {  
  
        try {  
            Class.forName(driver);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

4. 3 주요 로직 (도서 검색)

도서 검색

도서명으로 검색

검색

	저자	출판사	출판일	책상태	장르	재고량	위치
AAB112	마곳마사...	신용권	한빛미디어	1991/10/...	100	IT	9
AAB113	이것이C...	신용권	한빛미디어	1991/11/...	100	IT	2
AAA101	꽃을보듯...	나태주	지혜	2020/05/...	100	인문	3
AAA102	아몬드	손원평	창비	2017/03/...	100	인문	7
AAA103	페스트	알베르카...	민음사	2011/03/...	100	인문	5
AAA104	노년무위...	윤희기	소나무	2020/03/...	100	인문	4
고				2008/08/...	100	IT	2
케...				1999/03/...	100	IT	3
메...				2002/07/...	100	IT	7
상사				2005/12/...	100	역사	12
화사				2008/11/...	100	역사	2
				2006/10/...	100	역사	6
디머				1991/10/...	100	IT	9

뒤로가기

종료

```

//종료 버튼
endButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

//뒤로가기 버튼
backButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        contentPane.setVisible(false);
        FirstScreen firstScreen = new FirstScreen();
        firstScreen.setVisible(true);
    }
});
        
```

- switch문을 사용해서 조건을 처리하기 위해 배열을 사용해서 콤보 박스 만들고 검색버튼에 쿼리문을 전달하여 도서 목록에 출력 될 수 있도록 액션을 준다.
- 종료버튼은 System.exit(0)으로 실행이 멈추도록 했고, 뒤로가기는 setVisible로 현재화면을 false, 이전화면을 true

4. 3 주요 로직 (도서 검색)

```
// 검색버튼 (저자명, 출판사명, 도서명 분류 검색 기능)
searchButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        int index = searchCombo.getSelectedIndex();
        switch(index) {
            case 0 : {
                // 텍스트 필드에서 값을 가져온다 (책이름, 출판사로 검색)
                String searchText = tfsearch.getText();
                ArrayList<DTO> dtolist = new ArrayList<DTO>();
                try {
                    dtolist = dao.search(searchText);
                } catch (RecordNotFoundException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                // 쿼리문 있는 클래스 메소드에 검색
                tableModel.setNumRows(0);
                int dtoSize = dtolist.size();
            }
        }
    }
});
```

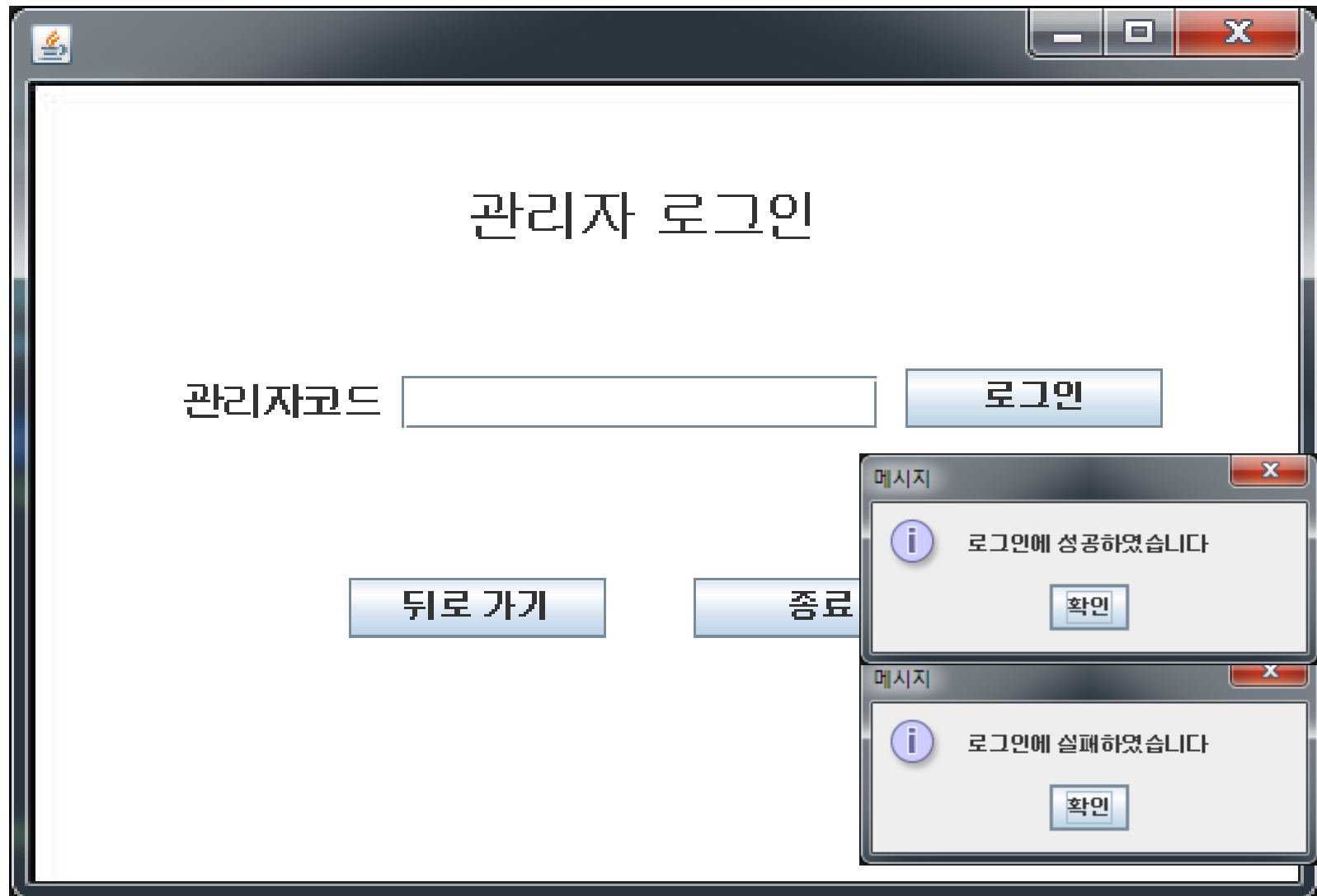
- 검색에 switch~case 문으로 조건을 받은 뒤 DAO의 search메소드에서 쿼리문을 전달하여 조건에 맞는 값을 가져와 해당 textarea에 출력

```
public ArrayList<DTO> search(String bookname) {
    Connection con = null;
    PreparedStatement pstmt = null;
    // 검색된 값을 넣는 문법
    ResultSet rs = null;
    // DTO를 담을 수 있는 어레이 리스트를 생성
    ArrayList<DTO> dtolist = new ArrayList<DTO>();
    try {
        con = DriverManager.getConnection(url, userid, passwd);
        String sql = "SELECT * FROM abookmanage WHERE bookname LIKE '%" + bookname + "%'";
        System.out.println(sql);
        System.out.println(bookname);

        pstmt = con.prepareStatement(sql);
        rs = pstmt.executeQuery();

        while (rs.next()) {
            // 검색된 결과를 담은 변수 1줄을 넣을 수 있는 DTO를 생성
            DTO dto = new DTO();
            dto.setCode(rs.getString("code"));
            dto.setBookname(rs.getString("bookname"));
            dto.setAuthor(rs.getString("author"));
            dto.setEditorial(rs.getString("editorial"));
            // System.out.println(rs.getDate("publish_day").getTime().toString());
            dto.setPublish_day(new Date(rs.getDate("publish_day").getTime()));
            dto.setCondition(rs.getString("condition"));
            dto.setKind(rs.getString("kind"));
            // 계산식을 위해 재고의 값은 인트
            dto.setRemain(rs.getInt("remain"));
            // 계산식을 위해 카테고리의 리턴 값을 숫자로 받는 것이 편하다
            dto.setCategory(rs.getInt("category"));
            // DTO를 넣을 수 있는 어레이 리스트에 추가
            dtolist.add(dto);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

4. 3 주요 로직 (관리자 로그인)



The image shows a graphical user interface for an administrator login system. The main window has a title bar with standard Windows controls (minimize, maximize, close). The title of the window is "관리자 로그인" (Administrator Login). The main content area contains the text "관리자 로그인" in a large font. Below this, there is a label "관리자코드" (Administrator Code) followed by a text input field. To the right of the input field is a button labeled "로그인" (Login). Below the input field and the "로그인" button are two more buttons: "뒤로 가기" (Go Back) and "종료" (End). Overlaid on the bottom right of the main window are two message boxes. The top message box has a title bar with the text "메시지" (Message) and a close button. It contains an information icon, the text "로그인에 성공하였습니다" (Login successful), and a button labeled "확인" (OK). The bottom message box also has a title bar with "메시지" and a close button. It contains an information icon, the text "로그인에 실패하였습니다" (Login failed), and a button labeled "확인" (OK).

관리자 로그인

관리자코드

로그인

뒤로 가기

종료

메시지

로그인에 성공하였습니다

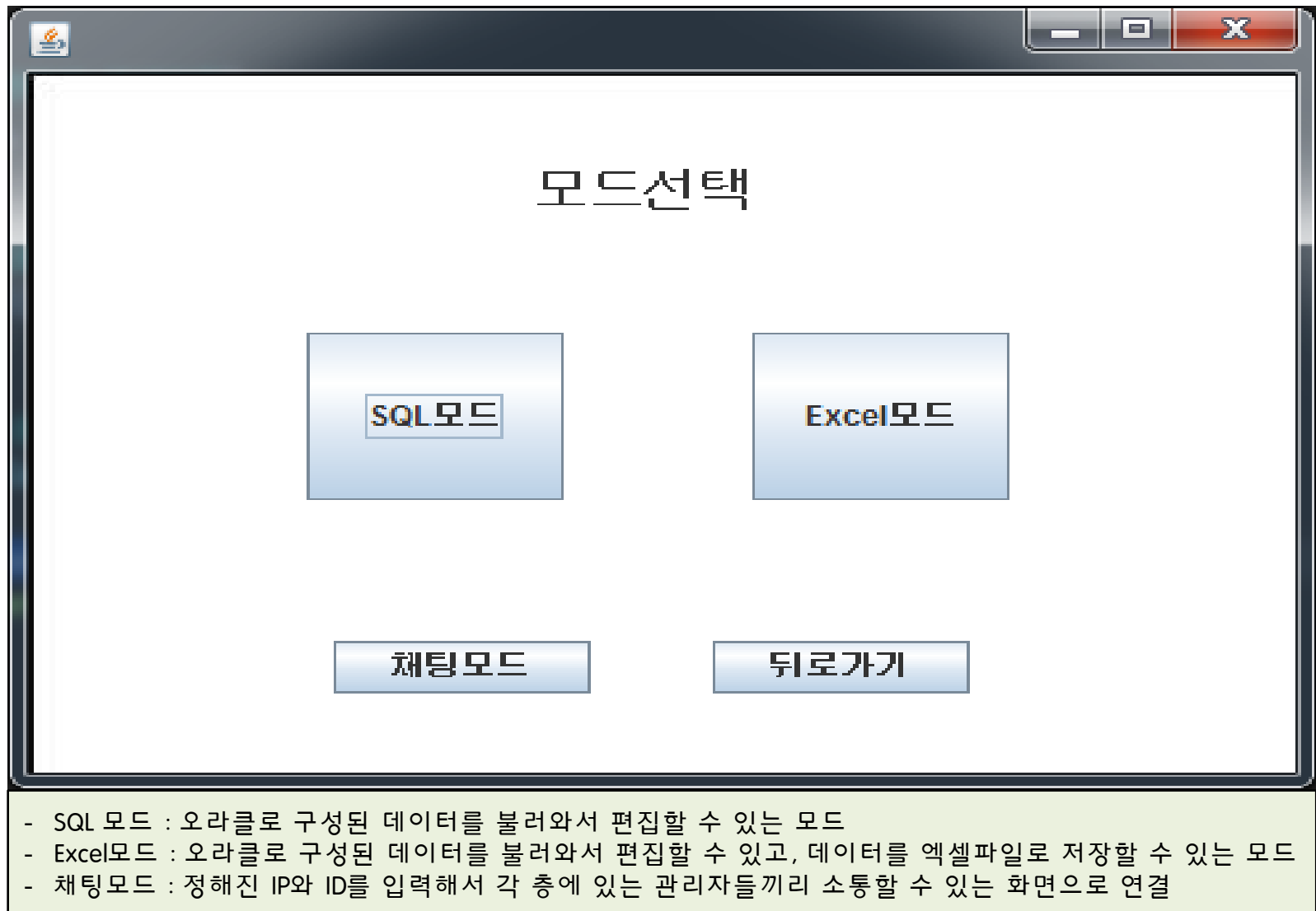
확인

메시지

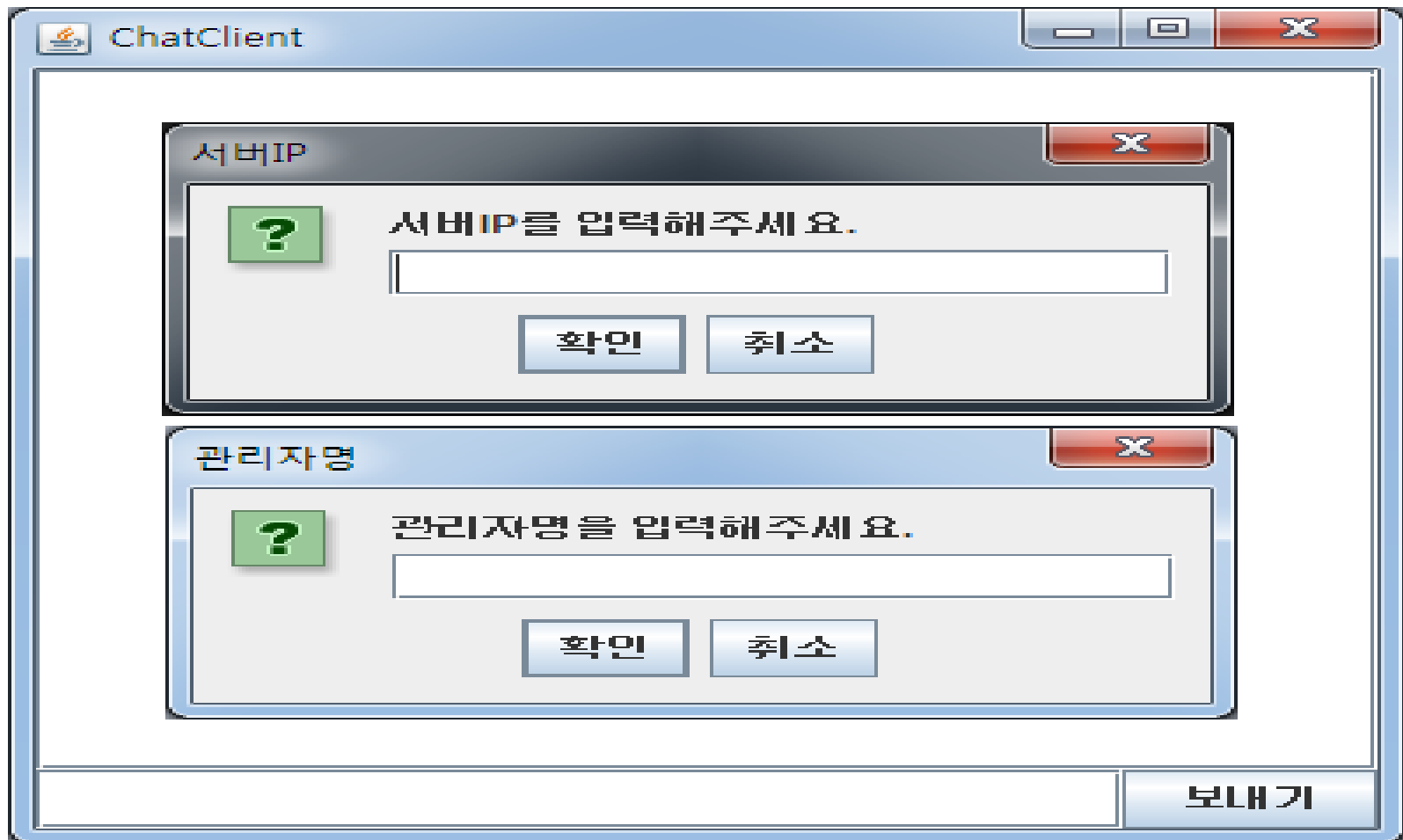
로그인에 실패하였습니다

확인

4. 3 주요 로직 (모드 선택)



4. 3 주요 로직 (채팅모드)



- 서버IP : 각 층에 있는 서버의 IP를 입력
- 관리자명 : 각 층의 관리자와 채팅시에 관리자를 구분
- 입력후 각 층의 관리자와 채팅프로그램으로 소통 가능

4. 3 주요 로직 (관리자 화면 SQL모드)

관리 화면 (SQL MODE)

도서명으로 검색

도서명으로 검색

저자명으로 검색

출판사로 검색

장르별로 검색

검색

코드

중복체크

책이름

저자

출판사

출판일

ex) xxxx/xx/xx

책상태

장르

재고

위치

코드	책이름	저자	출판사	출판일	책상태	장르	재고량	위치
AAA105	시절과 기분	김복곤	창비	2020/05/01	100	인문	12	1
AAA302	국화와 칼	루스 베네...	을유문화사	2008/11/15	100	역사	2	3
AAA303	예루살렘...	한나 아렌트	한길사	2006/10/10	100	역사	6	3
AAA112	이것이자...	신용권	한빛미디어	1991/10/23	100	IT	9	2
AAA113	이것이C#...	신용권	한빛미디어	1991/11/23	100	IT	2	2

총별 재고수량

1F

DB에 저장하기

DB에서 가져오기

필드 초기화

데이터 추가

데이터 삭제

수정

뒤로가기

종료

메시지

사용가능한 코드 입니다.

확인

메시지

중복된 코드 입니다.

확인

4. 3 주요 로직 (관리자 화면 Excel모드)

관리 화면 (Excel Mode)

도시명으로 검색

도시명으로 검색

저자명으로 검색

출판사로 검색

장르별로 검색

검색

코드

중복체크

책이름

저자

출판사

출판일

ex) xxxx/xx/xx

책상태

장르

재고

위치

코드	책이름	저자	출판사	출판일	책상태	장르	재고량	위치	총별 재고수량
AAA105	시절과 기분	김봉곤	창비	2020/05/01	100	인문	12	1	1F
AAB112	이것이자바								
AAB113	이것이C#이								
AAA101	꽃을보듯너...								
AAA102	아몬드								
AAA103	페스트								
AAA104	독나무의 파...								
AAA203	패턴 인식								
AAA204	디지털미다								
AAA205	달레마에 배...								
AAA301	홍, 군, 쇠								
AAA302	국화와 칼								
AAA303	예루살렘의 ...								
AAA112	이것이자바다	신용권	한빛미디어	1991/10/23	100	IT	9	2	
AAA113	이것이C#이...	신용권	한빛미디어	1991/11/23	100	IT	2	2	

메시지

파일이 저장되었습니다.

경로:C:\TemplezenBook.xlsx

확인

데이터 추가

데이터 삭제

수정

뒤로가기

종료

하기

저오기

Excel파일 만들기

4. 3 주요 로직 (중복 체크 및 총별 재고확인)

```
// 중복체크 버튼
overlapButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String code1 = codeLabel.getText();
        int tmp = 0;
        for(int i = 0; i < dao.searchAll().size(); i++) {
            String overlapCode = dao.searchAll().get(i).getCode();
            if(code1.equals(overlapCode)) { // AAA124 == AAA123
                JOptionPane.showMessageDialog(null, "중복된 코드 입니다.");
                codeLabel.setText(" ");
                tmp++;
                break;
            } else if(tmp == 0) {
                JOptionPane.showMessageDialog(null, "사용가능한 코드 입니다.");
                break;
            }
        }
    }
});
```

```
// 총별 재고 확인
floorCombo.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        int index = floorCombo.getSelectedIndex();
        int tmp = 0;
        switch(index) {
            case 0 : {
                for(int i = 0; i < dao.select().size(); i++) {
                    if(dao.select().get(i).getCategory() == 1) {
                        tmp += dao.select().get(i).getRemain();
                    }
                }
                floorLabel.setText(String.valueOf(tmp));
            } break;
            case 1 : {
                for(int i = 0; i < dao.select().size(); i++) {
                    // 책이름 검색 (관리자, 유저 모두사용)
                }
            }
        }
    }
});

public ArrayList<DTO> select() {
    ArrayList<DTO> list = new ArrayList<DTO>();
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    try {
        con = DriverManager.getConnection(url, userid, passwd);
        String query = "SELECT * FROM abookmanage";
        pstmt = con.prepareStatement(query);
        rs = pstmt.executeQuery();
        while (rs.next()) {
            DTO dto = new DTO();
            dto.setCode(rs.getString("code"));
            dto.setBookname(rs.getString("bookname"));
            dto.setAuthor(rs.getString("author"));
            dto.setEditorial(rs.getString("editorial"));
            dto.setPublish_day(rs.getDate("publish_day"));
            dto.setCondition(rs.getString("condition"));
            dto.setKind(rs.getString("kind"));
            // 계산식을 위해 재고의 값은 인트
            dto.setRemain(rs.getInt("remain"));
            // 계산식을 위해 카테고리의 리턴 값을 숫자로 받음
            dto.setCategory(rs.getInt("category"));
            list.add(dto);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if(rs != null) rs.close();
        if(pstmt != null) pstmt.close();
        if(con != null) con.close();
    }
    return list;
}
```

총별 재고수량	
2F	42
1F	
2F	
3F	저장하기

- 중복 체크 기능

- 총별 재고 확인 기능
책 이름을 검색해서 그 책에 재고값을 불러와 임시변수에 누적후 출력

4. 3 주요 로직 (도서 추가)

```
@Override
public void actionPerformed(ActionEvent arg0) {

    //get text로 모든 값을 가져온다
    String codetext = codeLabel.getText();
    String chulpan = chulpanLabel.getText();
    String genre = genreLabel.getText();
    String bookname = booknameLabel.getText();

    String date = dateLabel.getText();
    SimpleDateFormat transFormat = new SimpleDateFormat("yyyy/MM/dd");
    Date to = new Date();
    try {
        to = transFormat.parse(date);
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    String Remain = RemainLabel.getText();
    int remainint = Integer.parseInt(Remain);
    String author = authorLabel.getText();
    String condition = conditionLabel.getText();
    String category = categoryLabel.getText();
    int categoryint = Integer.parseInt(category);
    //가져온 값으로 DTO생성
    DTO dto = new DTO();
    dto.setAuthor(author);
    dto.setBookname(bookname);
    dto.setCategory(categoryint);
    dto.setCode(codetext);
    dto.setCondition(condition);
    dto.setEditorial(chulpan);
    dto.setKind(genre);
    dto.setPublish_day(to);
    dto.setRemain(remainint);
    //DAO에서 insert 한다
    tableModel.setNumRows(0);
    ArrayList<DTO> dtolistaftercreate = new ArrayList<DTO>();
    //데이터 베이스 안에 있는 값을 전체 검색
    dtolistaftercreate = dao.searchAll();
```

```
// 책정보 저장(관리자)
public int insert(DTO dto) {
    Connection con = null;
    PreparedStatement pstmt = null;
    int n = 0;
    try {
        con = DriverManager.getConnection(url, userid, passwd);
    } catch (SQLException e2) {
        // TODO Auto-generated catch block
        e2.printStackTrace();
    }
    String sql = "INSERT INTO abookmanage(code,bookname,author,editorial,publish
        +
        " VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)";
    try {
        pstmt = con.prepareStatement(sql);
        System.out.println(dto.getCode());
        pstmt.setString(1, dto.getCode());
        pstmt.setString(2, dto.getBookname());
        pstmt.setString(3, dto.getAuthor());
        pstmt.setString(4, dto.getEditorial());
        pstmt.setDate(5, convertToSQL(dto.getPublish_day()));
        pstmt.setString(6, dto.getCondition());
        pstmt.setString(7, dto.getKind());
        // 계산식을 위해 재고의 값은 인트
        pstmt.setInt(8, dto.getRemain());
        // 계산식을 위해 카테고리의 리턴 값을 숫자로 받는 것이 편하다
        pstmt.setInt(9, dto.getCategory());
        n = pstmt.executeUpdate();
        con.commit();
```

- 데이터 추가

각 텍스트필드에 값을 dto로 저장한뒤 textarea에 저장한 값을 출력

4. 3 주요 로직 (도서 삭제 및 수정)

// 책정보 삭제

```
public int delete(String bookcode) {
    Connection con = null;
    PreparedStatement pstmt = null;
    int n = 0;
    try {
        con = DriverManager.getConnection(url, userid, passwd);
        String sql = "DELETE FROM abookmanage WHERE code = ?";
        pstmt = con.prepareStatement(sql);
        pstmt.setString(1, bookcode);
        n = pstmt.executeUpdate();
        con.commit();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
```

- 도서 삭제

// 책정보 수정

```
public int update(DTO dto) {
    Connection con = null;
    PreparedStatement pstmt = null;
    int n = 0;
    try {
        con = DriverManager.getConnection(url, userid, passwd);
        String sql = "UPDATE abookmanage SET bookname=?, author=?, editorial=?, publish_day=?, condition=?, " +
            " kind=?, remain=?, category=? WHERE code = ?";
        pstmt = con.prepareStatement(sql);
        pstmt.setString(1, dto.getBookname());
        pstmt.setString(2, dto.getAuthor());
        pstmt.setString(3, dto.getEditorial());
        pstmt.setDate(4, convertToSQL(dto.getPublish_day()));
        pstmt.setString(5, dto.getCondition());
        pstmt.setString(6, dto.getKind());
        // 계산식을 위해 재고의 값은 인트
        pstmt.setInt(7, dto.getRemain());
        // 계산식을 위해 카테고리 및 리턴 값을 숫자로 받는 것이 편하다
        pstmt.setInt(8, dto.getCategory());
        pstmt.setString(9, dto.getCode());
        n = pstmt.executeUpdate();
        con.commit();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
```

- 도서 수정

Thank you.