Ⅳ. 영화 목록 조회

관리자 메뉴에서 '영화 목록 조회' 기능을 구현

- 이를 위해 AdminMenu 기능을 추가하고
- Movie 클래스를 생성한다

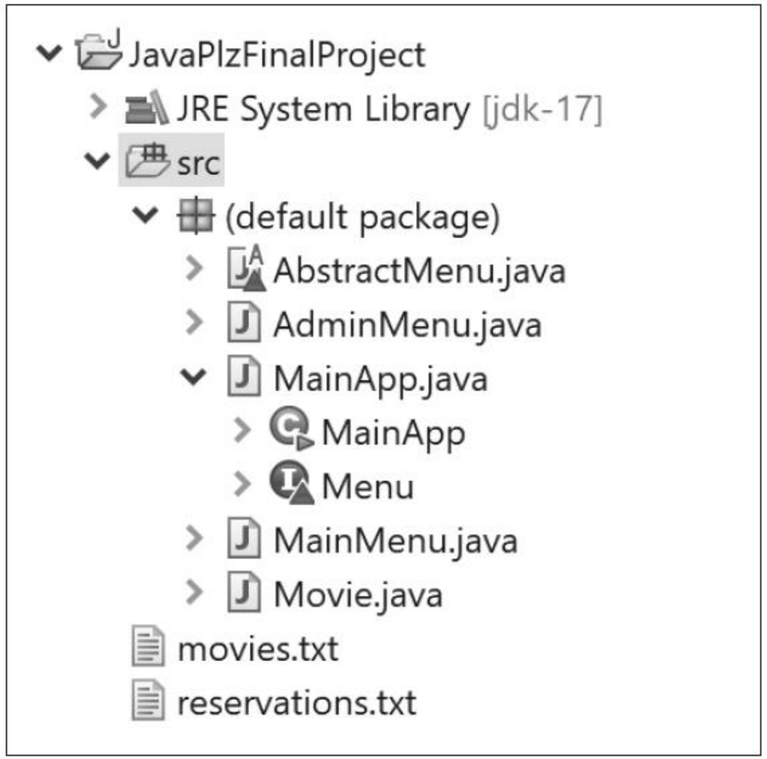


그림 13-10 영화 목록 조회 구현 후 프로젝트 파일 구조

```
01 import java.io.IOException;
02 import java.util.ArrayList;
03
04 public class AdminMenu extends AbstractMenu {
...    ...
21     public Menu next() {
22         switch (scanner.nextLine()) {
23             case "2":                // 2번 메뉴 선택 시
24                 printAllMovies();    // 영화 목록 출력
25                 return this;         // 관리자 메뉴 객체 반환
26             case "b": return prevMenu;
27             default: return this;
28         }
29     }
30
31     private void printAllMovies() {
32         try {
33             ArrayList<Movie> movies = Movie.findAll(); // 모든 영화를 가져옴
34             System.out.println();
35             for (int i = 0; i < movies.size(); i++) {
36                 System.out.printf("%s\n", movies.get(i).toString()); // 출력
37             }
38         } catch (IOException e) {
39             System.out.println("데이터 접근에 실패하였습니다."); // 예외 처리
40         }
41     }
42 }
```

새로 추가된 코드

새로 추가된 코드

새로 추가된 코드

AdminMenu.java 기능 추가

영화 목록 조회 기능을 추가. 영화 데이터는 Movie 클래스로부터 전달 받음.

- **01~02행:** 자바 API를 불러옴. 예외 처리와 데이터 관리를 위해 IOException과 ArrayList를 가져옴
- **23~25행:** next( ) 메소드에 2번 선택지를 추가하고, 다시 자기 자신의 메뉴로의 이동을 구현. 24행의 호출로 모든 영화가 출력
- **31~41행:** printAllMovies( ) 메소드. 모든 영화 정보를 출력. 33행의 Movie.findAll( ) 메소드를 통해 영화 정보를 ArrayList로 받아, 이를 반복문으로 출력. 예외는 try-catch 문에 의해 처리

```
01 import java.io.*;
02 import java.util.ArrayList;                    ---- 자바 API 불러오기
03
04 public class Movie {
05     private long id;         // 영화 대푯값
06     private String title;    // 영화 제목
07     private String genre;    // 영화 장르                    ---- 필드
08     private static final File file = new File("movies.txt");
                                // movies.txt 파일 객체
09
10     public Movie(long id, String title, String genre) {    // 생성자
11         this.id = id;
12         this.title = title;
13         this.genre = genre;
14     }
                입출력 예외 발생 시, 호출 위치로 전달 ----
15
16     public static ArrayList<Movie> findAll() throws IOException {
17         ArrayList<Movie> movies = new ArrayList<Movie>();
18         BufferedReader br = new BufferedReader(new FileReader(file));
19         String line = null;
20
21         while ((line = br.readLine()) != null) { // 파일을 한 행씩 읽어와 반복
22             String[] temp = line.split(",");  // 쉼표를 기준으로 문자열을 나눔
23             Movie m = new Movie(              // 영화 객체 생성
24                     Long.parseLong(temp[0]),  // 영화 대푯값
25                     temp[1],                  // 영화 제목
26                     temp[2]                   // 영화 장르
27             );
28             movies.add(m);  // 생성 영화 객체를 ArrayList에 추가
29         }
30         br.close();         // 파일 입력 흐름 해제
31         return movies;      // 영화 객체가 담긴 ArrayList 반환
32     }
                                ---- 모든 영화 정보를 ArrayList<Movie>에 담아 반환
33
34     public String toString() {
35         return String.format("[%d]: %s(%s)", id, title, genre);
36     }
37 }                           ---- 객체 정보를 문자열로 반환
```

Movie.java 생성

영화(Movie) 클래스를 작성. 이는 영화 데이터의 객체화와 파일 입출력을 담당. 해당 클래스는 점진적으로 개선될 예정.

- **01~02행:** 구현에 필요한 자바 API를 불러옴. 파일 입출력과 ArrayList를 사용하기 위함
- **05~08행:** 영화 객체를 위한 필드를 선언. 영화 대푯값, 영화 제목, 장르는 인스턴스변수로, movies.txt 파일 객체는 클래스변수로 선언.
- **16~32행:** findAll( ) 메소드. 파일로부터 모든 영화 정보를 반환. BufferedReader를 사용해 행 단위 파일을 입력받음. 쉼표를 기준으로 나눠 영화 객체를 생성한 뒤, ArrayList에 담아 반환
- **34~36행:** toString( ) 메소드. 영화 객체 정보를 문자열로 반환

Ⅳ. 영화 목록 조회

중간 점검: '영화 목록 조회' 기능 시연

① 관리자 메뉴 출력	1: 영화 등록하기 2: 영화 목록 보기 3: 영화 삭제하기 b: 메인 메뉴로 이동
② 이동할 메뉴 선택	메뉴를 선택하세요: 2
③ 영화 목록 출력	[1627175707]: 어벤쟈스(판타지) [1627175946]: 컨져링(호러) [1627175983]: 러부 액츄얄리(로맨스)

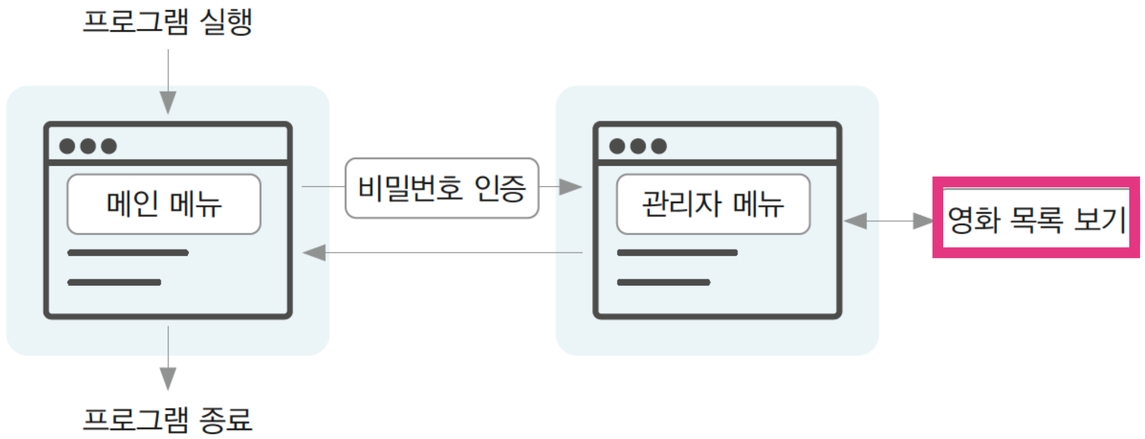


그림 13-11 현재까지 구현된 기능

V. 영화 등록

관리자 메뉴에서 '영화 등록' 기능을 구현

- 이를 위해 AdminMenu의 기능을 추가하고
- Movie의 기능도 추가한다

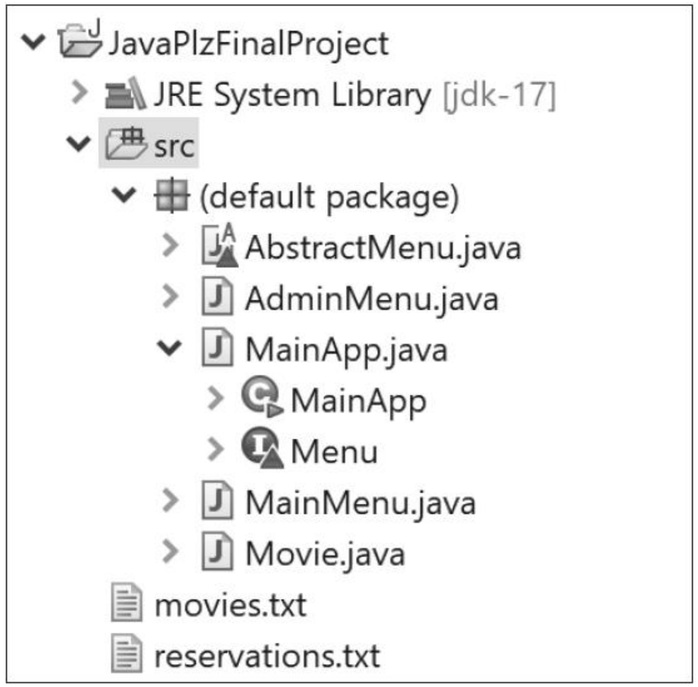


그림 13-12 영화 등록 구현 후 프로젝트 파일 구조

```
01 import java.io.IOException;
02 import java.util.ArrayList;
03
04 public class AdminMenu extends AbstractMenu {
...     ...
21     public Menu next() {
22         switch (scanner.nextLine()) {                    ------ 새로 추가된 코드
23             case "1":
24                 createMovie();      // 영화 등록 진행
25                 return this;        // 관리자 메뉴 객체 반환
26             case "2":
27                 printAllMovies();
28                 return this;
29             case "b": return prevMenu;
30             default: return this;
31         }
32     }
33
34     private void printAllMovies() { ... }
...     ....
45                                                          ------ 새로 추가된 코드
46     private void createMovie() {
47         System.out.print("제목: ");
48         String title = scanner.nextLine();       // 제목 입력
49         System.out.print("장르: ");
50         String genre = scanner.nextLine();       // 장르 입력
51         Movie movie = new Movie(title, genre);   // 영화 객체 생성
52         try {          ------ 영화 객체를 저장
53             movie.save();                        // 영화 객체를 저장
54             System.out.println(">> 등록되었습니다.");
55         } catch (IOException e) {                // 예외 처리
56             System.out.println(">> 실패하였습니다.");
57         }
58     }
59 }
```

AdminMenu.java 기능 추가

영화 객체 저장 기능을 추가. 사용자로부터 입력받은 영화 정보를 객체화하고, Movie 클래스를 통해 파일로 저장.

- **23~25행:** next( ) 메소드에 1번 선택지를 추가하고, 다시 자기 자신의 메뉴로 이동하도록 구현. 24행의 메소드 호출로 영화 등록이 수행
- **46~58행:** createMovie( ) 메소드. 영화 등록을 수행. 영화 제목과 장르를 입력받아 영화 객체를 만든 후, 53행에서 save( ) 메소드를 호출해 파일을 저장

```
01 import java.io.*;
02 import java.time.Instant;          ------ 새로 추가된 코드
03 import java.util.ArrayList;
04
05 public class Movie {
...     ...
39     public Movie(String title, String genre) {          ------ 새로 추가된 코드
40         this.id = Instant.now().getEpochSecond();       // 타임스탬프
41         this.title = title;
42         this.genre = genre;
43     }
44
45     public void save() throws IOException {
46         FileWriter fw = new FileWriter(file, true);
                                   // 이어쓰기(append) 모드 설정(true)
47         fw.write(this.toFileString() + "\n");
48         fw.close();
49     }
                                                  ------ 새로 추가된 코드
50
51     private String toFileString() {   // 객체 정보를 문자열로 변환
52         return String.format("%d,%s,%s", id, title, genre);
53     }
54 }                                                 ------ 새로 추가된 코드
```

Movie.java 기능 추가

영화 객체의 파일 저장 기능을 추가. 영화 정보는 movies.txt 파일에 기록.

- **02행:** 영화의 대푯값 생성을 위해 java.time 패키지의 Instant 클 래스를 불러옴
- **39~43행:** 새로 추가된 생성자. 제목과 장르를 입력받아 객체를 생성. 영화의 대푯값은 40행의 코드 Instant.now( ).getEpochSecond( )에 의해 초 단위 타임스탬프로 설정
- **45~49행:** save( ) 메소드. 영화 객체를 파일에 출력. FileWriter의 이어쓰기 모드를 true로 설정했으므로, 파일 출력은 맨 마지막 줄 끝에 작성
- **51~53행:** toFileString( ) 메소드. 영화 객체의 정보를 파일 출력 형식으로 변환. 이는 데이터 설계에 준한 것으로, 영화 대푯값과 제목, 장르를 쉼표로 구분하는 문자열

V. 영화 등록

중간 점검: '영화 등록' 기능 시연

① movies.txt 초깃값	01 1627175707,어벤쟈스,판타지 02 1627175946,컨져링,호러 03 1627175983,러부 액츄얄리,로맨스 04 ---- 개행 문자 '\n'에 의해 빈 줄 추가
② 관리자 메뉴 출력	1: 영화 등록하기 2: 영화 목록 보기 3: 영화 삭제하기 b: 메인 메뉴로 이동
③ 이동할 메뉴 선택	메뉴를 선택하세요: 1
④ 등록할 영화 정보 입력	제목: 굿 윌 헌터 장르: 성장 》 등록되었습니다.
⑤ movies.txt 결과	01 1627175707,어벤쟈스,판타지 02 1627175946,컨져링,호러 03 1627175983,러부 액츄얄리,로맨스 04 1627327520,굿 윌 헌터,성장 ---- 새로 추가된 영화 정보 05 ---- 개행 문자 '\n'에 의해 빈 줄 추가

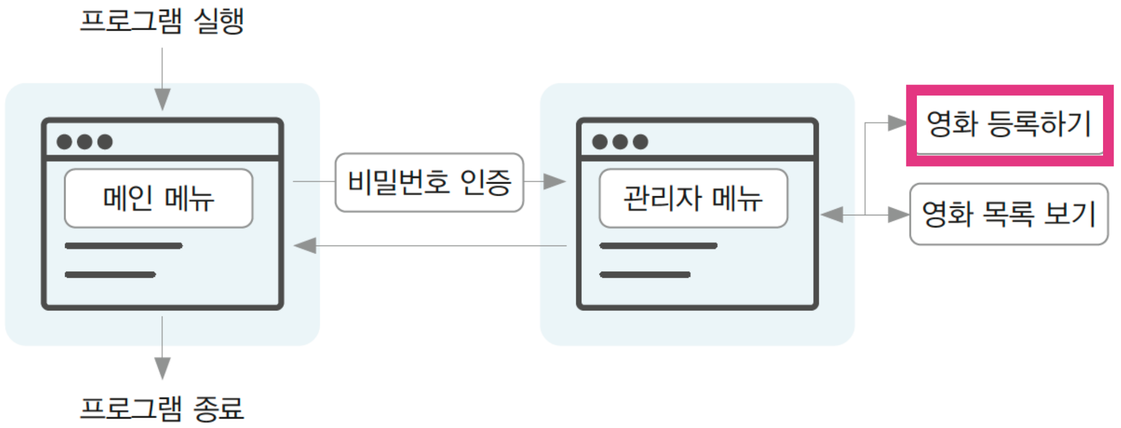
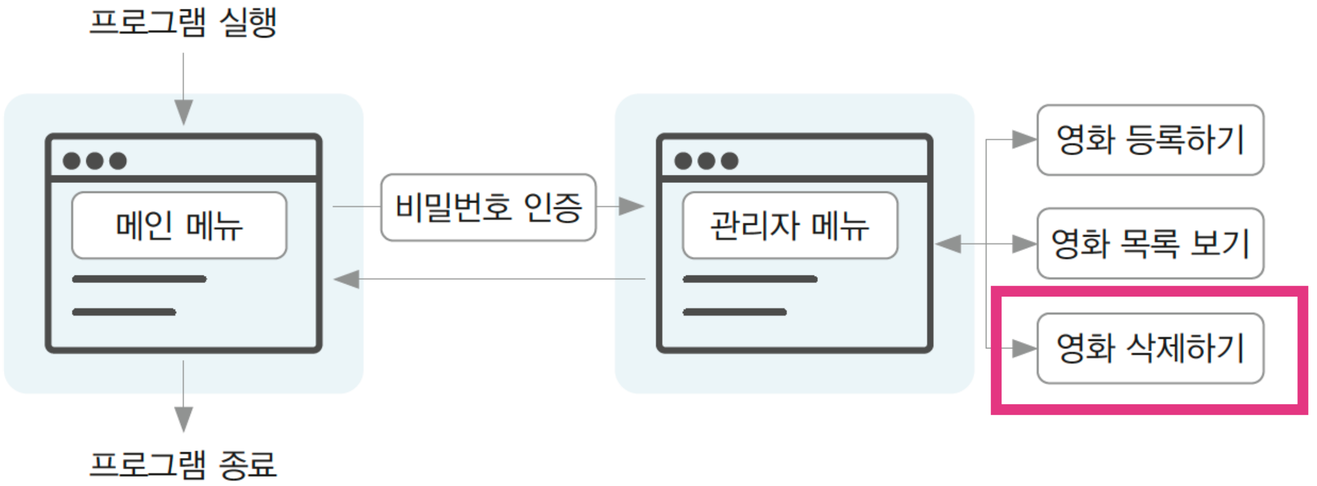


그림 13-13 현재까지 구현된 기능

VI. 영화 삭제

관리자 메뉴에서 '영화 삭제' 기능을 구현

- 이를 위해 AdminMenu의 기능을 추가하고
- Movie의 기능도 추가한다



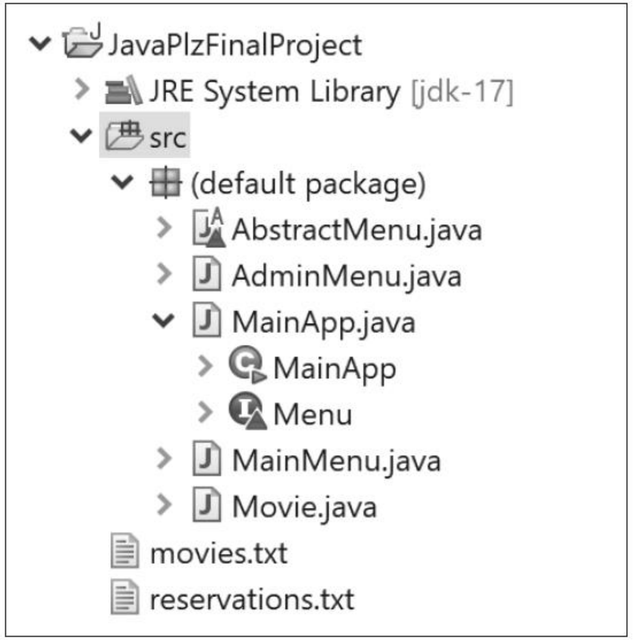


그림 13-14 영화 삭제 구현 후 프로젝트 파일 구조

```
01 import java.io.IOException;
02 import java.util.ArrayList;
03
04 public class AdminMenu extends AbstractMenu {
...     ...
21     public Menu next() {
22         switch (scanner.nextLine()) {
                                        새로 추가된 코드
...             ...
29             case "3":
30                 deleteMovie();    // 영화 삭제 진행
31                 return this;      // 관리자 메뉴 객체 반환(다시 관리자 메뉴가 나옴)
32             case "b": return prevMenu;
33             default: return this;
34         }
35     }
36
37     private void printAllMovies() { ... }
48
49     private void createMovie() { ... }
                                        새로 추가된 코드
62
63     private void deleteMovie() {
64         printAllMovies();        // 모든 영화를 출력
65         System.out.print("삭제할 영화를 선택하세요: ");
66         try {
67             Movie.delete(scanner.nextLine());
                                    // 사용자 입력값 기준으로 삭제 요청
68             System.out.println(">> 삭제되었습니다.");
69         } catch (IOException e) {
70             System.out.println(">> 삭제에 실패하였습니다.");
71         }
72     }
73 }
```

AdminMenu.java 기능 추가

영화 삭제 기능을 추가. 삭제를 위한 대푯값은 사용자로부터 입력받음

.

- **29~31행:** next( ) 메소드에 3번 선택지를 추가하고, 다시 관리자 메뉴로 돌아옴. 30행의 메소드 호출로 영화 삭제가 진행
- **63~72행:** deleteMovie( ) 메소드. 사용자는 먼저 출력된 영화 목록 중 삭제 대상의 대푯값을 입력. 입력된 값은 67행의 Movie.delete( ) 메소드로 전달되어 삭제 대상을 찾는데 사용. 삭제 중 발생한 예외는 try-catch 문에 의해 처리

```
01 import java.io.*;
02 import java.time.Instant;
03 import java.util.ArrayList;
04
05 public class Movie {
...     ...                                    새로 추가된 코드
55     public static void delete(String movieIdStr) throws IOException {
56         BufferedReader br = new BufferedReader(new FileReader(file));
57         String text = "";                          // 파일 복사를 위한 빈 문자열
58         String line = null;
59
60         while ((line = br.readLine()) != null) {
                                          // 파일을 행 단위로 읽어옴(반복)
61             String[] temp = line.split(",");  // 쉼표 기준으로 문자열을 나눔
62             if (movieIdStr.equals(temp[0])) { // 삭제 대상값을 찾으면
63                 continue;                  // 다음 반복으로 넘어감(복사되지 않게)
64             }
65             text += line + "\n";                   // 읽은 문자열을 누적하여 복사
66         }
67         br.close();                                // 입력 흐름 해제
68
69         FileWriter fw = new FileWriter(file);
                                  // FileWriter 객체 생성(덮어쓰기 모드)
70         fw.write(text);          // 파일 출력
71         fw.close();              // 출력 흐름 해제
72     }
73 }         복사 문자열 파일 출력
```

Movie.java 기능 추가

영화 정보 삭제 기능을 추가. 전달된 대푯값을 movies.txt 파일에서 찾아 해당 행을 삭제

- **56~58행:** 파일을 읽어오기 위한 변수들. BufferedReader 객체로 행 단위 입력을 준비
- **60~67행:** movies.txt 파일의 특정 행을 삭제. 기존 파일을 행 단위로 복사하는 코드가 있지만 삭제 대상이 발견된 경우엔 62~64행에 의해 복사되지 않도록 함. 그 결과 특정 행을 제외한 모든 값이 문자열 text에 복사
- **69~71행:** 특정 행을 제외한 문자열 text를 movies.txt 파일에 덮어씌움

VI. 영화 삭제

중간 점검: '영화 삭제' 기능 시연

① movies.txt 초깃값	01 1627175707,어벤져스,판타지 02 1627175946,컨져링,호러 03 1627175983,러부 액츄얄리,로맨스 04 1627327520,굿 윌 헌터,성장 05 ---- 개행 문자 '\n'에 의한 빈 줄 추가
② 관리자 메뉴 출력	1: 영화 등록하기 2: 영화 목록 보기 3: 영화 삭제하기 b: 메인 메뉴로 이동
③ 이동할 메뉴 선택	메뉴를 선택하세요: 3
④ 영화 목록 출력	[1627175707]: 어벤져스(판타지) [1627175946]: 컨져링(호러) [1627175983]: 러부 액츄얄리(로맨스) [1627327520]: 굿 윌 헌터(성장)
⑤ 삭제 대상 입력	삭제할 영화를 선택하세요: 1627327520
⑥ movies.txt 결과	01 1627175707,어벤져스,판타지 02 1627175946,컨져링,호러 03 1627175983,러부 액츄얄리,로맨스 04 ---- 개행 문자 '\n'에 의한 빈 줄 추가

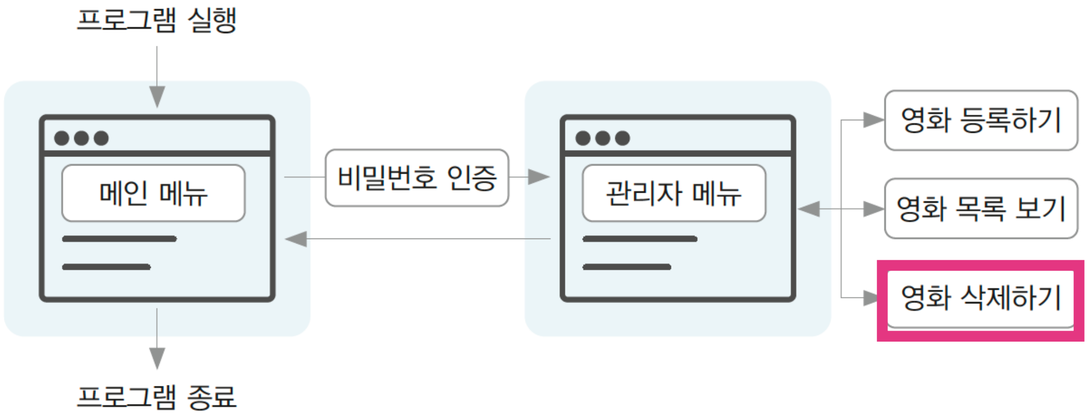


그림 13-15 현재까지 구현된 기능