

세탁기 실시간 사용 현황 제공 웹사이트 개발

조규철*, 박상용^o

*인하공업전문대학 컴퓨터정보과,

^o인하공업전문대학 컴퓨터정보과

e-mail: kccho@inhac.ac.kr*, 202044052@itc.ac.kr^o

Developing a Website to Serve Laundry Real-time Usage

Cho Kyu Cheol*, Park Sang Yong^o

*Dept. of Computer Science, Inha Technical College,

^oDept. of Computer Science, Inha Technical College

● 요약 ●

공동 주거공간에서 생활하는 사람들은 세탁기를 공동으로 사용해야 하는데, 규칙이 엄격하게 정해져 있지 않으면 본인이 원하는 시간에 사용하기 힘들다. 본 논문에서는 이러한 문제점을 해소하고자 개발한 세탁기 예약 및 실시간 이용 현황 조회 웹사이트의 연구 내용을 기록하였다. 개발된 사이트를 통해 사용자는 예약을 하여 원하는 시간에 해당 시설을 이용할 수 있다. 또한 실시간으로 사용 현황을 조회함으로써 본인의 사용 순서를 명확하게 인지할 수 있으며 결과적으로 시간을 절약할 수 있다.

키워드: 실시간 사용 현황(Real-time usage), 예약 시스템(Reservation system)

I. 서론

현대 사회는 학교, 혹은 직장 때문에 타지에서 생활하는 사람이 많다. 그중에는 금전적으로 여유가 돼서 모든 시설이 구비된 개인 공간에서 거주하는 사람도 있겠지만, 대다수는 기숙사, 셰어하우스와 같은 공용공간에서 여러 사람들과 함께 거주할 것이다. 이런 공용공간에서 살면 불편한 점은 세탁기와 같은 편의시설을 함께 사용해야 한다는 것인데, 그 수가 한정되어 있고 사용 순서 또한 정해져 있지 않은 경우가 많아서 원하는 시간에 사용하는 것은 현실적으로 어렵다. 이러한 문제점을 해소하고자 사용자가 원하는 시간에 세탁기를 사용할 수 있도록 예약과 실시간 사용 현황을 제공하는 시스템을 기획, 개발하였다.

II. 시스템 개발

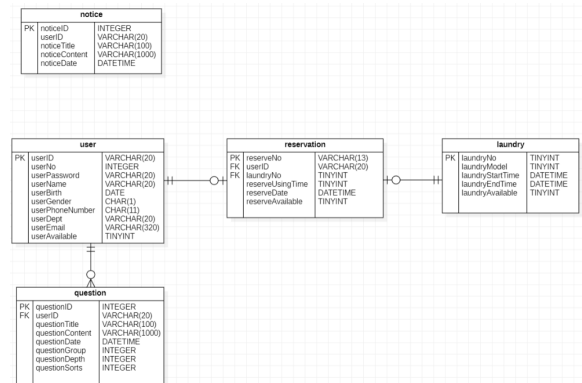


Fig. 1. ERD of the system

그림 1은 본 시스템의 ER 다이어그램이다. 메인 기능인 세탁기 예약 기능과 실시간 사용 현황을 제공하기 위해 사용되는 테이블은 회원 정보를 저장하는 user, 예약 정보를 저장하는 reservation, 세탁기 상태를 저장하는 laundry이다. reservation 테이블이 user, laundry 테이블을 참조하고 있으며 세 테이블 조인을 통해 실시간 사용 현황 기능이 구현된다.

Table 1. Function specification

기능 명세서			
구분	서비스(메뉴)	기능명	기능 설명
공통	회원가입	회원가입	[아이디],[비밀번호],[이름],[성별],[전화번호][소속][이메일] 총 7개의 정보를 입력
	로그인	로그인	[아이디],[비밀번호]를 입력하여 로그인
	검색	검색	원하는 글의 내용을 검색
사용자	예약	사용 예약	예약 시, 시간 옵션 2개(50분, 1시간 20분) 선택, 예약번호 생성 / 예약이 되어 있을 경우 중복 예약 불가능
		예약 내역	예약한 정보를 확인할 수 있음
		예약 취소	예약을 취소함(예약 정보가 삭제됨)
	실시간 현황	세탁기 사용현황	세탁기의 현재 사용 현황을 볼 수 있음(사용 중, 수거대기, 사용가능)
		대기번호	현재 대기 인원과 자신의 대기번호를 확인할 수 있음
Q&A	문의글 작성	문의 글 작성	
	문의글 수정	문의 글 수정	
	문의글 삭제	문의 글 삭제	
	문의글 삭제	문의 글 삭제	
관리자	관리	회원가입 관리	사용자가 회원가입 시 입력한 정보를 확인하여 회원가입 승인 및 거절
		공지글 작성	공지 글 작성
		공지글 수정	공지 글 수정
		공지글 삭제	공지 글 삭제
	Q&A	문의글 답변 작성	문의 글 답변 작성
		문의글 삭제	문의 글 삭제

또한, 공지사항을 작성하는 게시판에 사용되는 notice 테이블, 이용에 불편함 혹은 시설 고장 등으로 인한 문의를 할 수 있는 계층형 게시판을 구현하기 위해 사용되는 question 테이블도 필요하다.

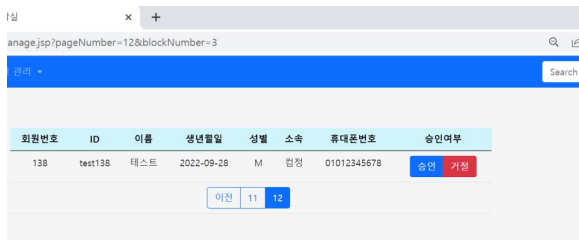


Fig. 2. Member registration approval page

사용자가 표 1의 회원가입 부분에 제시된 정보를 입력하여 회원가입을 하면 승인 대기 상태가 된다. 관리자는 그림 2의 회원관리 페이지에서 가입 신청한 회원의 정보를 확인하여 사용자의 가입 승인 또는 거절을 수행할 수 있다.



Fig. 3. Reservation main screen

승인된 사용자는 그림 3의 예약 페이지에서 예약과 예약 내역을 조회할 수 있다.

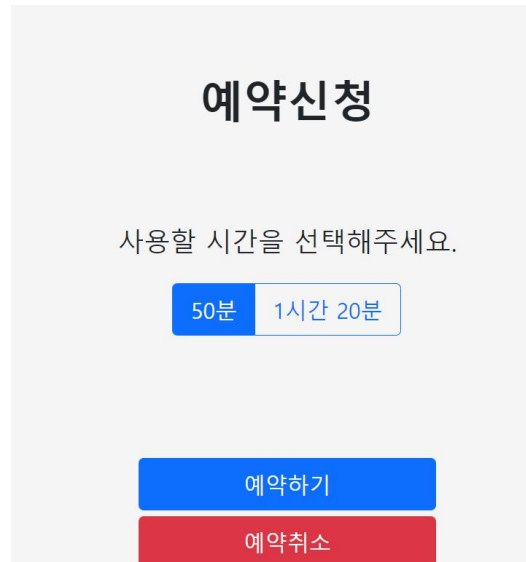


Fig. 4. Reservation screen

예약하기를 클릭할 경우, 그림 4의 예약 페이지로 이동하며, 예약을 할 수 있는데 이때 예약 가능한 시간은 [50분, 1시간 20분] 총 두 가지이다.

예약번호	예약날짜	사용시간	사용여부
2022102600007	2022-10-26 09:22:53	80분	미사용 취소
2022102500004	2022-10-25 17:14:55	50분	사용완료
2022102500003	2022-10-25 17:14:39	50분	사용완료

Fig. 5. Reservation details screen

그림 5의 예약 내역 페이지에서는 사용 전 상태의 예약 내용을 조회하여 예약 취소를 수행할 수 있다.

세탁기번호	세탁기상태	남은시간	예약번호	사용자명
세탁기[50분]				
1	수거대기	0	2022111500015	테스트
2	수거대기	0	2022111500016	실시간테스트1
3	수거대기	1	2022111500017	실시간테스트2
현재 대기하고 있는 총 인원 수[50분]				
2명				
내 예약번호			내 앞의 대기자 수	
2022111500019			1명	

Fig. 6. Real-time usage status screen

사용자가 예약을 진행한 뒤, 메인 페이지에서 그림 6과 같이 실시간 사용 현황을 조회할 수 있다. 세탁기의 상태는 [사용 가능/사용 중/수거 대기] 총 세 가지 상태가 있으며, 무인 시설에서 사용되는 시스템이므로 사용자가 예약할 때 예약 시간은 실제 사용 시간에 10분을 더한 시간이 부여된다. 다음 사용자가 세탁기를 사용하기 전까지 본인의 세탁물을 수거할 수 있는 여유시간이 필요하기 때문이다.

```
//사용 전이면서, 50분짜리에 예약한 예약 정보를 받아온다.
ReservationDAO sRsvDAO = new ReservationDAO();
LaundryDAO sLaundryDAO = new LaundryDAO();
int sLaundryCount = sLaundryDAO.LdryCount(1); //오늘 1 == 50분짜리 세탁기
Queue<Reservation> noUsingShortList = sRsvDAO.NoUsingList(50, sLaundryCount);

//실시간으로 데이터 불러주는 객체
RealTimeUsgDAO sRealTimeUsgDAO = new RealTimeUsgDAO();
HashMap<Integer, String> sLdryList = sRealTimeUsgDAO.TimeList(1, 50); // sLdry
```

Fig. 7. Queue containing reservation details

세탁기를 사용하기 전인 예약들은 모두 그림 7의 큐[1]에 삽입되며 아래에 세탁기의 번호, 상태를 포함하는 해시 맵[2]을 순회하여 사용 가능한 세탁기가 있으면 세탁기 테이블에 저장된 해당 세탁기에 큐의 가장 처음 삽입된 예약이 입력되는데 입력되는 순간에 해당 시간이 세탁기의 시작 시간에 '입력 시간 + 사용 시간 + 10분'의 시간이 세탁기의 종료 시간에 입력되고 세탁기의 상태는 사용 중으로 전환된다.

```
public HashMap<Integer, String> TimeList(int model, int UsingTime){
    HashMap<Integer, String> hm = new HashMap<>();
    try {
        getCon();
        String sql = "SELECT * FROM laundry JOIN reservation "
            + "ON laundry_laundryNo = reservation_laundryNo "
            + "JOIN user ON reservation_userID = user_userID "
            + "WHERE laundryModel = ? AND reservelUsingTime = ? AND reserveAvailable = ? AND laundryAvailable IN (?, ?)"
            + "ORDER BY reservelNo";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, model);
        pstmt.setInt(2, UsingTime);
        pstmt.setInt(3, 1);
        pstmt.setInt(4, 2);
        pstmt.setInt(5, 3);
        rs = pstmt.executeQuery();
        while(rs.next()){
            hm.put(rs.getInt("laundryNo"), rs.getInt("laundryAvailable") + " "
                + rs.getString("reservelNo") + " "
                + rs.getString("userName"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return hm;
}
```

Fig. 8. Method for real-time usage status

실시간 사용 현황 데이터를 화면에 출력하기 위해선 user, reservation, laundry 세 개의 테이블에 있는 정보들이 모두 필요하므로 그림 8과 같이 조인 쿼리문을 작성한 후 필요한 정보를 해시맵에 담아 화면에 출력할 수 있도록 하였다.

```
<%
    ArrayList<String> sOrderList = sRsvDAO.OrderList(50);
    String mySOrder = null;
%>
```

Fig. 9. list to indicate waiting numbers

```
function autoRefresh_div()
{
    var currentLocation = window.location;
    $("#RealTime").load(currentLocation + ' #RealTime');
}
setInterval('autoRefresh_div()', 1000);
```

Fig. 10. function that refreshes once every second

세탁기가 모두 사용 중일 경우 사용자의 예약은 그림 9의 대기 리스트[3]에 삽입되며, 그림 6의 실시간 사용현황은 그림 10의 비동기 통신[4]을 통해 1초마다 한 번씩 새로고침이 실행된다. 그리하여 사용자는 실시간으로 본인의 예약 상태 및 세탁기의 상태 체크가 가능하다.

글번호	제목	작성일	작성자
3	****삭제된 게시물입니다.****	2022-11-23 12:06:38	*****
2	테스트	2022-11-23 12:05:50	admin
12	RE : 테스트	2022-11-27 11:12:28	admin
1	****삭제된 게시물입니다.****	2022-11-23 01:53:21	*****

Fig. 11. Hierarchical board for other inquiries

예약과 실시간 사용 현황 조회와는 별개로 사용에 불편함이 있거나 궁금한 점이 있는 경우 자유롭게 문의를 할 수 있는 계층형 게시판 또한 존재한다. 해당 게시판의 게시글은 작성자와 관리자만 조회할 수 있으며, 답글 기능은 관리자만 사용할 수 있다.

III. 결론

본 시스템을 만들게 된 이유는 사용자의 소중한 시간을 지키기 위해서이다. 완성된 시스템에서 예약 기능을 통해 사용자는 본인의 사용 순서를 명확하게 인지할 수 있으며, 실시간으로 사용 현황을 조회함으로써 원하는 시간에 해당 시설을 이용할 수 있다. 그리하여 결과적으로 사용자는 시간을 절약할 수 있다.

REFERENCES

[1] Programiz, "Queue Data Structure", <https://www.programiz.com/dsa/queue>

- [2] W3 schools, "Java HashMap", https://www.w3schools.com/java/java_hashmap.asp
- [3] W3 schools, "Java ArrayList", https://www.w3schools.com/java/java_arraylist.asp
- [4] W3 schools, "jQuery - AJAX load() Method", https://www.w3schools.com/jquery/jquery_ajax_load.asp