

4조 Spring 프로젝트



목차

1. 개발배경 및 개요



2. 개발 구성



3. 화면 구현 및 기능 구현



후기 및 Q&A





stage.1

개발 배경 및 개요





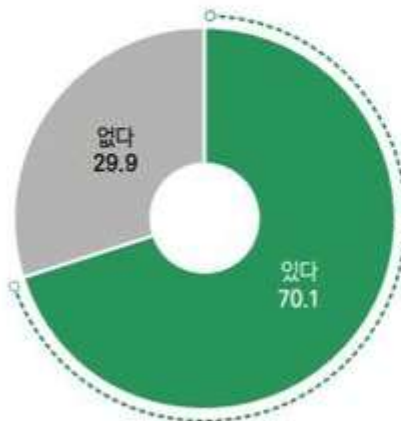
그림 2-2-1-4 PC 게임 소프트웨어 구입 경험 및 구입 형태

문) 귀하께서는 PC 게임 소프트웨어를 패키지(CD/DVD 등) 혹은 스팀(STEAM) 등의 온라인 사이트에서 구입하여 이용해 보신 적 있습니까?

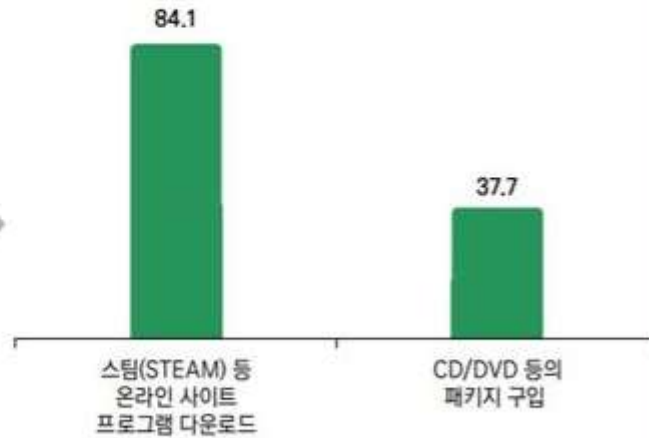
문) 그렇다면, 귀하께서 PC 게임을 구입한 형태는 다음 중 어떤 것이었습니까?

(단위: %, n=1,284)

(단위: %, n=구입 경험 '있다' 응답자 901, 중복 응답)



구입 경험



구입 형태

패키지 게임의 구매가 증가하는데 한국에서는 그 게임들을 관리하는 게임플랫폼이 없어서 제작해보았습니다.





| | | |
|---------|-----------------------------------|---|
| 프로젝트 | GAME(게임 플랫폼 사이트) | |
| 개발기간 | 2022. 03. 28 ~ 2022. 04. 08 (14일) | |
| 개발인원 | 3명 | |
| 참여자 | 정재규, 구주용, 김경래 | |
| 프로젝트 요약 | 게임플랫폼 제작 및 관리 | |
| 개발환경 | 운영체제 | Window 10 64Bit |
| | 개발언어 | JAVA, HTML, CSS, JavaScript, JQuery, JSON |
| | 개발도구 | SpringToolSuite4, MySQL, Maven |
| | 데이터베이스 | MySQL |
| | 프레임워크 | SpringBoot, JPA |



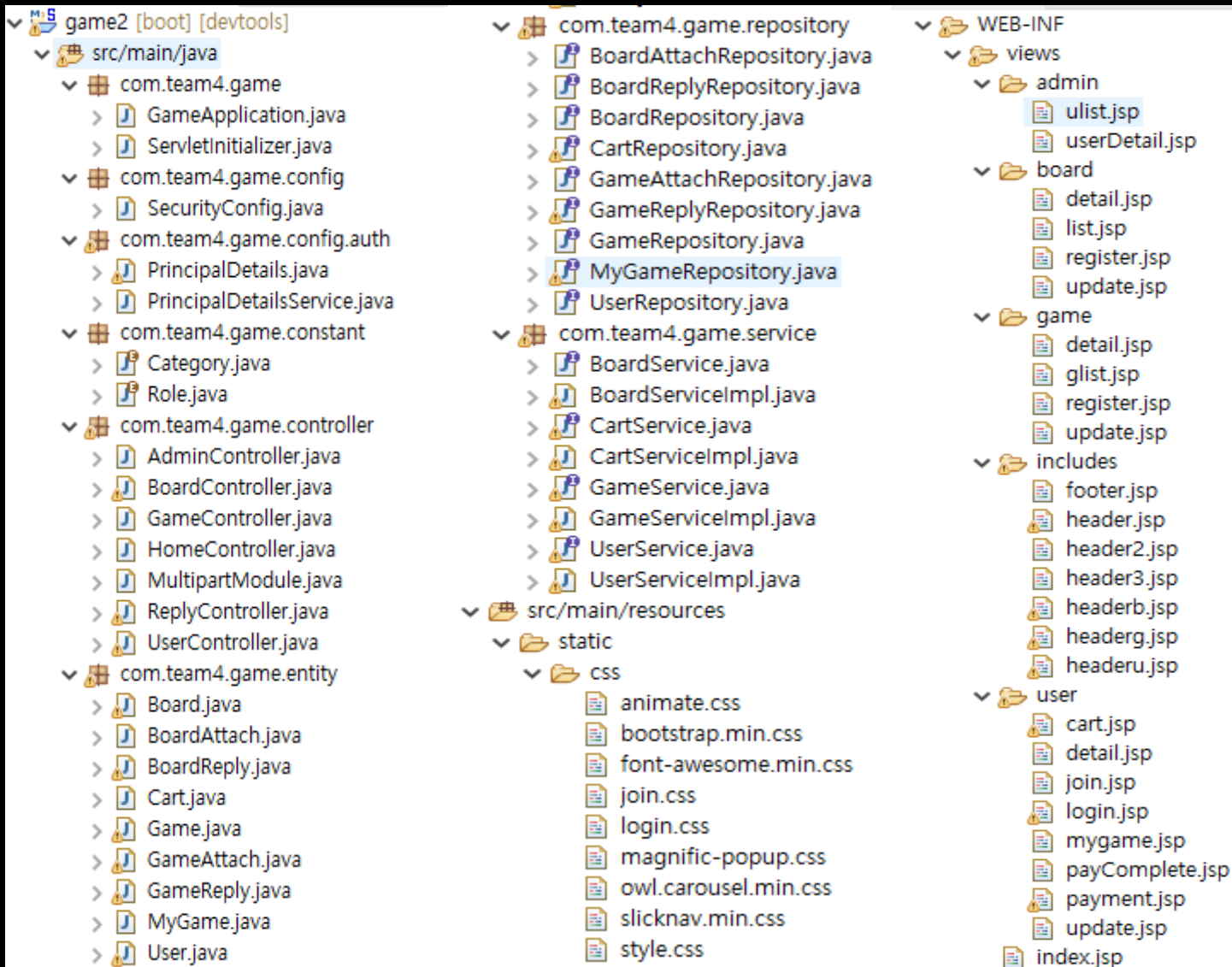
stage.2

개발 구성





| | |
|-----|--|
| 비회원 | 회원가입 게시판(보기, 검색) 게임(보기, 검색) |
| 회원 | 로그인 게시판(보기, 검색, 댓글, 글쓰기, 수정, 삭제) 게임(보기, 검색, 평점리뷰, 장바구니, 구매, 내 게임[구매] 목록) 회원정보(보기, 수정, 탈퇴) |
| 관리자 | 회원관리(목록, 보기, 삭제) 게시글삭제 게임관리(등록, 수정, 삭제) |



프로젝트를 구현하기 위해 사용한 패키지명과 폴더배치 구조입니다.
java, jsp, css, js, pom.xml 파일입니다.



```

34< dependency>
35    <groupId>org.springframework.boot</groupId>
36    <artifactId>spring-boot-devtools</artifactId>
37    <scope>runtime</scope>
38    <optional>true</optional>
39</dependency>
40< dependency>
41    <groupId>mysql</groupId>
42    <artifactId>mysql-connector-java</artifactId>
43    <scope>runtime</scope>
44</dependency>
45< dependency>
46    <groupId>org.projectlombok</groupId>
47    <artifactId>lombok</artifactId>
48    <optional>true</optional>
49</dependency>
50< dependency>
51    <groupId>org.springframework.boot</groupId>
52    <artifactId>spring-boot-starter-tomcat</artifactId>
53    <scope>provided</scope>
54</dependency>
55< dependency>
56    <groupId>org.springframework.boot</groupId>
57    <artifactId>spring-boot-starter-test</artifactId>
58    <scope>test</scope>
59</dependency>
60< dependency>
61    <groupId>org.springframework.security</groupId>
62    <artifactId>spring-security-test</artifactId>
63    <scope>test</scope>
64</dependency>
65    <!-- https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper -->
66< dependency>
67    <groupId>org.apache.tomcat.embed</groupId>
68    <artifactId>tomcat-embed-jasper</artifactId>
69</dependency>
70    <!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
71< dependency>
72    <groupId>javax.servlet</groupId>
73    <artifactId>jstl</artifactId>
74</dependency>
75    <!-- https://mvnrepository.com/artifact/org.springframework.security/spring-security-taglibs -->
76< dependency>
77    <groupId>org.springframework.security</groupId>
78    <artifactId>spring-security-taglibs</artifactId>
79</dependency>
80
81    <!-- querydsl -->
82< dependency>
83    <groupId>com.querydsl</groupId>
84    <artifactId>querydsl-jpa</artifactId>
85</dependency>
86
87< dependency>
88    <groupId>com.querydsl</groupId>
89    <artifactId>querydsl-apt</artifactId>
90</dependency>
91

```

New SPring Start Project

작업환경을 설정해주었으며 , Lombok,MySQL Driver Spring Security 등 사용.



```
1 server.port=8081
2 spring.mvc.view.prefix=/WEB-INF/views/
3 spring.mvc.view.suffix=.jsp
4
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6 spring.datasource.url=jdbc:mysql://localhost:3306/gamedb?useSSL=false&serverTimezone=Asia/Seoul&&characterEncoding=UTF-8
7 spring.datasource.username=koo
8 spring.datasource.password=1234
9
10 #LAZY loading error
11 spring.jackson.serialization.fail-on-empty-beans=false
12
13 spring.jpa.properties.hibernate.show_sql=true
14
15 #spring.jpa.properties.hibernate.format_sql=true
16 #logging.level.org.hibernate.type.descriptor.sql=trace
17
18 spring.jpa.hibernate.ddl-auto=update
19 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
```

1. MySQL DB와 연결해주는 작업
2. hibernate 사용 - 자바 언어를 위한 ORM 프레임워크이며, JPA의 구현체 JPA 인터페이스를 구현하며, 내부적으로 JDBC API를 사용합니다.

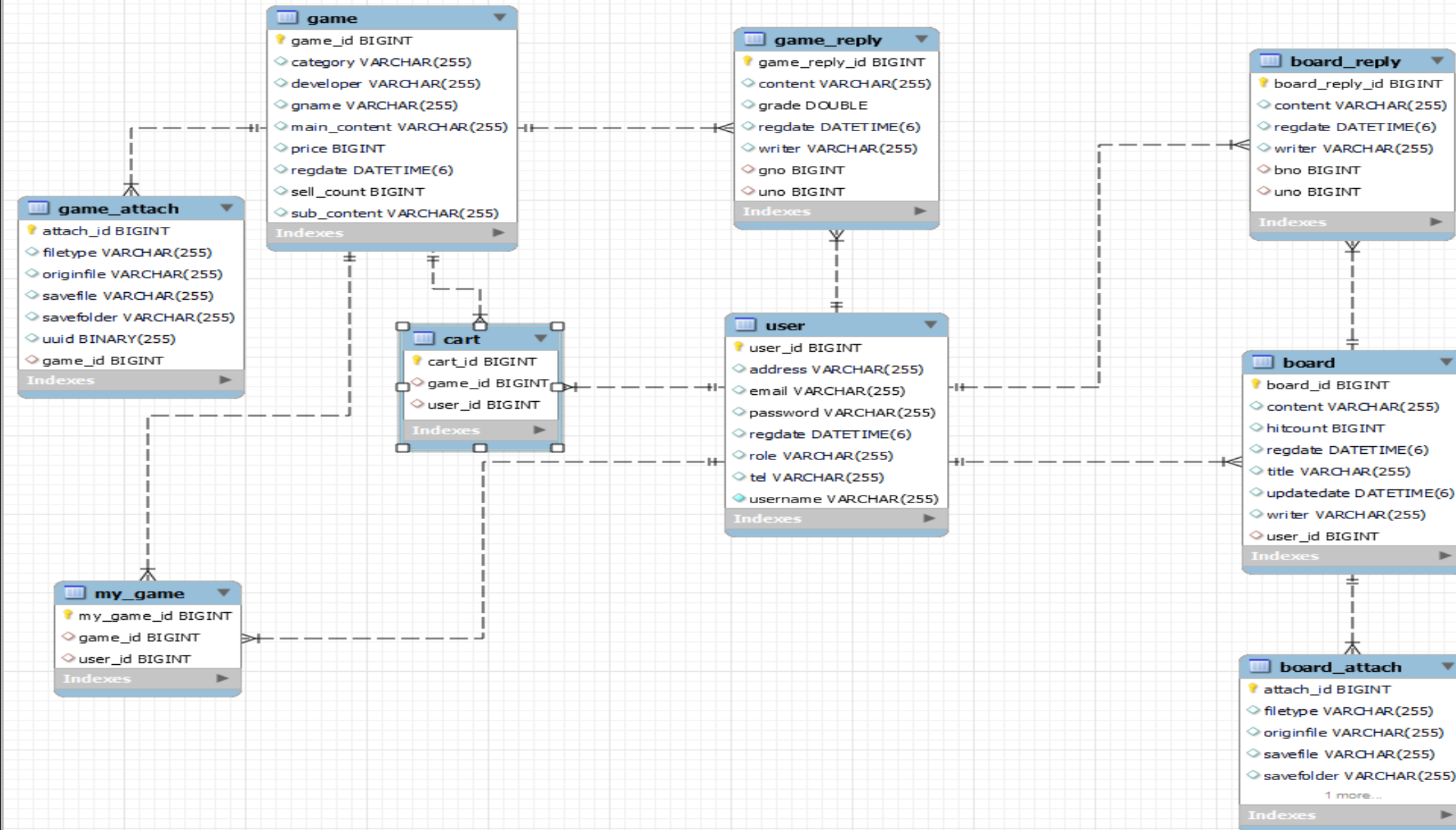
Entity - (board, cart, game, user.....)



```
1 package com.team4.game.entity;
2
3 import java.util.Date;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30 @Entity
31 @Data
32 @AllArgsConstructor
33 @NoArgsConstructor
34 public class Board {
35     @Id
36     @GeneratedValue(strategy = GenerationType.IDENTITY)
37     @Column(name="board_id")
38     private Long id;
39
40     private String title;
41     private String writer;
42     private String content;
43
44     private Long hitcount;
45     @CreationTimestamp
46     @Temporal(TemporalType.TIMESTAMP)
47     private Date regdate;
48     @CreationTimestamp
49     @Temporal(TemporalType.TIMESTAMP)
50     private Date updatedate;
51
52     @OneToMany(mappedBy = "board", fetch=FetchType.LAZY, cascade = CascadeType.ALL)
53     @JsonIgnoreProperties("board") // 무한참조 방지
54     private List<BoardReply> boardReply;
55
56     @ManyToOne(fetch=FetchType.LAZY)
57     @JoinColumn(name="user_id")
58     private User user;
59
60     @OrderBy("id desc")
61     @OneToMany(mappedBy = "board", fetch=FetchType.LAZY, cascade = CascadeType.ALL)
62     @JsonIgnoreProperties("board") // 무한참조 방지
63     private List<BoardAttach> boardAttaches;
64
65     @PrePersist
66     public void prePersist() {
67         this.hitcount= this.hitcount==null? 0 : this.hitcount;
68     }
69
70 }
71
```

1. Lombok을 사용한 @어노테이션을 사용하여 DB에 테이블생성 및 객체 생성.
2. @OneToMany, @ManyToOne 사용하여 다른 테이블과의 참조 관계 형성.

MySQL - ERD



Repository - (board, cart, game, user.....)



```
1 package com.team4.game.repository;
2
3+ import org.springframework.data.domain.Page;
4
5
6
7
8
9
10 public interface BoardRepository extends JpaRepository<Board, Long> {
11
12
13     Page<Board> findByTitleContaining(String keyword, Pageable pageable);
14
15     Page<Board> findByWriterContaining(String keyword, Pageable pageable);
16
17     Page<Board> findByContentContaining(String keyword, Pageable pageable);
18
19     Page<Board> findByTitleContainingOrWriterContainingOrContentContaining(String keyword, String keyword1, String keyword2 ,Pageable pageable);
20
21     void deleteByUserId(Long id);
22
23 }
24
```

```
1 package com.team4.game.repository;
2
3+ import java.util.List;
4
5
6
7
8
9
10
11 public interface GameReplyRepository extends JpaRepository<GameReply, Long>{
12
13-     @Query("select r from GameReply r join fetch r.game where game_id=?1")
14     public List<GameReply> gameReplyList(Long gameId);
15
16-     @Query("select AVG(grade) from GameReply r where gno = ?1")
17     Double findOneAvg(Long gameId);
18
19     public void deleteByUserId(Long id);
20 }
21
```

Repository 를 사용
-Entity에 의해 생성된 DB에 접근하는 메서드들을 사용하기 위한 인터페이스.

-@Query를 사용하여 sql 문을 작성

Service, ServiceImpl - (board, cart, game, user.....)



```
1 package com.team4.game.service;
2
3 import java.util.List;
4
12
13
14
15 public interface BoardService {
16
17     public void insert(Board board, User user);
18
19     void insertBoardAndFile(Board board, User user, List<BoardAttach> attachList);
20
21     public List<Board> boardList();
22
23     public Board findById(Long id);
24
25     public void update(Board board);
26     public void delete(Long id);
27
28     public Long count();
29
30     public void insetReply(BoardReply reply);
31
32     public List<BoardReply> replyList(Long bno);
33
34     public void replyDelete(Long id);
35
36     public Page<Board> findAll(Pageable pageable);
37
38     public Page<Board> findList(String field, String keyword, Pageable pageable);
39
40 }
41
42
```

```
1 package com.team4.game.service;
2
3 import java.util.List;
4
23
24
25
26 @Service
27 public class BoardServiceImpl implements BoardService {
28     @Autowired
29     private BoardRepository boardRepository;
30
31     @Autowired
32     private BoardReplyRepository replyRepository;
33
34     @Autowired
35     private BoardAttachRepository boardAttachRepository;
36
37     @PersistenceContext
38     EntityManager em;
39
40
41     @Transactional
42     @Override
43     public void insert(Board board, User user) {
44         // TODO Auto-generated method stub
45         board.setUser(user);
46         boardRepository.save(board);
47     }
48
49     //재입, 파일 동시에 등록
50     @Transactional
51     @Override
52     public void insertBoardAndFile(Board board, User user, List<BoardAttach> attachList) {
53         // TODO Auto-generated method stub
54         board.setUser(user);
55         Board board2=boardRepository.save(board);
56         System.out.println(board2);
57         if(attachList.size()>0) {
58             for(BoardAttach boardAttach:attachList) {
59                 boardAttach.setBoard(board2);
60                 boardAttachRepository.save(boardAttach);
61             }
62         }
63     }
64
65     @Override
66     public Page<Board> findList(String field, String keyword, Pageable pageable) {
67         System.out.println("field....."+field+"keyword....."+keyword);
68         // TODO Auto-generated method stub
69         Page<Board> boardList=null;
70         if(field.equals("") || field==null) {
71             boardList= boardRepository.findAll(pageable);
72         }
73         else if(field.equals("title")) {
74             boardList= boardRepository.findByTitleContaining(keyword, pageable);
75         }else if(field.equals("writer")) {
76             boardList= boardRepository.findByWriterContaining(keyword, pageable);
77         }else if(field.equals("content")) {
78             boardList= boardRepository.findByContentContaining(keyword, pageable);
79         }else if(field.equals("cwt")) {
80
81         }
82     }
83 }
```

Service Interface 만들기 , Service Interface 구현한 class 만들기

Controller - (board, cart, game, user.....)



```
1 package com.team4.game.controller;
2
3 import java.io.IOException;
4
5 @Log
6 @Controller
7 @RequestMapping("/board/**")
8 public class BoardController {
9
10     @Autowired
11     private BoardService boardService;
12     @Autowired
13     private UserService userService;
14     @Autowired
15     private MultipartModule multipartModule;
16
17     @GetMapping("register")
18     public void insert() {
19
20     }
21
22     // @PostMapping("insert1")
23     public String insert1(Board board) {
24         User user=userService.findByUsername(board.getWriter());
25         boardService.insert(board,user);
26         return "redirect:/board/list";
27     }
28     @PostMapping("insert")
29     public String insert(Board board, HttpServletRequest request,
30         @RequestParam("files") List<MultipartFile> files, @AuthenticationPrincipal PrincipalDetails principal) throws IllegalStateException, IOException {
31         List<BoardAttach> attachList = multipartModule.fileUploadBoard(request, files);
32         boardService.insertBoardAndFile(board, principal.getUser(), attachList);
33         return "redirect:/board/list";
34     }
35
36     //전체보기(페이징)
37     @GetMapping("list")
38     public String listPage(@RequestParam(name="field",defaultValue="") String field,
39         @RequestParam(name="keyword",defaultValue="")String keyword,
40         @PageableDefault(size = 10, sort="id", direction = Sort.Direction.DESC) Pageable pageable, Model model) {
41         Page<Board> searchList = boardService.findList(field,keyword,pageable);
42         long pageSize=searchList.getSize();
43         long blockSize=31;
44         long rowNm=searchList.getTotalElements();
45         long totPage=(long)Math.ceil((double)rowNm/pageSize);
46         long currPage=searchList.getPageable().getPageNumber();
47         long startPage=((currPage)/blockSize)*blockSize;
48         long endPage=startPage+blockSize;
49         if(endPage>totPage)
50             endPage=totPage;
51         if(totPage==0)
52             endPage=1;
53         boolean prev=startPage>0?true:false;
54         boolean next=endPage<totPage?true:false;
55
56         model.addAttribute("startPage", startPage);
57         model.addAttribute("blockSize", blockSize);
58         model.addAttribute("endPage", endPage-1);
59         model.addAttribute("prev", prev);
60         model.addAttribute("next", next);
61     }
62 }
```

만들어 둔 Service 를
Controller 에서 호출하기.

@Service 라는 어노테이션
으로 선언한 Class 를 사용
하기위해서 @Autowired
라는 어노테이션을 사용해
서 해당 서비스를 스프링에
등록시켜줌.

JSP, CSS - (main, board, cart, game, user.....)



```
1 @charset "UTF-8";
2 /*Basic design of webpage*/
3 @*{
4   margin:0;
5   padding: 0;
6   box-sizing: border-box;
7   font-family: 'Poppins', sans-serif;
8 }
9
10 @html{
11   background: url("../img/slider-bg-2.jpg");
12   background-position: center;
13   background-repeat: no-repeat;
14   background-size: cover;
15   height: 969px;
16 }
17
18 @body{
19   display: grid;
20   place-items: center;
21   text-align: center;
22   background-size: cover;
23 }
24 /*Basic design of login form*/
25 @.content{
26   width: 330px;
27   border-radius: 10px;
28   padding: 40px 30px;
29   margin-top: 100px;
30   box-shadow: -3px -3px 9px #aaa9a9a2,
31             3px 3px 7px rgba(147, 149, 151, 0.671);
32 }
33 /*Design the headings*/
34 @.content .text{
35   font-size: 25px;
36   font-weight: 600;
37   margin-bottom: 35px;
38   color: rgb(247, 233, 233);
39 }
40 /*Design the input spaces*/
41 @.content .field{
42   height: 50px;
43   width: 100%;
44   display: flex;
45   position: relative;
46 }
47
48 @.field input{
49   height: 100%;
50   width: 100%;
51   padding-left: 45px;
52   font-size: 18px;
53   outline: none;
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2
3
4
5 <%@ include file="../includes/headerb.jsp" %>
6
7 <div class="container">
8
9   <h2>${board.writer} 글쓰기</h2>
10
11   <div class="form-group">
12     <label for="id">아이디</label>
13     <input type="text" class="form-control" id="id" name="id" value="${board.id}" readonly="readonly">
14   </div>
15   <div class="form-group">
16     <label for="title">제목</label>
17     <input type="text" class="form-control" id="title" name="title" value="${board.title}" readonly="readonly">
18   </div>
19   <div class="form-group">
20     <label for="writer">작성자</label>
21     <input type="text" class="form-control" id="writer" name="writer" value="${board.writer}" readonly="readonly">
22   </div>
23   <div class="form-group">
24     <label for="file">파일</label>
25     <div>
26       <ul>
27         <c:forEach items="${board.boardAttachs}" var="fileInfo">
28           <li style="list-style:none">
29             <%-- 실제경로(RealPath)가 아니어도 찾아볼 --%>
30             
31             ${fileInfo.originfile}
32           </li>
33         </c:forEach>
34       </ul>
35     </div>
36   </div>
37
38   <div class="form-group">
39     <label for="writer">조회수</label>
40     <input type="text" class="form-control" id="writer" name="writer" value="${board.hitcount}" readonly="readonly">
41   </div>
42   <div class="form-group">
43     <label for="content">내용</label>
44     <textarea class="form-control" rows="5" id="content" name="content" readonly="readonly">${board.content}</textarea>
45   </div>
46
47   <div class="form-group text-right">
48     <c:if test="${principal.user.username==board.writer}">
49       <button type="button" class="btn btn-secondary btn-sm" id="btnUpdate">수정하기</button>
50       <button type="button" class="btn btn-secondary btn-sm" id="btnDelete">삭제하기</button>
51     </c:if>
52     <button type="button" class="btn btn-secondary btn-sm" id="btnList">목록보기</button>
53   </div>
54
55   <section class="mb-5" style="color: black;">
56     <div class="card bg-light">
57       <div class="card-body">
58         <!-- Comment form-->
59       </div>
60     </div>
61   </section>
62 </div>
```

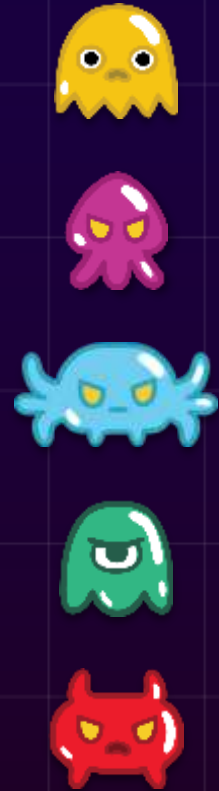
게임플랫폼 사이트인 만큼 프론트 부분도 중요하게 생각하였습니다.

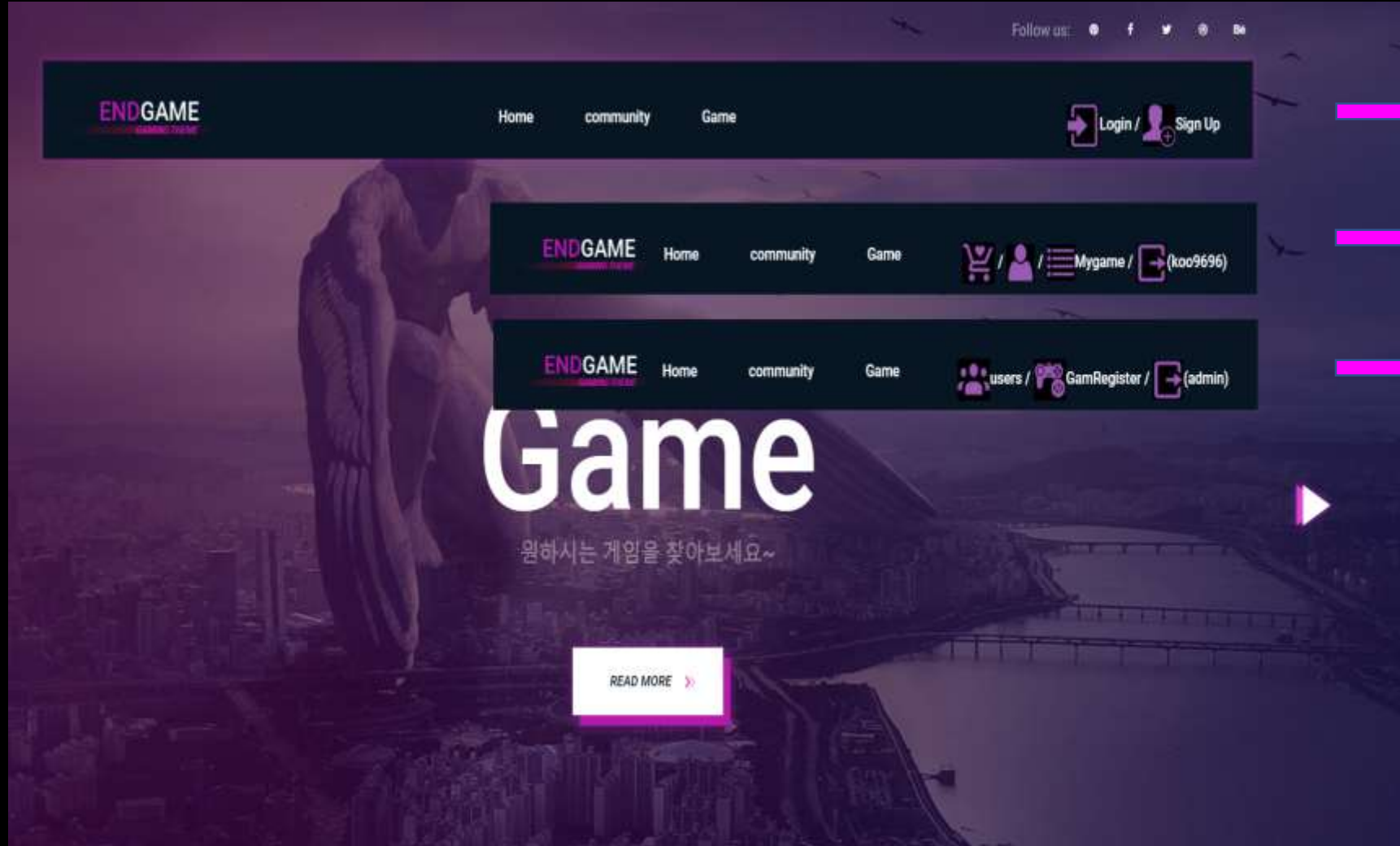
css를 사용하여 jsp에 <link>태그를 사용하여 사용, 메인화면에서 다른 페이지와 중복되는 부분은 header , footer 로 나누어서 <%@ ...@> 스크립트 태그를 사용하여 중복화면 호출.



stage.3

화면 및 기능 구현





비회원

회원

관리자

Spring Security 를 사용
해 권한을 부여, 권한에
따라 접근할 수 있는 영
역을 달리함.



ENDGAME
THE ULTIMATE GAMING EXPERIENCE

Login

ID

Password

Log in

Or

Facebook Google

ENDGAME
THE ULTIMATE GAMING EXPERIENCE

회원가입

아이디: 중복확인

비밀번호

비밀번호 확인

email

휴대폰번호를 입력하세요

주소를 입력하세요

회원가입



Community
Home / Community

Community(6)
글쓰기

| 번호 | 제목 | 작성자 | 작성일시 | 조회수 |
|----|---------------------------|---------|------------|-----|
| 7 | 게임하면서 들을만한 음악 추천드립니다~ | koo9696 | 2022-04-08 | 1 |
| 6 | 추천할만 게임 있나요?? 추천좀 해주세요~~~ | koo9696 | 2022-04-08 | 0 |
| 5 | 다들 점심식사 하셨나요?? | koo9696 | 2022-04-08 | 1 |
| 4 | 열등생 갯쟁이 같이 하실분 구원@@ | kk9797 | 2022-04-08 | 1 |
| 3 | 서른이턱 해어 너무 많아요ㅜㅜ | kk9797 | 2022-04-08 | 1 |
| 2 | 카트 벅어 너무 심하지 않나요?? | kk9797 | 2022-04-08 | 3 |

1
ENTER SEARCH
Search

ENDGAME

Home Games Reviews News Contact

비회원 : 글보기
 회원 : 글쓰기, 수정, 댓글
 관리자:글삭제

Community
Home / Community

글쓰기

제목:
ENTER TITLE

작성자:
koo9696

파일업로드:
파일 선택 선택된 파일 없음

내용:

Submit

ENDGAME

Home Games Reviews News Contact

검색과 페이징 기능

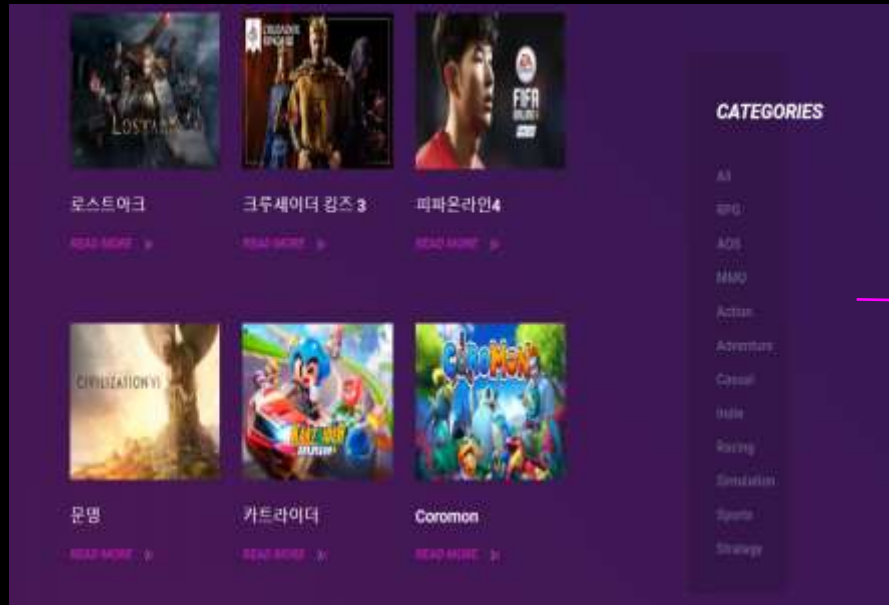


카테고리&검색 결과로 나오는 게임들 5개의 이미지를 띄움

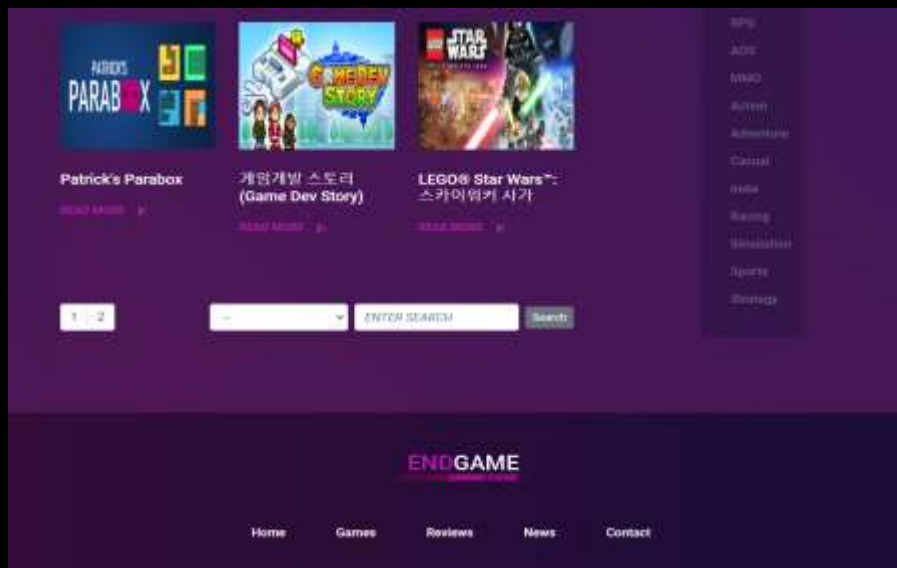
Games

[Home](#) / [Games](#)

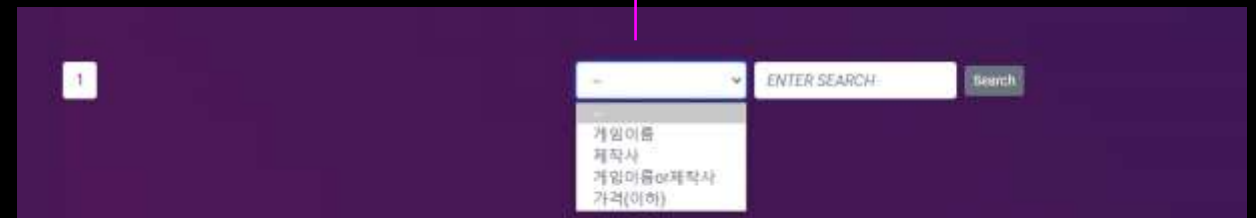




카테고리를 선택 시
해당 카테고리의 게임들만 보여줌



Community 와 마찬가지로 검색 과 페이징 기능이
가능하며 게임의 가격으로 검색했을 시 가격의 내림
차순으로 정렬





2022-04-17 / 개발사 : 넥슨게임즈 / 발매량 : 1

게임의 상세정보
(사진, 발매일, 개발자, 가격, 주설명, 보조설명,
판매량)

평점과 리뷰 작성 가능
평점의 평균을 바로 계산

로스트아크

설명1

설명2

Show

평점

3 / 5

가격

가격 15 원

카드결제

→ 카트에 바로 담기

1

JOIN THE DISCUSSION AND LEAVE A COMMENT

Reply Write



kkr9797 ★1 2022-04-08
왜 만들었냐요? 노골



koo9696 ★5 2022-04-08
재밌어요!!! @kkr9797 장난지나?

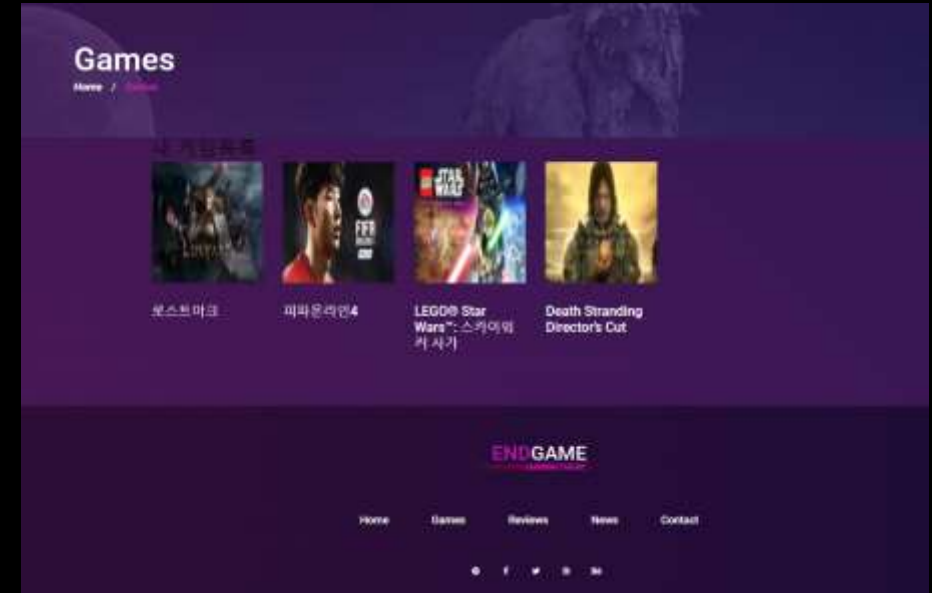


kkr9595 ★3 2022-04-08
그냥 할만함

장바구니 기능 및 결제



장바구니에서 게임 개별 삭제
가격 총합을 계산
장바구니나 내 게임 목록에 있
을 경우 카트에 추가 불가능
(중복 불가능)



결제 시 내 게임목록에 추가
제품 판매량 증가

2022-04-17 / 개발사 : 스마일게이트 / 판매량 : 1



User

Home / User

koo9696 유저정보

회원번호

아이디

이메일

주소

전화번호

등록일

[회원수정](#) [회원탈퇴](#) [로그아웃](#)

ENDGAME

www.endgame.co.kr

[Home](#) [Games](#) [Reviews](#) [News](#) [Contact](#)

내 정보 수정 가능
DB와 Session을 동시에 수정해서 회원정보를 수정하여도 다른 곳에서 principle을 써도 수정한 정보가 반영되어 있음.

회원 탈퇴 시 유저의 모든 활동 내역 삭제
(댓글, 작성글, 카트, 내 게임목록, 평점)



Games

Home / Games

게임등록

게임이름:

가격:

개발자:

카테고리: All

- AB
- RPG
- AOS
- 파탈
- MMO
- Action
- Adventure
- Casual
- Indie
- Racing
- Simulation
- Sports
- Strategy

파일업로드:

주제명:

모조내역:

Submit

ENDGAME

User

Home / User

User Total(4)

| 회원번호 | 아이디 | 이메일 | 주소 | 전화번호 | 권한 |
|------|---------|-----------------|---------------|--------|------------|
| 6 | jkk9595 | 1234 | 010-1234 | 다대포 | ROLE_USER |
| 5 | kkk9797 | kkk@naver.com | 010-1234-1234 | 부산 서면 | ROLE_USER |
| 4 | koo9696 | koo@naver.com | 01088663713 | 부산 기장군 | ROLE_USER |
| 1 | admin | admin@naver.com | 010 | 달해내 | ROLE_ADMIN |

1

ENDGAME

Home Games Reviews News Contact

수정하기

삭제하기

2022-04-17 / 개발사 : 스마일게이트 / 판매량 : 0

로스트아크

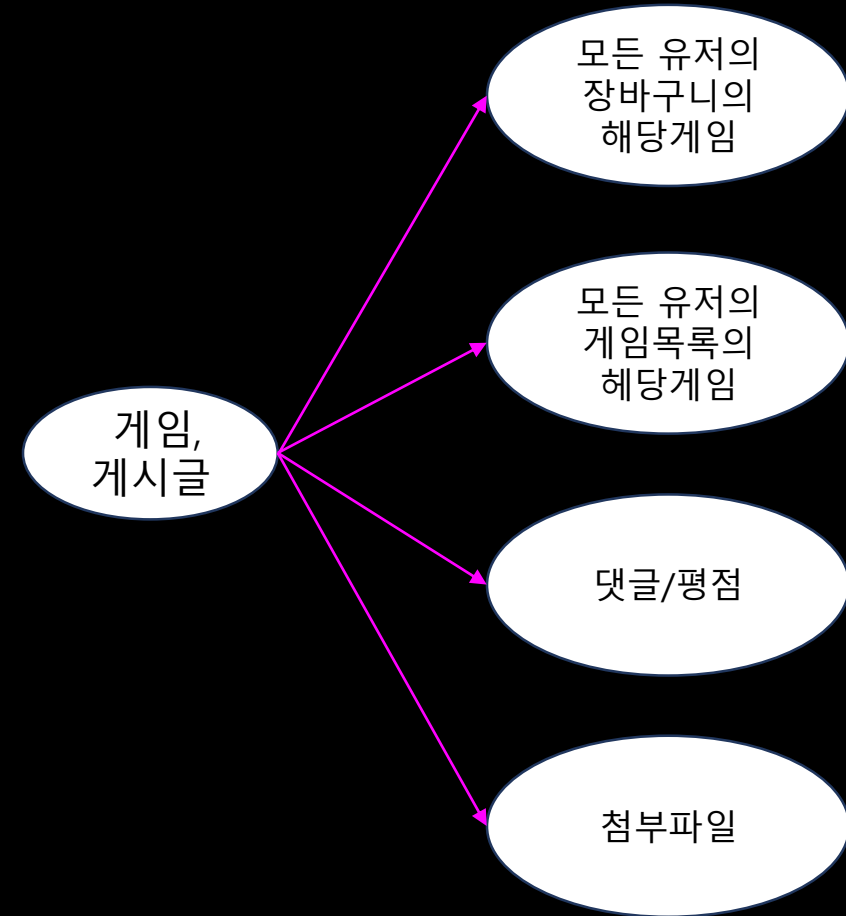
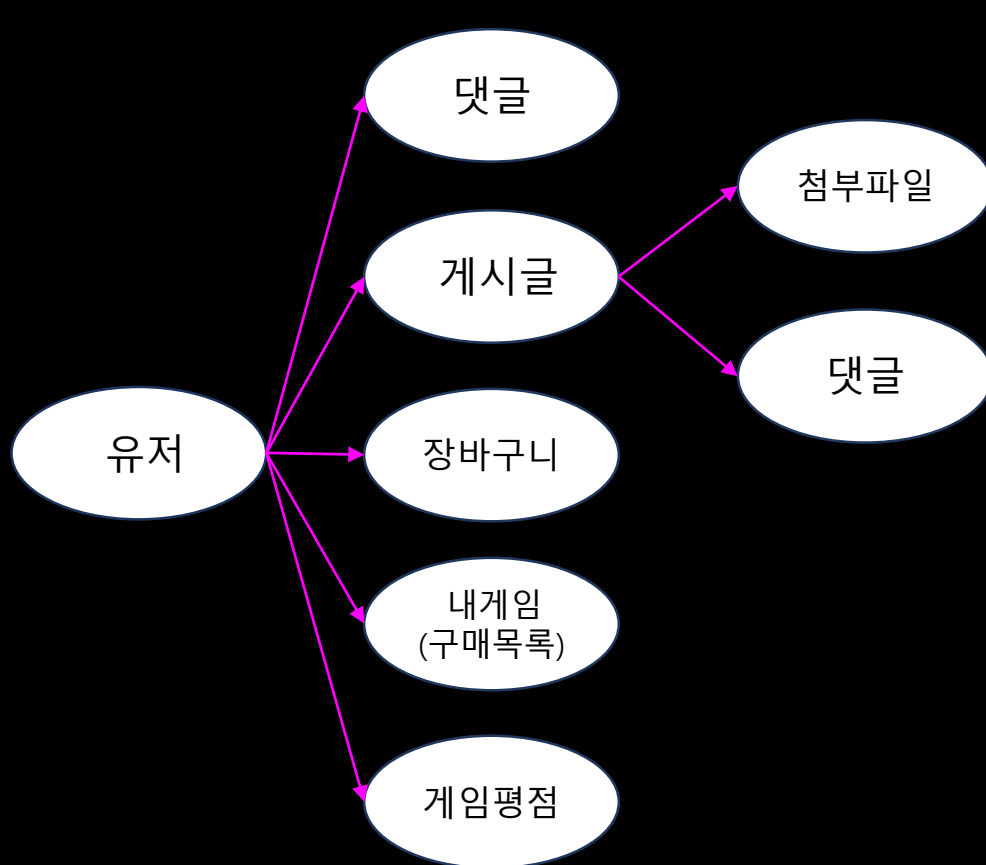
관리자: 게임등록, 수정 삭제
삭제 시 카트, 내 게임목록, 평점도 함께 삭제

회원삭제 : 탈퇴와 마찬가지로 유저가 활동한 모든
활동내역(글, 댓글, 카트, 구매, 평점)도 같이 삭제

삭제 절차



상위 요소를 삭제할 경우 하위요소 전부 삭제하도록 설정을 하여 오류제거
유저는 유저가 활동한 모든 내역 / 해당 게임, 게시글이 있는 모든 곳에서 삭제





stage.4

후기





정재규 :

처음에 나름 의견교환이 되었다고 생각했으나 막상 분업했던 것들을 합칠 때 여러 문제가 생기는 걸보니 부족했다는 것을 알았습니다. 다음 팀프로젝트를 할 때는 서로 잘 이해했는지 확인을 하면서 진행하려고 합니다.

김경래 :

Spring 과 JPA에 대한 개념이 충분치 않은 상태에서 개발을 진행하고 하다보니 어려움이 많이 있었습니다 . 개발만 중요한 것이 아니라 DB설계와 주소 등 기획과 구성의 중요성을 알았고 팀원들이 도와줌으로써 팀원 간의 소통, 협동의 중요성을 많이 느꼈습니다.

구주용 :

수업시간에 배웠던것을 활용하는 것도 중요하지만 무엇보다도 개인이 아닌 여러명에서 같이 진행되어야 하는 프로젝트이기 때문에 서로의 의사소통이 무엇보다도 중요하다고 생각 되었습니다. 저 또한 개발만 중요한 것이 아니라 DB설계와 주소 등 기획과 구성의 중요성을 알았고 팀원들이 도와줌으로써 팀원 간의 소통, 협동의 중요성을 많이 느꼈습니다.