**Portfolio (포트폴리오)**

**Project Info**

|  |  |
| --- | --- |
| **프로젝트 명** | SAILS |
| **개발 기간** | 2022.xx.xx ~ 2023.xx.xx |
| **참여인원** | 3명 |
| **담당업무** | 제안 참여 및 발표 , 기획 참여 , 게시판 기능 및 디자인 구현, 메신저 디자인 구현, 관리자 페이지(Q&A, 전체게시물, 신고게시물) 상세보기를 Ajax를 이용해 기능과 함께 구현 |

**Use Technology**

|  |  |
| --- | --- |
| **개발 환경** | Windows10 |
| **사용 도구** | Eclipse Oxygen.2 Release, Sql Developer |
| **사용 기술** | Java, Jsp, Spring4.3, Oracle 12xe, Ajax, jQuery |

**Introduction**

|  |
| --- |
| **트렌디한 주제와 합리적인 소비를 도와주는 SAILS사이트** |
| 요즘 e스포츠가 엄청난 인기를 이끌고 있습니다. 옛날에 비하면 주변에서 볼 수 있는 PC방의 수가 많아졌고  2018년에는 아시안게임 e스포츠 시범종목으로 채택되기까지 했습니다. 사회적 이슈가 되면 될수록 소비자들의 게임에 대한 관심이 자연스럽게 높아지고 소비 또한 비례하여 증가했습니다.  이런 점과 국내에 게임가격비교사이트가 많지 않은 점을 고려하여 주제를 선택해보았고 SAILS를 만들게 되었습니다.  “한 눈에 보기 편하며 좋은 페이지”가 저희가 생각하며 그려온 SAILS입니다.  SAILS는 사용자들이 편리하게 사용하게 하기 위하여 알아보기 쉽게 UI를 구현하였으며 사용자 친화적 UX를 구현하였습니다.    - 사진1. 구현화면 -  사진1은 실제 구현화면으로 메인 페이지와 검색 페이지입니다. 사용자들의 편리성을 위하여 한눈에 메뉴들을 보고 사용할 수 있도록 구현하였으며 실제 메인 페이지의 게임목록들은 스크롤을 이용하여 내리면 게임목록들이 바뀌며 표시됩니다. 메인이라는 말에 걸맞게 한 눈에 SAILS을 축약하여 볼 수 있습니다. |

**Develop Detail**

|  |
| --- |
| **제안** |
| 점점 증가하는 게임을 즐기는 사람들과 콘솔 게임을 즐기는 사람들이 나날이 늘어가는 지금 시대에 자기가 가지고 있는 게임기에 맞춰 게임을 좀 더 싸게 구입하기 위한 정보를 제공해주고 싶은 마음에 이런 아이템을 구상하였다. 이미 해외에는 여러 가격 비교 사이트가 있지만 한국어로 제공되는 사이트가 없으므로 한국어로 제공되는 가격 비교 사이트의 필요성이 있습니다. 한국은 전 세계 6위에 이르는 큰 규모의 게임 시장을 보유하고있습니다. 이 점은 SAILS의 필요성과 연관됩니다.  SAILS의 장점으로는 현재 세일 정보 관련 사이트는 해외 사이트밖에 없는 게 대부분이라 한국어로 만들어진 사이트는 한국 게이머들에게 접근성 높게 다가올 수 있습니다.  실시간 환율 정보로 인한 더 싼 마켓(국가별)에서 구입할 수 있는 정보를 제공을 할 것입니다.    - 사진2. 제안 프레젠테이션 -  사진2를 통한 SAILS의 필요성을 뒷받침하였습니다. |
|  |
|  |
| **기획** |
| 화면 기획 시 고려사항으로는 각 사용자 별 필요한 기능의 정의와 사용자 친화적 UX를 구현하고 •간편한 UI로 한눈에 보기 쉬운 홈페이지를 구성하여 사용자에게 편의를 제공할 수 있도록 하는 것, •가격비교를 쉽게 볼 수 있는 화면 구현,게시판 커뮤니티와 쪽지 기능을 통한 유저들의 의사소통 활성화가 최우선 고려사항 이였습니다.    - 사진 3. 기능 정의서 –  사진3은 기능정의서의 일부이며 크게 페이지로 분류하고 페이지에 들어간 기능을 중점으로 기능정의를 작성했습니다.    - 사진4. 메인 페이지, 검색 페이지 -    - 사진5. 로그인 페이지 –     * 사진6. 게임상세보기 페이지 –   사진 4, 5, 6과 같이 각각의 용도에 맞추어 사용자 친화적인 UI/UX를 구현하였습니다. |
|  |
| **DB 설계** |
| **[모델링]**  - <https://www.draw.io> 사이트를 사용하여 DB모델링  - PK, FK를 적용시켜 Table마다의 고유 번호를 만들고 또 외래키에 번호를 만들어 Table끼리 참조하여 사용하였습니다.    - 사진7. ERD –  **[테이블 정의서]**  - 모델링 데이터를 기준으로 테이블 정의서 작성        - 사진8. 테이블 정의서 - |

|  |
| --- |
| **사용 한 Framework** |
| **[Spring 4.3]**  - Annotation Driven을 통한 컨트롤러 호출  (Controller, Service, Repository)  **[My Batis]**  - My Batis와 Oracle 연동 구현 |

|  |
| --- |
| **개발** |
| **[동기화방식의 페이지 이동 구현]**  **- Form 태그를 이용한 값을 가지고 페이지를 이동**    **- 사진9. Form 태그에 저장해둔 hidden 값에 이벤트 발생 객체의 값을 받아와 저장하고 submit을 이용하여 값을 가지고 주소를 이동 –**  **[Ajax를 활용한 비동기식 데이터 처리 후 동적 웹 구현]**  - Ajax를 이용하여 비동기식 데이터를 처리하여 동적으로 화면을 구현하였습니다.      - **사진 10. Ajax를 활용한 비동기식 데이터 처리 후 html을 이용하여 list를 받아와서 동적으로 테이블 구현** -     * 사진11. 메인 페이지, 검색 페이지 –      * 사진 12. 게시판 페이지 –      * 사진 13. 관리자 페이지 -   **[html Append를 이용한 동적 웹 구현]**  **-** 사진 13의 게시물관리 상세보기 페이지에서의 Append를 이용한 동적 VIEW구현(자유게시판)    - 사진 15 - |
|  |
| **테스트 케이스** |
| **-** 사진 16 테스트 케이스 **–**  기능 테스트를 자료형식으로 올린 테스트 케이스의 일부이고 케이스 명을 기준으로 테스트 준비, 시나리오, 예상 결과 값, 에러 및 예외상황 처리(정상, 오류), 테스트 일자, 오류사항, 조치일자, 조치사항 순으로 작성하였습니다**.** |