|  |
| --- |
| **프로젝트 개발 계획서**  **[ 코드 관리 및 소통 게시판 ]** |
|  |

**□ 개발 개요**

○ 요약 설명

| **팀 명** | 3조 |
| --- | --- |
| **작품명** | 코드 관리 및 소통 게시판 |

○ 개발 목적 및 목표

- 편리하게 코드를 게시하고 소통하는 기능을 목표로 한다.

○ 개발 배경 및 동기

-개발자가 실제로 사용할 수 있는 웹을 만들자

-깃허브의 아쉬운 부분과 핵심적인 기능을 구현하자

**□ 팀 구성 및 업무 분장**

| **No** | **구분** | **성명** | **팀 내 담당 업무** |
| --- | --- | --- | --- |
| 1 | 팀장 | 노수빈 | 발표 및 ppt 제작  프론트앤드 디자인 |
| 2 | 팀원 | 금보성 | E-R 다이어그램 작성  게시판 코드하이라이트 기능  구현 |
| 3 | 팀원 | 오동용 | 차트 댓글 검색기능 |
| 4 | 팀원 | 조은경 | 유스케이스 다이어그램  회원가입, 로그인 기능 구현 |

○ 프로젝트 수행 일정

| No | 내용 | **2023년** | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 09/01(금) | | 09/04(월) | | 09/05(화) | | 09/06(수) | | 09/07(목) | |
| 1 | 테마 및 목표 설정 |  |  |  |  |  |  |  |  |  |  |
| 2 | BD설계 구성 |  |  |  |  |  |  |  |  |  |  |
| 3 | 게시판 구현 |  |  |  |  |  |  |  |  |  |  |
| 4 | 코드박스로그인 |  |  |  |  |  |  |  |  |  |  |
| 5 | 버그수정및 디자인 통일 |  |  |  |  |  |  |  |  |  |  |
| 6 | 댓글,검색  다크모드 |  |  |  |  |  |  |  |  |  |  |
| 7 | 빌표준비 |  |  |  |  |  |  |  |  |  |  |

**□ 유스케이스 다이어그램**

|  |
| --- |
|  |

**□ 유스케이스 명세서**

1. 회원가입

| **유스케이스** | 회원가입 |
| --- | --- |
| **개요** | 게시판을 이용하고자 하는 사용자의 ID, Password, 이름을 입력하고 DB에 중복되는 ID가 없을 경우 회원가입이 성공적으로 이루어진다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 회원가입 버튼을 클릭했을 경우 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1. 사용자가 회원가입 버튼을 클릭했을 경우 회원가입 페이지로 이동한다.  2. 사용자의 ID, Password, 이름을 입력한다.  3. 회원가입 하고자 하는 사용자의 ID를 DB의 데이터와 중복 검사를 진행한다.  4. 정상적으로 회원가입이 완료된다.  ※ 대안 흐름  1. 기존 사용자의 ID와 중복될 경우   1. 중복된 아이디가 존재한다는 오류 메시지가 도출된다. 2. 입력된 ID와 다른 ID를 입력한다. 3. 중복될 경우 1번으로 돌아간다. |
| **후행조건** | DB에 새롭게 회원가입한 사용자의 정보를 저장한다. |
| **비 기능적 요구사항** |  |

2. 로그인

| **유스케이스** | 로그인 |
| --- | --- |
| **개요** | 회원가입을 완료한 사용자는 로그인을 통해 게시판의 기능들을 정상적으로 이용할 수 있다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 회원가입을 통해 DB에 저장된 ID와 Password 를 올바르게 입력했을 경우 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1. 회원가입을 완료한 사용자는 본인의 ID와 Password를 각각 입력 란에 정확히 기재한다.  2. ID 및 Password가 정확히 입력 되었는지 DB와 비교/확인 한다.  3. 올바르게 입력 하였다면, 게시판 페이지로 이동한다.    ※ 대안 흐름  1. DB에 저장된 ID 및 Password가 사용자가 입력한 ID 및 Password와 다를 경우   1. 아이디 혹은 비밀번호가 일치하지 않다 라는 경고 메세지 도출된다. 2. 기존에 입력된 ID 및 Password와 다른 정보를 입력한다. 3. DB에 저장된 내용과 일치하지 않다면 1번으로 돌아간다. |
| **후행조건** | 로그인 버튼을 클릭하여 게시판 페이지에 접속한다. |
| **비 기능적 요구사항** |  |

3. 로그아웃

| **유스케이스** | 로그아웃 |
| --- | --- |
| **개요** | 로그인을 정상적으로 진행한 사용자가 로그아웃을 원할 경우 해당 기능을 사용할 수 있다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 로그아웃 버튼을 클릭했을 경우 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1. 로그아웃 버튼을 클릭하여 로그아웃을 진행한다. |
| **후행조건** | 로그인 페이지로 이동한다. |
| **비 기능적 요구사항** |  |

4. 차트 & 게시판 목록 보기

| **유스케이스** | 차트 & 게시판 목록 보기 |
| --- | --- |
| **개요** | 메인 페이지에 날짜, 사용자별 글 작성 횟수를 표시하는 그래프와 게시글의 목록을 표시한다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 메인페이지에 접속한다. |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1. 사용자의 로그인 여부를 확인한다.  2. 메인페이지에 접속한다.  3. DB에서 Codeboard테이블의 내용 정보를 불러온다.  4. 정보를 차트 또는 표를 이용하여 표시한다.    ※ 대안 흐름  1. 로그인을 하지 않고 메인 페이지 접속 시   1. 로그인 페이지를 띄운다. |
| **후행조건** |  |
| **비 기능적 요구사항** |  |

5. 게시글 작성

| **유스케이스** | 게시글 작성 |
| --- | --- |
| **개요** | 사용자가 게시글의 제목, 내용을 작성하고 등록할 수 있다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 로그인이 되어있어야한다.  게시글 작성 페이지로 이동한다. |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1. 게시글 작성 페이지로 이동한다.  2. 게시글의 제목과 내용을 적는다.  3. 필요하다면 코드추가 버튼을 눌러 코드를 작성한다.  4. 추가 버튼을 눌러 게시판에 등록한다.    ※ 대안 흐름  1. 제목란이 누락되어있을 경우   1. 제목을 입력해 달라는 오류 메시지가 도출된다.   2. 내용란이 누락되었을 경우   1. 내용을 입력해 달라는 오류 메시지가 도출된다. |
| **후행조건** | DB의 Codeboard 테이블에 저장된다. |
| **비 기능적 요구사항** |  |

6. 게시글 수정

| **유스케이스** | 게시글 수정 |
| --- | --- |
| **개요** | 사용자가 게시글의 제목과 내용을 수정 할 수 있다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 로그인이 되어 있어야 한다.  특정 게시글의 수정 페이지로 이동해야한다. |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1. DB에서 특정 게시글의 내용을 읽어와 출력한다.  2. 제목 또는 내용을 수정한다.  3. 필요하다면 코드추가 버튼을 눌러 코드를 작성한다.  4. 수정 버튼을 눌러 게시판에 등록한다.    ※ 대안 흐름  1. 제목이 누락되어있을 경우   1. 제목을 입력해 달라는 오류 메시지가 도출된다.   2. 내용이 누락되었을 경우   1. 내용을 입력해 달라는 오류 메시지가 도출된다. |
| **후행조건** | DB의 Codeboard 테이블에 저장된다. |
| **비 기능적 요구사항** |  |

7. 게시글 보기

| **유스케이스** | 게시글 보기 |
| --- | --- |
| **개요** | 게시글의 제목, 작성자 이름, 날짜, 게시글 내용, 댓글을 확인 할 수 있다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 메인 페이지에서 Details 버튼을 클릭했을 경우 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1. DB 에서 Codeboard 테이블의 내용 정보를 불러온다.  2. Comment 란에 댓글을 작성하거나 삭제 할 수 있다 |
| **후행조건** |  |
| **비 기능적 요구사항** |  |

8. 게시글 삭제

| **유스케이스** | 게시글 삭제 |
| --- | --- |
| **개요** | 게시글을 삭제한다 |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 게시글이 있어야 한다 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1.삭제페이지에서 삭제 버튼을 누르면 DB에서 데이터가 삭제 된다  2.index페이지로 돌아온다 |
| **후행조건** |  |
| **비 기능적 요구사항** |  |

9. 게시판 검색

| **유스케이스** | 게시판 검색 |
| --- | --- |
| **개요** | 게시글의 데이터를 입력해 검색할수 있다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 게시글이 있어야 한다 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1.게시글 이나 게시자의 텍스트 일부를 검색한다.  2.검색 조건에 맞는 게시글만 필터링 하여 Index페이지를 다시 로드한다  ※ 대안 흐름  1.검색어 없이 버튼을 누르면 전체 게시글을 다시 불러온다 |
| **후행조건** |  |
| **비 기능적 요구사항** |  |

10. 댓글쓰기

| **유스케이스** | 댓글쓰기 |
| --- | --- |
| **개요** | 디테일 페이지에서 익명의 사용자가 댓글을 남길수 있다 |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 게시글의 ID번호를 불러와야한다 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1.댓글을 작성하면 Comment Db에 content ID가 등록된다.  2.디테일 페이지 아래쪽에 남긴 댓글과 구분선 , 삭제버튼이 추가된다.  ※ 대안 흐름  1.댓글내용을 작성하지 않고 등록 버튼을 누를시 아무런 작동을 하지 않는다 |
| **후행조건** |  |
| **비 기능적 요구사항** |  |

11. 댓글 삭제

| **유스케이스** | 댓글 삭제 |
| --- | --- |
| **개요** | 디테일 페이지의 남긴 댓글을 삭제할 수 있다. |
| **관력 액터** | 사용자 |
| **우선순위** |  |
| **선행조건** | 댓글이 있어야 한다 |
| **이벤트 흐름** | ※ 정상적 사건의 흐름  1.댓글의 삭제 버튼을 누르면 Comment 테이블의 데이터를 삭제한다 |
| **후행조건** | 삭제할 댓글이 있어야 한다 |
| **비 기능적 요구사항** |  |