|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. 개발 환경  1. **개발툴**    * Eclipse JAVA EE IDE Photon Version을 기본 개발툴로 사용한다.    * Oracle Database11gExpress Edition을 Database 관리툴로 사용한다. 2. **버전관리계획**    * Git 및 GitHub를 이용하여 단위 개발 및Version 관리를 한다.    * 테스트 서버의 관리 및 작업 내용 반영은 최종 개발 책임자가 작업한다.    * 개발 진행중인 작업 내용의 경우 개인 단위의 백업관리가 이루어지며, 개발 완료 작업 내용과 일(daily)단위의 백업은 최종 개발 책임자가 맡는다. 3. **데이터베이스 접근 계획**    * 모든 개발자는 하나의 데이터베이스 서버를 공유하여 사용한다.    * 데이터베이스의 접근은 WAS의 자체 Database Pooling모듈을 사용하며, DB 리소스는 독립적으로 획득할 수 없다.  2. 명명 규칙  1. **디렉토리 명명 규칙**    * 디렉토리 명칭은 영문으로 사용한다.    * <context root>하위에 메뉴의 깊이를 기준으로 디렉토리를 생성하며, 약어를 사용하지 않는다. 2. **JSP 파일 명명 규칙**    * 파일의 주요 기능을 표기한다.  |  |  |  | | --- | --- | --- | | XXXXX\_YYYYY.jsp  ⓐ ⓑ ⓒ | | | | ⓐ | 기능명 | 핵심 기능에 대한 약어 | | ⓑ | 구분자 | ‘\_’ | | ⓒ | 디렉토리 명칭 혹은 주요 기능 | (예시)  1. 게시판(board) : main\_board.jsp 등  2. 채널(channel) : add\_channel.jsp 등  3. 프로젝트(project) : index\_project.jsp, main\_project.jsp, add\_project\_user.jsp 등  4. 사용자(user) : confirm\_user\_email.jsp, regist\_user\_by\_email\_api.jsp, update\_user\_profile.jsp 등  5. 기타 : canvas.jsp, todoList.jsp 등 |  * 구분자를 제외한 모든 섹션은 영문자로 시작하며, 소문자와 대문자를 적절히 혼합하여 사용하며, 가급적 의미를 알 수 없는 약어를 사용하지 않는다. * 사용되는 확장자는 아래와 같다.  |  |  | | --- | --- | | **파일 형식** | **파일 확장자** | | Java Servlet Page(JSP) | .jsp | | Cascading Style Sheet(CSS) | .css | | Syntactically Awesome Style Sheets | .scss | | JavaScript | .js | | Hyper Text Markup Language | .html | | Web Resource | .gif, .jpg, .png | | JSON | .json | | Extensible markup language | .xml | | JAVA | .java | | Java Archive | .jar |  1. **패키지 명명 규칙 및 자바 파일 명명 규칙**    * 패키지명은 3 자리 혹은 4자리 이상으로 정의한다.      + com.nomwork.project.web : Controller 파일 정의      + com.nomwork.mybatis : aop 및 log 파일 정의      + com. nomwork.project.dto : dto 파일 정의      + com. nomwork.project.dao: Service 및 DAO 파일 정의    * 자바 파일은 3자리 이상의 package명 하위 단에 작성한다. 2. **자바스크립트 파일 명명 규칙**    * 파일이 사용되는 기능명으로 명명한다      + main\_board : 게시판      + add\_channel : 채팅방 추가      + regist\_user\_by\_api: api를 통한 회원 가입   예) board\_setting.js : 게시판 기능에 사용되는 자바스크립트 파일   1. **클래스 명명 규칙**  * DAO, DTO의 경우 해당 데이터베이스 테이블명칭을 그대로 갖는 클래스 이름을 사용하며, com.nomwork.(기능명).dto, com.nomwork.(기능명).dao 을 상위 패키지로 갖는다.   예1) com.nomwork.map.dto*.MapDto.class*  예2) com.nomwork.map.dao*.MapDao.class*   * Jar 파일의 경우 WEB-INF/lib 폴더에 위치 시킨다.  1. **객체 및 변수 명명 규칙**  * 객체의 경우 해당 객체가 페이지 내에서 유일할 경우 첫글자를 소문자로 바꾼 클래스명을 사용한다.(단, 상수는 예외)   - 예1) 클래스 ChannelDto의 객체 c\_dto   * 변수의 경우 단순 증감이며 for문 등의 제어문 내에서 선언되는 경우 <i,j,k> 등의 간단한 소문자를 사용한다.(‘p’는 예약된 변수로써, PreparedStatement의 index에서 사용한다.)   - 예) *for(int i=0;i<10;i++)*   * 특정 정보를 담고 있는 변수인 경우, 해당 정보의 성격을 나타낼 수 있는 명칭을 사용한다.   - 예) int insert\_text\_res*= “insert text의 결과”;*   * HTML 내의 <input> 태그로 사용되는 필드 값을 전달받을 경우 동일한 이름의 변수에 할당한다.*;*  1. **데이터베이스 명명 규칙**  * 테이블 명칭 : 테이블의 용도를 알수있도록 명명한다.   예) 회원 정보 테이블 : USERINFO   * 필드 명칭 : 외부참조키가 아닌 경우 ‘일부 혹은 모든 TABLE NAME’+’필드 정보’를 붙이는 형태로 사용하며, 외부참조키(FK)인 경우는 해당 FK의 원본 필드 명칭을 그대로 사용한다.   예1) USERNO   * 시퀀스 명칭 : 말머리로 <필드명>+’SEQ’을 붙이는 형태로 사용한다   예 1) USERNOSEQ 3. 개발 구조  1. **디렉토리 구조**  * 경로는 클래스 및 인터페이스 파일의 경우 java resource 폴더를, 그 외 파일들은   <context root>를 상위로 갖는다.   * Directory 이름은 의미있는 하나의 영문 단어로 구성하고 전부 소문자로 표기하도록 함. 또한 혼동을 피하기 위해 가급적 약어를 사용하지 않도록 함. * 다음은 디렉토리 구성 규칙의 예임.   + - JSP 파일 : <context root>+ ‘/WebContent/(기능명폴더)/jsp’   ex) ‘/WebContent/calendar/jsp’   * + - JAVA Class 파일 : <java resource> + <package>     - JAVA Source 파일 : <java resource> + <package>     - XML 파일 : <context root>+‘/WEB-INF/’     - 이미지 파일 디렉토리 : <context root> + ’/resources/image’     - JavaScript 디렉토리 : <context root> + ‘resources/js’     - CSS 디렉토리 : <context root> + ‘resources/css’  4. 코드 구조  1. **JAVA 코드 구조**    * DAO 구조 예제 (MyBatis 사용)      * + DTO 구조는 일반적인 BEAN의 형태를 갖는다  1. **JSP 및 HTML 코드 구조**  * 예제 |