EMB0000407071b8

Story Of Dosic

EMB0000407071b8

**[Member List]**

**김현재 – 게시판 관리**

**문승훈 – DB & 로그인**

**이옥희 – 채팅기능 구현**

**장지영 – 챗봇 구현**

**목차**

1. 개요

1) Story of Dosic 이란?

2) 개발도구

3) Layered Architecture

4) Class Diagram

5) Database ERD

2. 기능별 설명

1) 로그인

- 구현사항

- 초기계획과 달라진 점

2) 채팅

- 구현사항

- 초기계획과 달라진 점

3) 게시판

- 구현사항

- 초기계획과 달라진 점

4) 챗봇

- 구현사항

- 초기계획과 달라진 점

3. 실행 매뉴얼

1) 실행 영상

2) 실행 방법

**1. 개요**

**1) Story of Dosic(SoD) 이란?**

- Stoty of Dosic(Department of smart ict convergenc)

- 스마트 ICT융합공학과만의 커뮤니티 웹사이트

- 스융공의 친목 도모 및 정보 공유

**2) 개발도구**

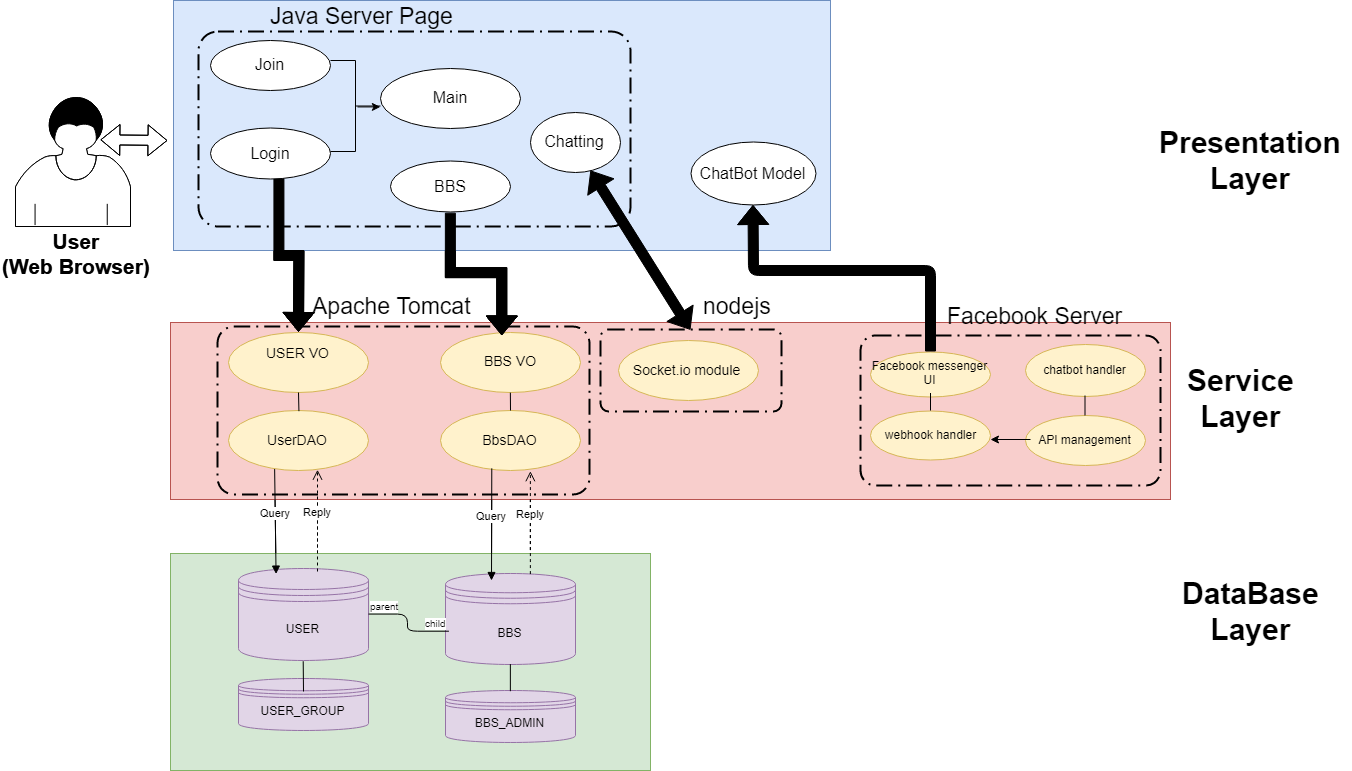
Eclipse Runtime Target: Web Apache Tomcat 8.5

DB: Mysql 8.0

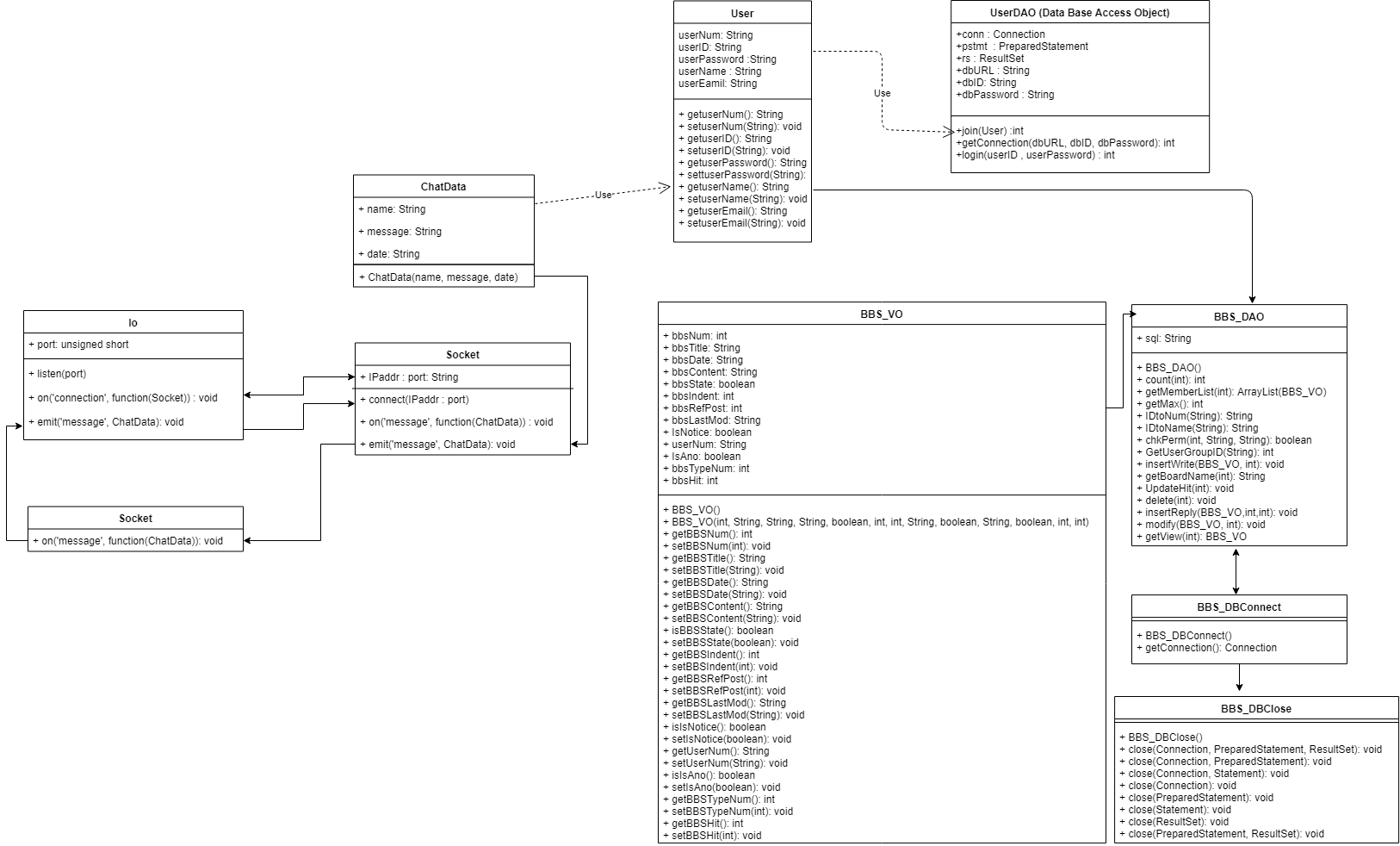
JSP-DB 연동 :mysql jdbc driver 8.0.16

socket : node.js socket.io 모듈

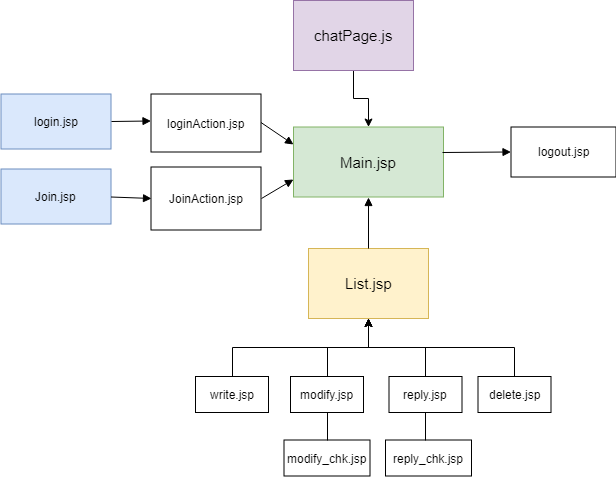
**3) Layered Architecture**



4) Class Diagram(클래스 다이어그램 / 챗봇 제외)

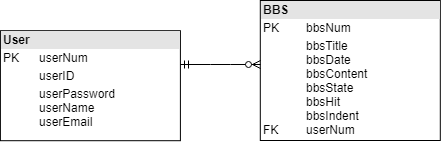


**5) Web Page 구성**

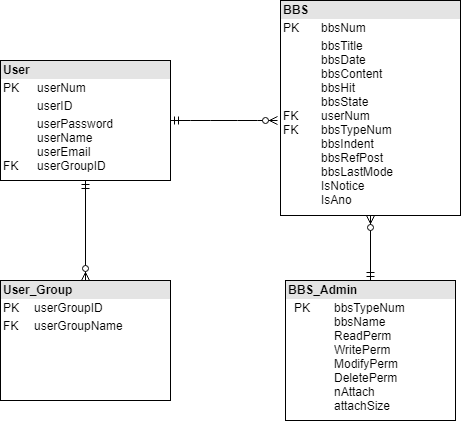
****

**6) Database ERD**

**[초기 ERD]**

****

**[개선 ERD]**



변경 내용: User\_Group / BBS\_Admin Entity 추가

User, BBS: 로그인, 회원가입, 게시판기능에 사용될 javaBeans (DB Field 이름과 동일)

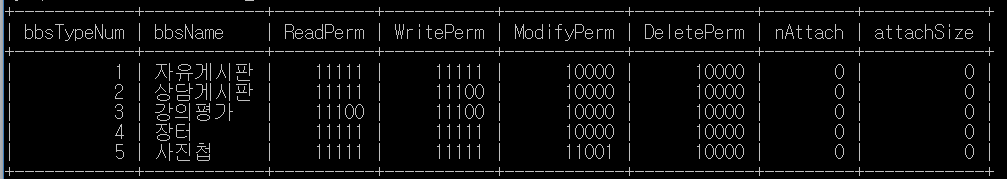
[User\_Group]

각 유저가 속한 회원 종류를 표시하고자 하는 Entity



[BBS\_Admin]

게시판 종류별로 고유번호를 부여하고 읽기, 쓰기, 수정, 삭제, 첨부파일 개수와 파일크기를 명세하는 Entity

****

User\_Group, BBS\_Admin의 필요성

-> 각 회원 그룹을 설정함으로써 종류별 게시판 마다 접근 권한을 달리할 수 있음.

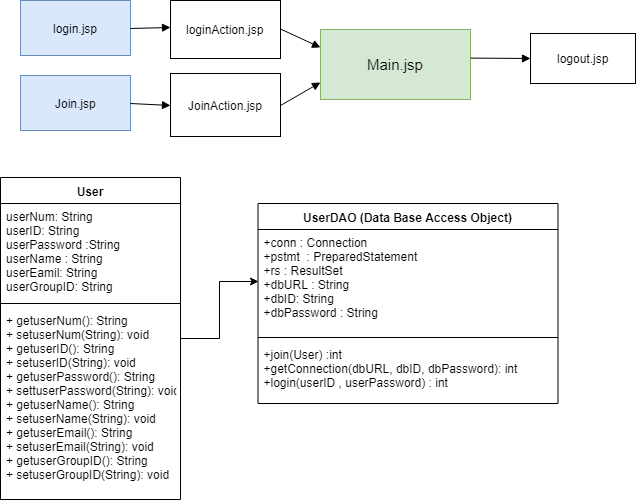
Ex) 상담게시판의 ‘쓰기 권한’(WritePerm)은 각 자리수별 권한의 ON/OFF를 의미함.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **관리자** | **학생회** | **학생** | **교수** | **조교** |
| 1(ON) | 1(ON) | 1(ON) | 0(OFF) | 0(OFF) |

**2. 기능별 설명**

**1) 로그인**

- 구현사항



① User

JAVA beans 여러 JSP 페이지에 사용 될 수 있다.

DB의 UserTable과 동일하게 아이디, 비빌번호, 학생명, 학번을 멤버 변수로 가지고

이에 대한 getter, setter 메소드로 구성된다.

② UserDAO

User가 회원가입 기능을 수행할 때 DB에 유저 정보를 등록하고

로그인 기능을 수행할 때 DB에 등록된 유저 정보와 대조하는 역할을 한다.

성공적으로 회원가입, 로그인이 완료될 경우

main.jsp 로 이동한다.

-Connection conn : 데이터베이스에 접근하기 위한 객체

-PreparedStatement pstmt : 정해진 SQL문을 이용해서 상태정보(ID)를 가져온다.

-ResultSet rs: 데이터베이스에 접근한 결과를 저장하는 객체

- dbURL: Mysql 서버 주소

- dbID, dbPassword: 데이터베이스에 저장된 ID, 패스워드와 비교할 값

- int getConnection(dbURL, dbID, dbPassword)

mysql driver와 연결하여 DB에 실제적으로 접근할 수 있도록 하는 메소드

- int join(User user)

회원가입을 수행할 메소드로 SQL 쿼리문을 통해 DB에 유저 정보를 저장한다.

로그인 수행결과에 따라 리턴 값을 반환한다

★ 이때, DB에 사전등록되지 않은 학번의 유저는 가입할 수 없다. (학번 Check Box 활용)

- int login(userID, userPassword)

로그인 기능을 수행할 메소드로 SQL 쿼리문을 통해 DB에 등록된 유저 정보와 대조한다.

로그인 수행결과에 따라 리턴 값을 반환한다

ex) return 값 -> 1: 로그인 성공/ 0: 비밀번호 불일치 / -1: ID가 존재하지 않음.

-2: 데이터베이스 오류

loginAction.jsp: userDAO 객체를 생성하여 아이디와 비밀번호를 입력받는다.

로그인 실패시 -> 경고메시지를 띄우고 login.jsp로 되돌아간다.

로그인 성공시 -> main.jsp로 이동한다.

joinAction.jsp: userDAO 객체를 생성하여 학번, 아이디, 비밀번호, 이메일을 입력받는다.

회원가입 실패시 -> 경고메시지를 띄우고 join.jsp로 돌아간다.

회원가인 성공시 -> index.jsp로 이동한다.

logoutAction: 세션을 종료시켜 로그인 된 유저를 로그아웃시키고 login.jsp로 이동한다.

* **초기계획과 변경 사항**

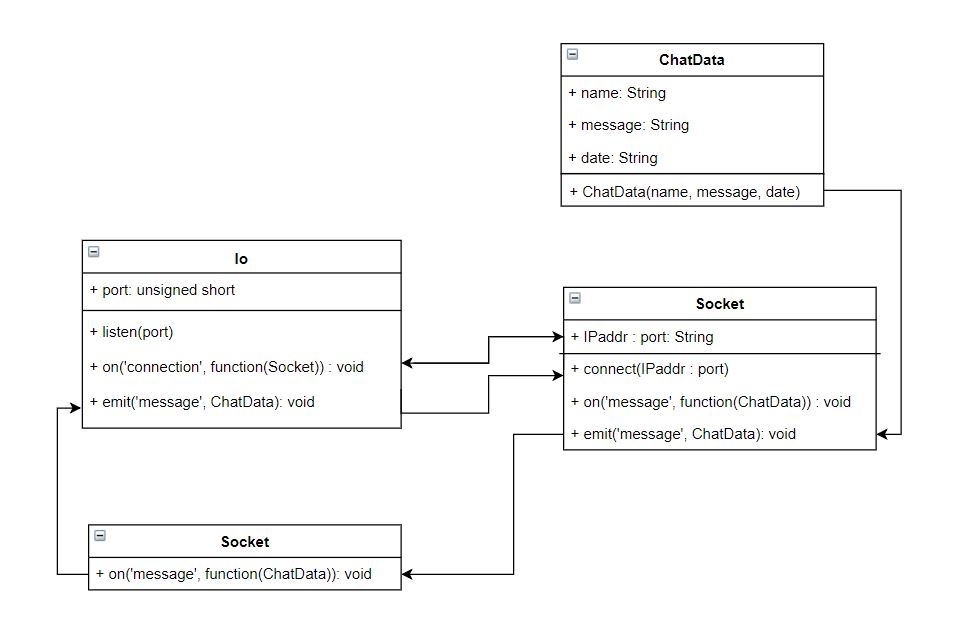
|  |  |
| --- | --- |
| **초기 계획** | **변경 내용** |
| 학번 정보 사전입력을 통한 외부인(스마트 ICT융합공학과 학생 외의) 가입방지 | 학번정보 사전입력 X |
| [변동 원인]  주임 교수님의 동의가 있다 하더라도 각 학생들의 이름과 학번은 개인 정보이기 때문에 모든 개개인들에게 동의를 받아야만 입력 가능. | |

추후 개선 계획: 학과 학생들의 동의를 받기 or 학생 학번의 범위 (ex.20171300 ~ 20171399)를 입력.

* 이 외에 변동사항 없이 모두 구현완료.

**2) 채팅**

- 구현사항



클라이언트 측

서버 측

* **초기계획과 변경사항**

|  |  |
| --- | --- |
| **초기 계획** | **변경 내용** |
| - Apache Tomcat 서버, main.jsp 페이지 내에 채팅 기능 삽입. | - Apache tomcat 서버와는 별도로 node.js 서버 실행  Tomcat 서버의 포트번호: 8080  Node.js 서버의 포트번호: 5003 |
| [변동 원인]  사용자의 정보를 받아오고 채팅을 한 화면의 웹페이지 내에서 실행하기 위하여  Apache Tocmat – node.js 서버간 연결해야하는 문제가 있었음.  내부에 들어가는 채팅 기능은 node.js의 socket.io모듈을 사용해서 구현했기 때문에 별도의 서버실행이 불가피함. (웹 페이지 서버 – 채팅 서버간의 분리) | |

* 이 외에 변동사항 없이 모두 구현완료.

**① ChatData : Object – 서버와 클라이언트가 message 이벤트로 주고 받을 데이터**

name : 채팅 참여자 이름(실명)

message : 보내고자 하는 채팅 내용

date : 채팅 보낸 시간

ChatData(name, message, date) : ChatData의 생성

**② Socket : Object – 클라이언트 소켓 객체**

IPaddr : port : 서버의 IP address와 서버 소켓의 포트번호

connect(IPaddr : port) : 서버 소켓에 연결요청, 서버의 IP address와 port번호를 인자로 넘겨준다.

on(‘message’, function(ChatData))

* message : 사용자 정의 이벤트, 서버로부터 message이벤트를 받는다.
* function(ChatData) : message 이벤트를 서버로부터 받았을 때, 화면에 Data의 내용(name, message, date)를 보여주는 함수

emit(‘message’, ChatData)

* message : 사용자 정의 이벤트, message 이벤트를 서버에 전달한다.
* ChatData : message 이벤트를 서버에 전달할 때 함께 전달할 데이터

**③ Io : Object – 서버 소켓 객체**

port : 서버 소켓의 포트번호

listen() : 서버 소켓을 생성 및 실행한다.

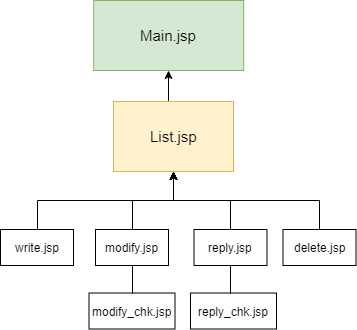
on(‘connection’, function(Socket))

* connection : 클라이언트가 연결할 때 발생하는 이벤트
* function(Socket) : 접속한 클라이언트의 정보를 가진 소켓 객체 생성

emit(‘message’, ChatData)

* message : 사용자 정의 이벤트, message 이벤트를 서버에 접속한 모든 클라이언트에게 전달한다.
* ChatData : 한 클라이언트에게서 받은 데이터(name, message, date)를 이벤트를 발생시킨 클라이언트를 포함한 모든 클라이언트에게 전달한다.

**3) 게시판**

- 구현사항

Write(쓰기), modify (수정), reply(답변), delete(삭제) 는 버튼으로 표시되며

모든 기능 수행 후의 결과는 List.jsp에 반영된다.

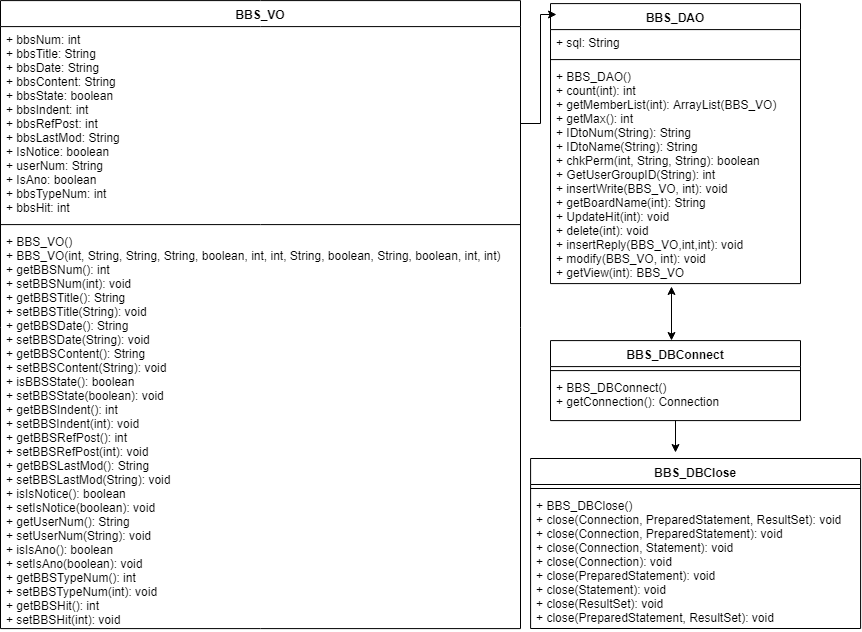
자유게시판, 상담게시판, 장터, 사진첩 등의 모든 게시판의

쓰기, 수정, 답변, 삭제 기능은

리스트 하위의 .jsp페이지에서 수행되고

수정과 답변의 경우 \_chk.jsp 에서 내부적으로 접근권한을 체크한다.

접근 권한이 없는 글에 대한 명령을 수행하려 할 경우 경고메시지가 출력된다.



**[메소드 설명]**

count(int): 게시물 개수 카운트

bbsTypeNum(게시판 번호)를 입력받아 삭제되지 않은 게시물의 개수를 구한다.

getMemberList(int): 게시판 게시물 정보 로드

bbsTypeNum(게시판 번호)를 입력받아 게시글 목록에 표시될 정보(제목, 조회수 등)를 불러온다.

getMax(): 게시물 최대 번호 카운트

bbs 테이블의 최대 bbsNum값을 불러온다.

IDtoNum(String): 유저 학번 로드

유저의 ID를 통해 학번을 불러온다.

IDtoName(String): 유저 이름 로드

유저의 ID를 통해 이름을 불러온다.

chkPerm(int, String, String): 권한 확인함수

유저의 이름과 게시판 번호, 확인할 권한을 입력받고, 해당 게시판에서 유저가 권한이 있는지 확인한다.

GetUserGroupID(String): 유저 그룹 확인함수

유저가 어느 그룹에 속해있는지 확인한다.

insertWrite(BBS\_VO, int): 게시물 삽입함수

BBS\_VO 객체를 이용해 새로운 게시물을 삽입한다.

getBoardName(int): 게시판 이름 로드

bbsTypeNum(게시판 번호)을 통해 게시판 이름을 불러온다.

UpdateHit(int): 게시물 조회수 증가

게시물을 볼 때마다 조회수를 1씩 증가시킨다.

delete(int): 게시물 삭제

해당 게시물을 삭제한다. (bbsState를 1로 변경)

insertReply(BBS\_VO,int,int): 답글 삽입

답글 게시물을 삽입한다.

modify(BBS\_VO, int): 게시물 수정

해당 게시물을 수정한다.

getView(int): 게시물 로드 함수

해당 글 정보를 받아와 객체를 받아온다.

* **초기계획과 변경사항**

|  |  |
| --- | --- |
| **초기 계획** | **변경 내용** |
| - 회원 권한, 게시판 종류를 User, BBS Entity에 하나의 필드로 관리 | - 회원 권한을 하나의 필드가 아닌 별도의 User\_Group + BBS\_Admin Entity 추가 |
| [변동 원인]  각 유저의 권한과 게시판 종류를 Entity 단위가 아닌 Field의 값으로만 다루게되면  각 게시판별 페이지를 따로 생성해야하는 구현의 어려움이 생김.  -> DB 설계 보완(별도의 Entity 추가)을 통해 각 유저와 게시판의 그룹, 종류를 설정하여 유지 & 보수가 용이하게함. | |

* 이 외에 초기계획과 달라진 변동사항은 추가적으로 없으나,

게시판별 게시물에 대한 목록 표시, 추가, 삭제 등의 기능을

DB와 연동하여 페이지에 표시하는 기능을 아직 구현하지 못함.

**4) 챗봇**

**- 구현사항**

스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **Class Domain: 챗봇의 기본단위인 도메인생성과 삭제에 관한 클래스**

void CreateDomain(): 도메인을 생성한다

* String DomainName: domain 이름을 저장하는 변수
* String DomainCode: domain코드를 저장하는 변수
* String Language: 사용할 자연어 변수

boolean Manage(String DomainName): 해당 도메인의 대화의 작업을 관리한다. 대화가 잘 업데이트 되면 true를 error가 발생하면 false를 return한다.

* boolean check

void DeleteDomain(): 도메인을 삭제한다.

1. **class Dialog: 챗봇의 대화 생성하는 클래스**

void CreateDialog():

* String DialogName: 대화 생성의 이름 저장하는 변수
* String DialogType: 사용자 발화 유형의 상태를 정의하는 변수

void RegisterA(): 대화의 답변을 등록한다.

* StringA: 답변들을 저장할 array

void RegisterQ(): 사용자가 질문할 것이라 예상되는 질문들을 등록한다.

* StringQ:: 질문을 저장할 array

void GetQ(): 실제 사용자에게 질문을 받는다

* String Q: 사용자가 한 질문을 저장할 변수

String GiveA(String Q): 사용자에게 받은 질문을 가지고 그에 해당하는 답변을 준다.

1. **class Build: 챗봇을 학습하는 클래스**

void Build(String Adjust): 챗봇의 대화 모델을 빌드 한다.

* String Adjust: 빌드 내용 수정할 것이 있으면 개발자로부터 받는다. 없다면 null.

String BuildState(): 현재 빌드 되고 있는 현황 확인

1. **Create\_API: API를 생성하는 클래스**

void API\_gateway(): product에서 project를 만드는 메소드

* String ProductName: 생성할 product이름

void Create\_API(): API를 생성하는 메소드

void Stage\_API(): stage API를 생성

* String StageName: 만들 stage이름을 저장하는 변수
* String Endpoint: stage의 endpoint저장하는 변수

1. **Invoke\_URL**

void Copy\_IU(): invoke url을 복사하는 메소드

* String Urladdress: 복사된 url의 주소를 저장하는 변수

1. **페이스북 메신저와 연동**

* 이 부분은 제가 직접 하는 부분이 아니고 페이스북 측에서 해주었습니다.

[초기 계획 명세]

재학생에게 정보전달

* 재학생에게 해당하는 질문과 응답

1. 건국대학교 내의 건물 위치: 캠퍼스 맵 제공
2. 과사무실의 관련한 정보 제공

수험생에게 정보전달

* 직접적인 정보보다는 대부분 웹사이트 링크를 통해 정보를 전달한다. -> 수험생들에게는 정확한 정보가 중요하기 때문에 개인의 주관이 들어간 답변을 내놓을 수 없다.
* 수험생에게 해당하는 질문과 응답

1. 건국대학교 입학관련 질문: 건국대학교 입학처 홈페이지
2. 건국대학교 위치: 폐쇄형 답변을 통해 교통수단에 따른 답변
3. 건국대학교 내의 건물 위치: 캠퍼스 맵 제공
4. 스마트 ICT 융합공학과에 관한 질문(커리큘럼, 교수님): 스마트 ICT 융합공학과 홈페이지

|  |  |
| --- | --- |
| **초기 계획** | **변경 내용** |
| 1. main.jsp 웹페이지 내에 챗봇이 종속되게 구현 | 페이스북 메신저와 연동하여 페이스북 메신저를 통해서 챗봇 이용  - 챗봇 도메인을 페이스북 메신저와 연동 – Invoke API를 facebook messenger에 넘긴다. |
| 2. 전 질문에 대한 오픈형 답변 | 건국대학교로 오는 길을 물어보는 질문만이 폐쇄형 답변 형식을 가짐. |
| [변동 원인]   1. REST API 기반의 End-point를 확장하지 못함. (능력 부족)   2.문제점: 현재 개발자 “장지영”의 계정만이 챗봇 이용이 가능하다.   1. 원인: 페이스북의 정책 – 광고 또는 홍보를 제한하기 위해 앱 검수와 개인 인증 요구, 요구사항을 만족했음에도 페이스북의 허가가 나오지 않았다. 2. 앱 검수: 스크린 캐스트를 통한 권한 이용 동영상 제작, 개인정보처리방침작성 및 URL 생성 3. 개인 인증: “계약서를 받고 검토하고 제출할 수 있는 이메일 주소를 제공하고, 승인된 신분증 또는 문서의 이미지 사본을 업로드해야 합니다. 개인으로 인증하면 회원님 요청의 특정 권한 또는 기능을 이용할 수 없거나 특정 권한 또는 기능이 앱 설치 프로그램별로 제한됩니다. 또한 계약서 1개에 서명해야 합니다.” | |

**3. 실행 매뉴얼**

**1) 실행 영상**

**링크참조.**

**2) 스토리 오브 도식 백배 즐기기!**

**[재학생]**

1, 회원가입하기

1. 회원가입 버튼을 클릭하세요
2. 본인의 학번, 아이디, 비밀번호, 이메일을 입력하고 ‘회원가입’ 버튼을 클릭하세요. 이때, 모든 요소를 입력하지 않거나 이메일 같은 경우 조건에 맞는 이메일 주소가 아니면 경고창이 뜹니다.

2, 로그인하기

\*로그인을 해야 메인 페이지로 이동할 수 있어요

1. 회원가입이 되어 있지 않은 경우에 로그인을 시도하면 아이디가 존재하지 않는다는 경고창이 뜹니다. 이 경우 회원가입을 먼저 하세요.
2. 아이디와 비밀번호를 입력하고 로그인을 합니다.

3, 채팅 이용하기

1. 채팅 입력창에 보내고 싶은 말을 보내 보세요
2. 나의 아이디와 내가 보낸 시간도 확인할 수 있답니다. 즐거운 채팅 하세요~

4, 챗봇 이용하기

건국대학교와 스마트 ICT 융합공학과에게 궁금한 것들을 많이 물어보세요~

1. “챗봇을 이용해 보세요” 버튼을 클릭하면 챗봇을 이용할 수 있는 페이스북 메신저가 뜹니다
2. 페이스북에 로그인 하세요
3. 건국대학교 가는 방법, 학교내의 건물과 시설물들 위치, 편입학 관련정보, 우리 과의 교수님들과 커리큘럼 정보 모두모두 알 수 있습니다.

\*질문 예시: “학생회관이 어디야?”, “건대 가려면 몇 번 버스 타고 가야해?”, “모집인원 알려줘”, “과사무실 전화번호 좀” 등등

5, 메인 페이지 둘러보기

1. 링크: 필요한 사이트에 해당하는 아코디언을 클릭하면 해당 사이트의 간단한 설명이 나오고, 그걸 클릭하면 해당 사이트로 이동합니다

\*건국대학교 홈페이지, 이 캠퍼스, 상허 기념도서관, 스마트 ICT 융합공학과 홈페이지, 스마트 ICT 융합공학과 공식 페이스북 페이지

1. 사진첩: 스마트 ICT 융합공학과 학생들의 사진을 볼 수 있습니다. 사진은 슬라이드 쇼로 넘어가는데 원하는 사진에 마우스를 올리면 슬라이드 쇼가 멈추니 마음껏 보고 귀여워해 주세요!
2. 스케쥴 노트: 이제는 나의 스케쥴을 SOD에서 관리해 보세요! 제목, 날짜, 내용을 입력하고 확인 버튼을 클릭하면 내용이 스케쥴 노트에 등록이 됩니다!

6, 게시판

자유게시판:자유롭게 너의 이야기 나의 이야기 해보세요~ 무엇이든 올려보아요! 서로 댓글도 달아보고 따봉도 날려보아요~

상담게시판: 나 혼자 끙끙 앓지 말고 우리 도식이들에게 털어놓아요~ 아무도 모르게 나의 고민을 올리면 분명 우리 함꼐 해결할 수 있을거에요

사진첩: 남는건 사진뿐!! 우리들의 추억을 사진첩에 간직해 보아요~

장터게시판: 아껴쓰고 나눠쓰고 바꿔쓰고 다시쓰는 '아나바다 도식이 장터!' 믿을 수 있는 사람들과의 win-win 거래! 다 쓴 전공&교양서적, 안 입는 옷을 장터에서 사고 팔아보아요!

**[수험생]**

수험생 친구들은 아쉽게 챗봇만 이용할 수 있어요! 꼭 입학해서 SOD 많이 이용해 주세요~