



SHIBA!



목차

01 프로젝트 개요

02 순서도 및 클래스

03 기간 별 수행 일지

04 기타

01

프로젝트 개요





주 제

애견인을 위한 SNS 웹 어플리케이션



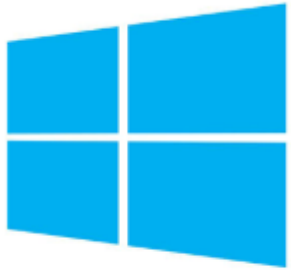
기획 의도



- ◆ 사람의 경우, 인스타, 페이스북 등의 SNS가 대표적이지만 동물의 경우, 대표하는 커뮤니티가 없음
- ◆ SNS에 콘텐츠를 자랑하고 싶어하는 사람들의 심리를 이용하여 2000만 애견 인구를 위한 SNS 기획



구현 언어 및 개발 환경



Windows 10

OS

Windows 10



구현 언어

JAVA 8



디자인

Scene Builder

02

순서도 및 클래스





기능 설계

로그인

로그인

회원가입

메인

나의 SHIBA

커뮤니티

채팅

SNS

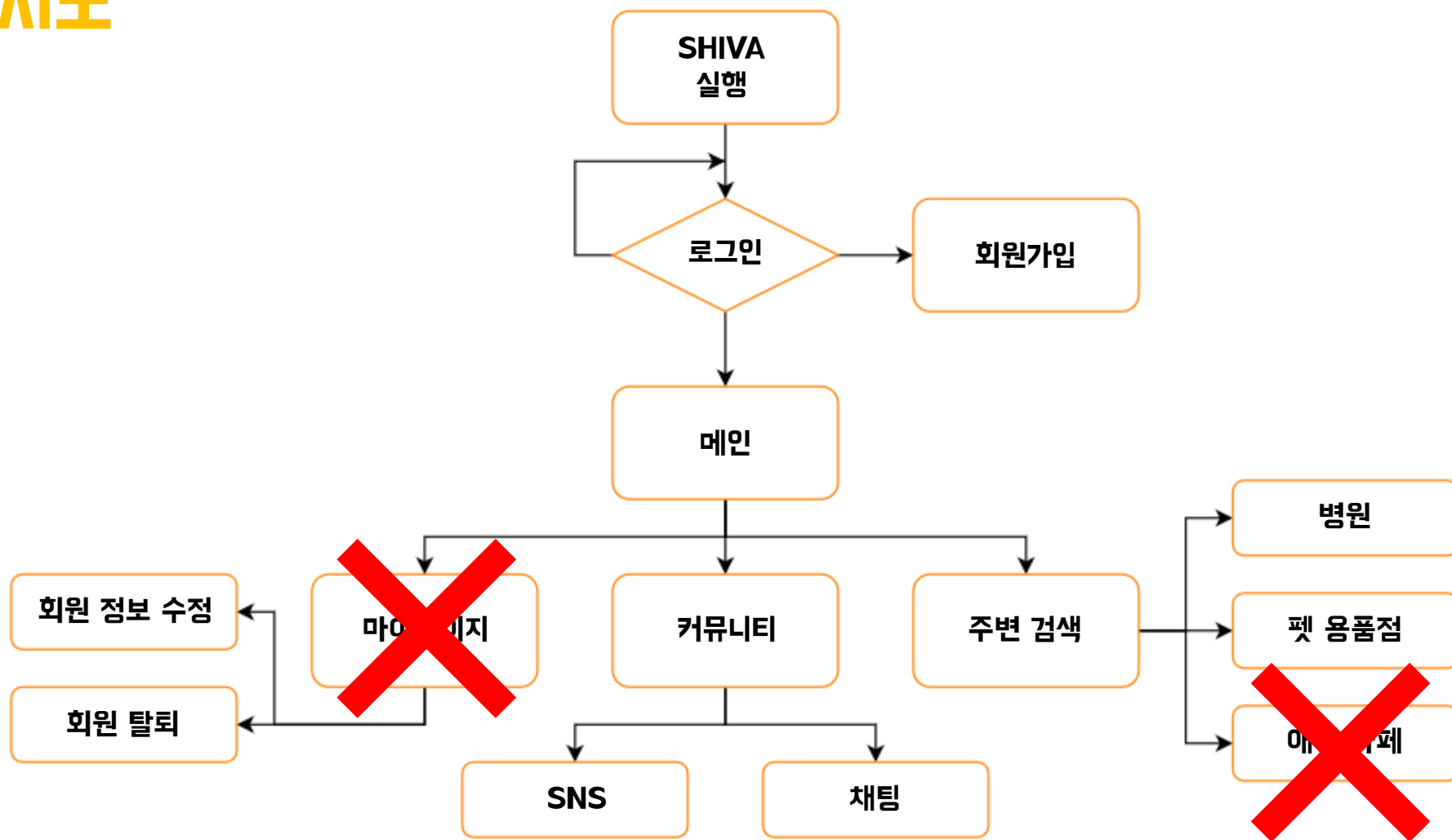
주변 검색

병원

펫 용품점

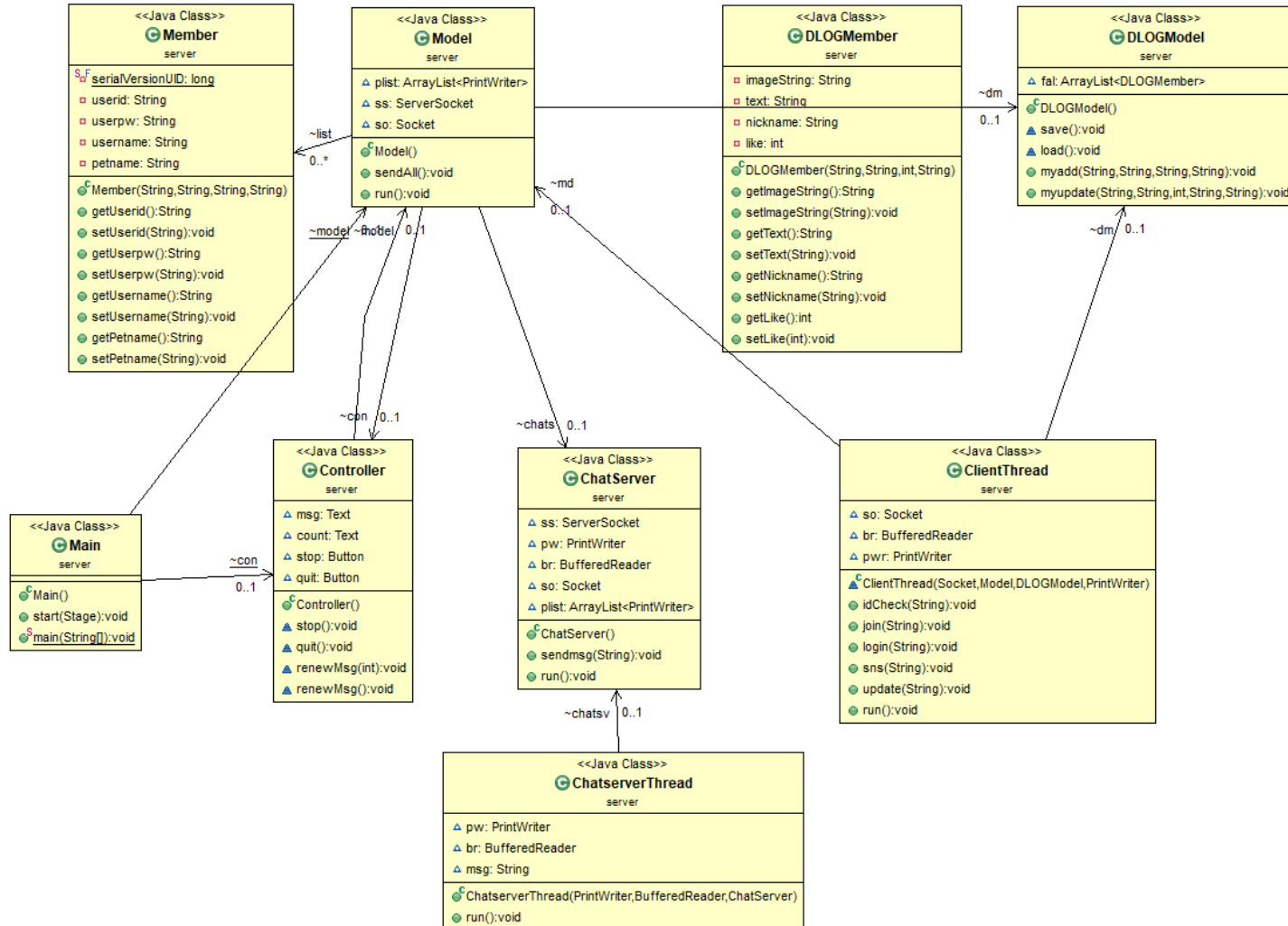


순서도



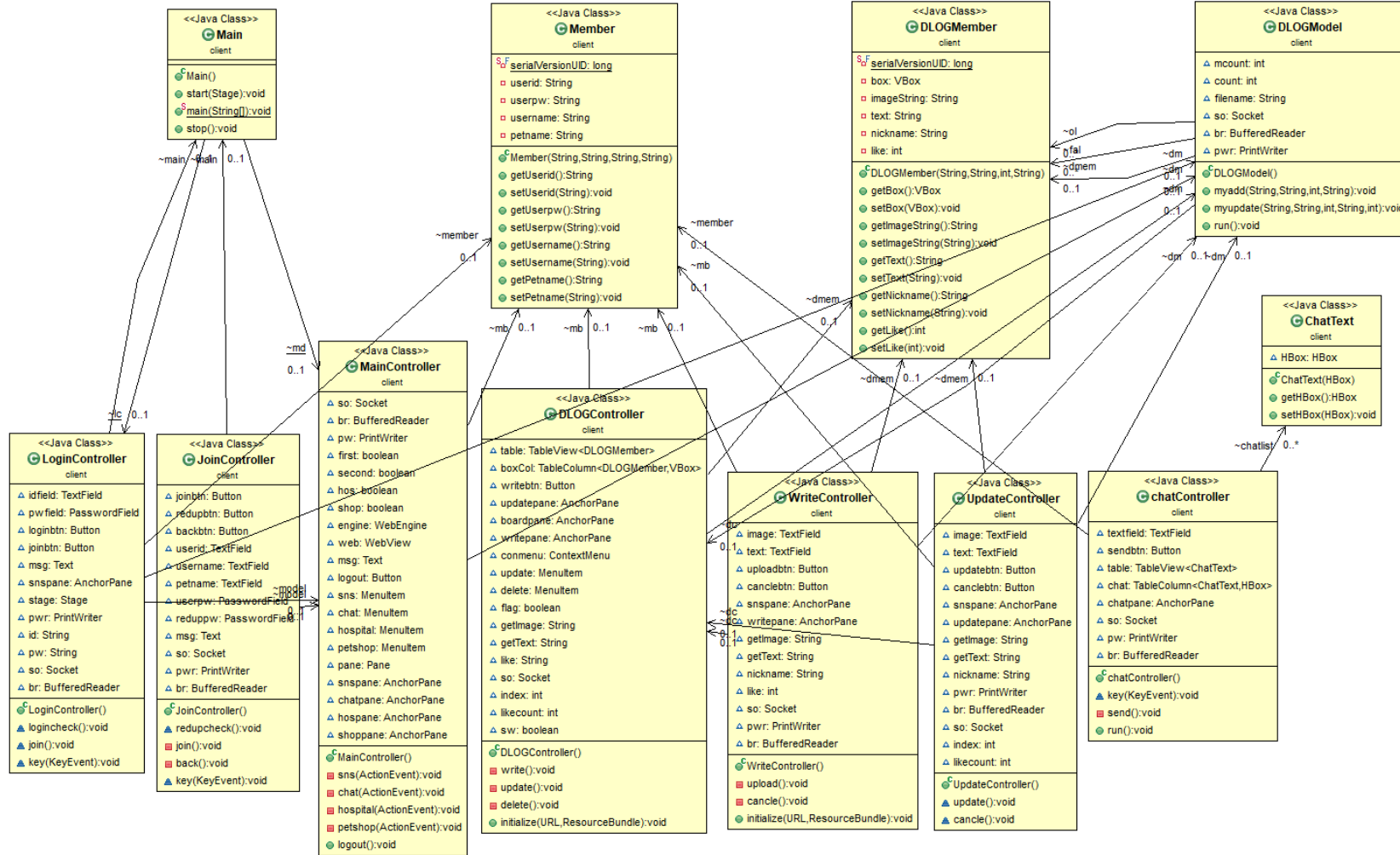


클래스 구상 (Server)





클래스 구상 (Client)



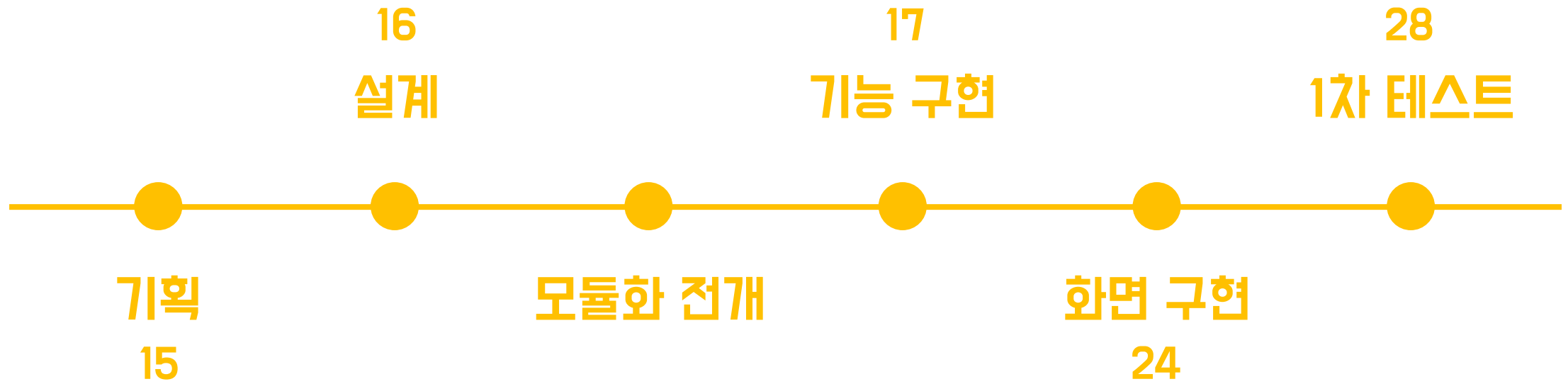
03

기간 별 수행 일지





기간 별 수행 계획

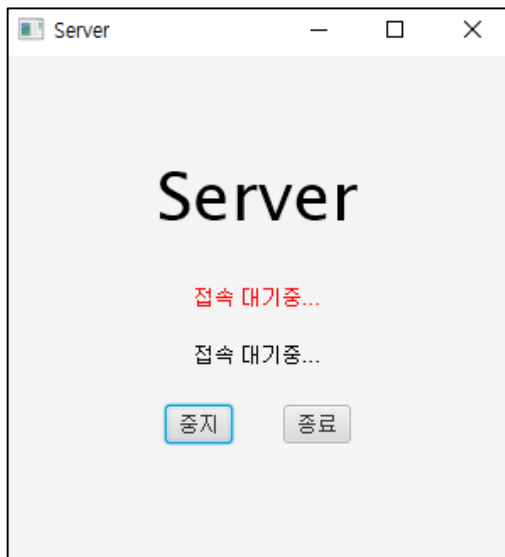




기간 별 수행 일지

Day 2 ~ Day6
(10.16 ~ 10.20)

회원가입 & 로그인



Server 구현

Server Model :

1. 서버 소켓 열기
(DLOG 포트 : 7777.
채팅 포트 : 7979)
2. 클라이언트로부터 받아오는 정보가
ID 중복 확인, 회원가입, 로그인,
SNS 내용인지 판별
(Thread, 구분자 ★와 . 활용)

```
@Override
public void run() {
    try {
        while(true) {
            br = new BufferedReader(new InputStreamReader(so.getInputStream()));
            String first = br.readLine();
            if(first == null) break;
            String[] secon = first.split("#");
            if (secon[0].equals("id")) idCheck(secon[1]);
            if (secon[0].equals("회원가입")) join(secon[1]);
            if (secon[0].equals("로그인")) login(secon[1]);
            if (secon[0].equals("sns")) {sns(secon[1]);}
            if(secon[0].equals("update")) {update(secon[1]);}
        }
    }
}
```



기간 별 수행 일지

Day 2 ~ Day6
(10.16 ~ 10.20)

회원가입 & 로그인

로그인

SHIVA

로그인

ID

PW

로그인

ID 찾기 PW 찾기 회원가입

Client 구현

LoginController:

1. 소켓 연결 (포트 번호 : 7777)
2. 서버로 보낼 **로그인** 정보를
구분자 '★'와 '.', '를 이용해 구분
3. 장면 전환으로
회원가입, 메인 화면으로 넘어감
(소켓, 모델, 멤버 전달)

ex) 로그인 ★ id, pw
sep [0] [1]
memer [0] [1]

```
id = idfield.getText();  
pw = pwfield.getText();  
  
so = new Socket("127.0.0.1", 7777);  
pwr = new PrintWriter(so.getOutputStream());  
br = new BufferedReader(new InputStreamReader(so.getInputStream()));
```

```
pwr.println("로그인# " + id + "," + pw);  
pwr.flush();
```

```
// 메인화면  
Stage join = (Stage) loginbtn.getScene().getWindow();  
FXMLLoader loader = new FXMLLoader(getClass().getResource("main.fxml"));  
Parent root = loader.load();  
join.setScene(new Scene(root));  
MainController mc = loader.getController();  
mc.so = this.so;  
mc.br = this.br;  
mc.pw = this.pw;  
mc.dm = dm;  
mc.mb = this.member;
```



기간 별 수행 일지

Day 2 ~ Day6
(10.16 ~ 10.20)

회원가입 & 로그인

회원가입

SHIVA

회원가입

아이디

중복확인

이름

비밀번호

비밀번호 확인

애완견 이름

회원 가입하기

back

Client 구현

JoinController:

1. 가입 버튼 활성화 조건

- (1) **소켓** 연결 후 ID 중복 검사
- (2) 비밀번호 확인 검사
- (3) 모든 필드 채웠는지 검사

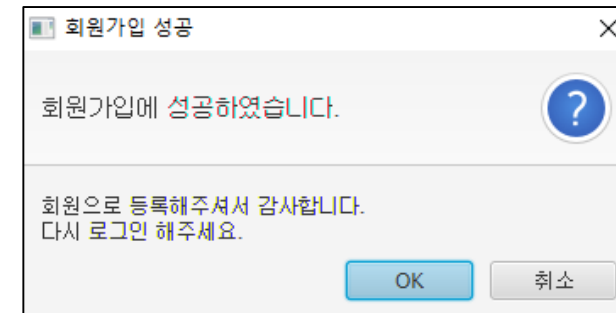
2. 회원가입 성공 시, Alert를 이용해 알림

3. 이 역시 구분자 사용하여 서버로 전송

ex) 회원가입 ★ id. pw. name. petname
sep [0] [1]
memer [0] [1] [2] [3]

```
}else {  
    so = new Socket("127.0.0.1",7777);  
    pwr = new PrintWriter(so.getOutputStream());  
    br = new BufferedReader(new InputStreamReader(so.getInputStream()));  
  
    pwr.println("id#"+userid.getText());  
    pwr.flush();  
}
```

```
Alert alert = new Alert(AlertType.CONFIRMATION);  
alert.setTitle("회원가입 성공");  
alert.setHeaderText("회원가입에 성공하였습니다.");  
alert.setContentText("회원으로 등록해주셔서 감사합니다.\n다시 로그인 해주세요.");  
alert.showAndWait();
```





기간 별 수행 일지

Day 7 ~ Day 10 SNS (DLOG) 구현
(10.21 ~ 10.24)



Client 구현

DLOGController:

1. TableView 활용

(1) Initialize에 table 목록 더블 클릭 시.
좋아요. 좋아요 해지 기능 구현

(2) Initialize에 table의 각 셀에 VBox로
삽입함을 정의

2. 글쓰기 버튼 구현

(WriteController로
소켓, 멤버, DLOG모델 전달)

3. 우클릭 시 ContextMenu를 활용한

수정, 삭제 기능 구현

(선택된 Table의 인덱스 전달)

```
@Override
public void initialize(URL location, ResourceBundle resources) {
    table.setOnMouseClicked(e->{
        if(e.getClickCount()==2 && sw==false) {
            index = table.getSelectionModel().getSelectedIndex();
            this.likecount = dm.ol.get(index).getLike();
            likecount += 1;
            sw = true;

            dm.myupdate(dm.ol.get(index).getNickname(), dm.ol.get(index).getImageString(),
                likecount, dm.ol.get(index).getText(), index);

        }else if(e.getClickCount()==2 && sw==true) {
            index = table.getSelectionModel().getSelectedIndex();
            this.likecount = dm.ol.get(index).getLike();
            likecount -= 1;
            sw = false;

            dm.myupdate(dm.ol.get(index).getNickname(), dm.ol.get(index).getImageString(),
                likecount, dm.ol.get(index).getText(), index);

        }
    });
    boxCol.setCellValueFactory(new PropertyValueFactory<DLOGMember, VBox>("box"));
```



기간 별 수행 일지

Day 7 ~ Day 10 SNS (DLOG) 구현 - 글쓰기
(10.21 ~ 10.24)



Client 구현

WriteController:

1. 사진의 URL과 글 내용을 구분자 SNS★와 `로 서버 전송 (String으로 저장 후 전송)
2. 서버로부터 받은 자신 + 다른 클라이언트의 SNS 내용을 DLOG Model의 ArrayList에 저장
3. ArrayList를 ObserverList에 저장 후 Table에 setItems

```
@FXML private void upload() throws FileNotFoundException{
    System.out.println("(upload)socket : " + so);

    getImage = image.getText();
    getText = text.getText();
    nickname = "[ " + mb.getPetname() + " ] 이의 일상";

    try {
        pwr = new PrintWriter(so.getOutputStream());
        pwr.println("sns#" + nickname + "," + getImage + "," + like + "," + getText);
        pwr.flush();
    }
```

```
String result = br.readLine().trim();

String[] re = result.split("#");
if(re[0].equals("sns")) {
    String[] initfal = re[1].split(",");
    int count = Integer.parseInt(initfal[2]);
    dm.ol.clear();
    dm.myadd(initfal[0], initfal[1], count, initfal[3]);
}

dc.table.setItems(dm.ol);
```




기간 별 수행 일지

Day 7 ~ Day 10
(10.21 ~ 10.24)

SNS (DLOG) 구현 - 글쓰기



Server 구현

Model:

1. 클라이언트로부터 받은 사진의 URL과 글 내용을 DLOGModel에 저장
2. File에도 저장
3. 모든 클라이언트에게 SNS 내용 전송

```
public void sns(String vbox) {  
    String[] sns vbox = vbox.split(".");  
    dm.myadd(sns vbox[0], sns vbox[1], sns vbox[2], sns vbox[3]);  
    pwr.println("sns" + sns vbox[0] + "," + sns vbox[1] + "," +  
        sns vbox[2] + "," + sns vbox[3]);  
    pwr.flush();  
}
```

```
void save() {  
    File f = new File("sns.txt");  
    FileOutputStream fos = null;  
    ObjectOutputStream oos = null;  
    try {  
        fos = new FileOutputStream(f);  
        oos = new ObjectOutputStream(fos);  
        oos.writeObject(fal);  
        oos.flush();  
    }
```

```
void load() {  
    File f = new File("sns.txt");  
    FileInputStream fis = null;  
    ObjectInputStream ois = null;  
    try {  
        fis = new FileInputStream(f);  
        ois = new ObjectInputStream(fis);  
        fal = (ArrayList<DLOGMember>)ois.readObject();  
    }
```



기간 별 수행 일지

Day 7 ~ Day 10 SNS (DLOG) 구현 - 수정
(10.21 ~ 10.24)



Client 구현

UpdateController:

1. 우클릭 시 수정, 삭제 기능 구현
2. 선택된 테이블의 인덱스를 DLOG Model에 전달
3. 전달받은 인덱스에 해당하는 ArrayList 내용 수정
4. 서버에도 수정된 내용 전달
5. 서버의 DLOG Model의 해당 인덱스 내용도 수정

```
@FXML private void update() {  
    try {  
        index = table.getSelectionModel().getSelectedIndex();  
  
        FXMLLoader loader = new FXMLLoader(getClass().getResource("update.fxml"));  
        updatepane = loader.load();  
        boardpane.getChildren().clear();  
        boardpane.getChildren().add(updatepane);  
        UpdateController uc = loader.getController();  
        uc.dc = this;  
        uc.dm = this.dm;  
        uc.mb = this.mb;  
        uc.so = this.so;  
        uc.index = this.index;  
        uc.likecount = this.likecount;
```

```
public void myupdate(String nickname, String imageString, int like, String text, int index) {  
    fal.set(index, new DLOGMember(nickname, imageString, like, text));  
    ol.set(index, new DLOGMember(nickname, imageString, like, text));  
}
```



기간 별 수행 일지

Day 7 ~ Day 10 SNS (DLOG) 구현 - 수정
(10.21 ~ 10.24)



〈 수정 전 〉



〈 수정 후 〉



기간 별 수행 일지

Day 7 ~ Day 10
(10.21 ~ 10.24)

SNS (DLOG) 구현 - 삭제

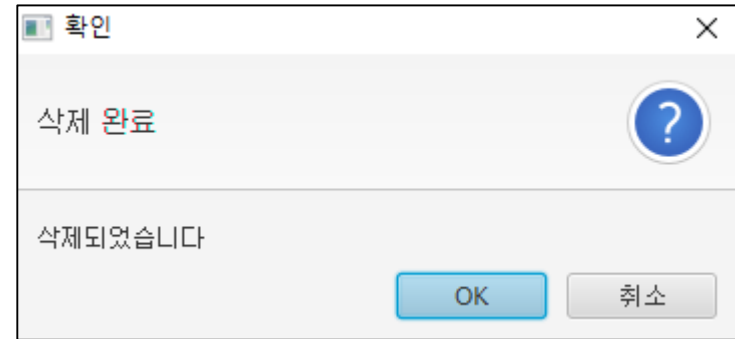


Client 구현

DLOGController:

1. 수정과 마찬가지로 선택된 인덱스 저장
2. ArrayList와 Observer에서 해당 인덱스 내용 삭제
3. Alert로 알림

```
@FXML private void delete() {  
    int index = table.getSelectionModel().getSelectedIndex();  
    dm.fal.remove(index);  
    dm.ol.remove(index);  
  
    Alert al = new Alert(AlertType.CONFIRMATION);  
    al.setHeaderText("삭제 완료");  
    al.setContentText("삭제되었습니다");  
    al.showAndWait();  
}
```





기간 별 수행 일지

Day 7 ~ Day 10 SNS (DLOG) 구현 - 좋아요
(10.21 ~ 10.24)



Client 구현

DLOGController:

1. '좋아요' 조건

(1) 본인 게시글 좋아요 가능

(2) 더블 클릭 1번 : 좋아요

더블 클릭 2번 : 좋아요 해지



기간 별 수행 일지

Day 11
(10.25)

주변검색



Client 구현

MainController:

1. WebView 활용
2. APP의 작은 사이즈를 고려해
Mobile 사이즈로 웹뷰 구현

```
@FXML private void hospital(ActionEvent e) throws IOException {  
    hos = e.isConsumed();  
    if(hos == false) {  
        FXMLLoader loader = new FXMLLoader(getClass().getResource("hospital.fxml"));  
        web = loader.load();  
        engine = web.getEngine();  
        pane.getChildren().clear();  
        pane.getChildren().add(web);  
        engine.load("https://m.map.naver.com/search2/search.nhn?query=%EB%8F%99%EB%AC");  
        hos = true;  
    }  
}
```



기간 별 수행 일지

Day 11
(10.25)

주변검색



〈 동물 병원〉



〈 펫 용품점〉



기간 별 수행 일지

Day 12 ~ Day 13
(10.26 ~ 10.27) 채팅



Client 구현

ChatController:

1. 포트 번호 : 7979
2. 소켓 연결
3. TableView를 HBOX로 구성
 - (1) 입장은 파란색, 가운데 정렬
 - (2) '나'가 말하는 채팅은 오렌지 색 바탕

```
@Override
public void run() {
    try {
        so = new Socket("127.0.0.1", 7979);
        pw = new PrintWriter(so.getOutputStream());
        br = new BufferedReader(new InputStreamReader(so.getInputStream()));

        pw.println "[" + member.getPetname() + "] 님이 입장하였습니다.";
        pw.flush();
    }
}
```

```
while(true) {
    HBox box = new HBox();
    String text = br.readLine().trim();
    System.out.println("클라가 받은 버퍼 : " + text);
    if(text.contains(":")) {
        String[] name = text.split(" :");
        Text tx2 = new Text(text);
        box.getChildren().add(tx2);
        chatlist.add(new ChatText(box));
        if(name[0].equals(member.getPetname())) {
            box.setStyle("-fx-background-color: #f89b00;");
        }
    }
    else if(text.contains("입장")) {
        HBox box1 = new HBox();
        Text tx = new Text(text);
        box1.setAlignment(Pos.CENTER);
        tx.setFill(Color.BLUE);
        box1.getChildren().add(tx);
        chatlist.add(new ChatText(box1));
    }
}
```



기간 별 수행 일지

Day 12 ~ Day 13
(10.26 ~ 10.27) 채팅

메인 페이지

로그아웃

SHIBA

시바 님, 안녕하세요 !

커뮤니티 ▾ 주변검색 ▾

실시간 채팅

[시바] 님이 입장하였습니다.

[백구] 님이 입장하였습니다.

시바 : 오늘 7시에 산책시키러 가실분

백구 : 저요!

시바 : 배변 봉투랑 목줄 꼭 챙겨오세요

백구 : 제 사랑도 챙겨가겠습니다

시바 : 어머

백구 : 개팅으로 맺은 인연 오래가봅시다

< >

전송

메인 페이지

로그아웃

SHIBA

백구 님, 안녕하세요 !

커뮤니티 ▾ 주변검색 ▾

실시간 채팅

[백구] 님이 입장하였습니다.

시바 : 오늘 7시에 산책시키러 가실분

백구 : 저요!

시바 : 배변 봉투랑 목줄 꼭 챙겨오세요

백구 : 제 사랑도 챙겨가겠습니다

시바 : 어머

백구 : 개팅으로 맺은 인연 오래가봅시다

< >

전송

04

7월





개선 사항

- ◆ 오라클, AWS 등의 DB 구축
- ◆ 디자인적 요소
- ◆ 마이페이지, ID, 비밀번호 찾기 미구현 (시간 부족)
- ◆ 애견카페 미구현 (동일한 코드이므로 생략)
- ◆ SNS 삭제 시 업데이트 미구현 (시간 부족)



기대 효과

- ◆ 고양이, 새 등의 다양한 동물들로 확대 가능



참조 사람

◆ 김성곤, 김회준, 김수철



THANK YOU!