



Office Communication solution

참여자 :

이동열 윤빈 서진호 김재성 김범진

- 01** 개요
- 02** 클래스 및 DB구성
- 03** ERD
- 04** 프로그램 시연
- 05** QnA

배운 것,
그리고
할 수
있는 것.



개발 환경

- 언어 : Java ver8
- IDE : Eclipse
- DBMS : Oracle
- OS : Window10
- 라이브러리 :
 - Javax.mail
 - Jdbc.Oracle
 - Jtattoo(Swing Look and feel Util)

목표



사무 업무용
커뮤니케이션
프로그램 잔디

+



?



?



?

- 디자인 클로닝

- ◆ SWING을 이용 패널간의 이동을 통한 기능 스위칭 구현.

- DB관리

- ◆ 오라클 연동을 통한 회원, 게시판, 채팅등 데이터 관리

- 채팅

- ◆ 서버소켓을 활용하여 멀티 클라이언트 채팅기능 구현

- DB의 가시화 및 코드 브랜치 관리

- ◆ ?????

개발 일정

아이디어 수립

~ 11/13

레이아웃 작업

~ 11/20

기능 구현

~ 11/26

리팩터링 및 기능 합치기

~ 11/29

최종 구현 및 문서작업

~ 11/30

VO모델

-회원 관리

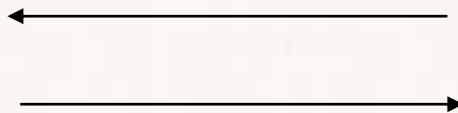
- registMemberVO

-게시판 관리

- boardVO
- boardGroupVO

-채팅관리

- ChatVO
- ChatMemberVO
- MessageVO



DB

LoginPage

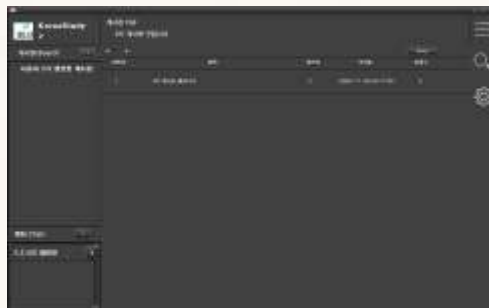
로그인

ID

PASSWORD

로그인

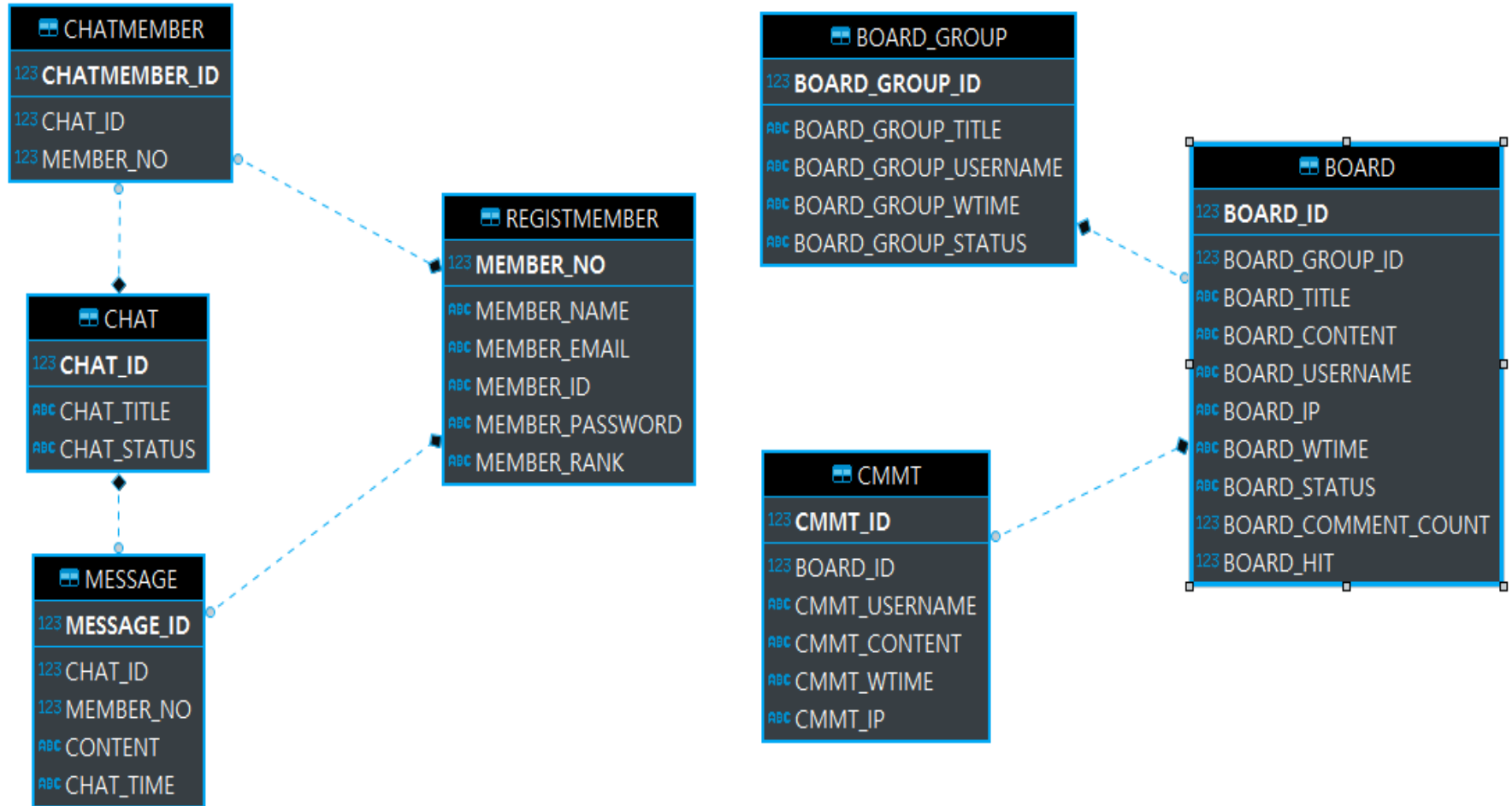
MainApp



Util

- Method Library
- DBManager

ERD



게시판 개발 주안점(패널 추가)

// 글쓰기 등록 시 board content에 패널추가.

```
public void addBoardPanel() {  
    boardMain.boardPanel_center_content.removeAll();  
    panelList.removeAll(panelList);  
    labelListNum.removeAll(labelListNum);  
    labelListTitle.removeAll(labelListTitle);  
    labelListUser.removeAll(labelListUser);  
    labelListTime.removeAll(labelListTime);  
    labelListHit.removeAll(labelListHit);  
}
```

```
for (int i = 0; i < boardList.size(); i++) {  
    // 해당 num값을 받아온다.. 이벤트 처리를 위하여..  
    int num = i;  
}
```

// 패널 생성

```
panelList.add(new JPanel());  
panelList.get(i).setBorder(new MatteBorder(0, 0, 1, 0, (Color) new Color(  
panelList.get(i).setBackground(Color.DARK_GRAY);  
panelList.get(i).setLayout(null);  
boardMain.boardPanel_center_content.add(panelList.get(i));  
panelList.get(i).setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CUR
```

// 글 번호 라벨

```
labelListNum.add(new JLabel(Integer.toString((ALLCOUNT - ((boardMain.p  
labelListNum.get(i).setForeground(Color.WHITE);  
labelListNum.get(i).setFont(new Font("HY전고딕", Font.PLAIN, 12));  
labelListNum.get(i).setBounds(38, 20, 57, 15);  
panelList.get(i).add(labelListNum.get(i));
```

// 글 제목 라벨

```
labelListTitle.add(new JLabel(boardList.get(i).getBoard_title()));  
labelListTitle.get(i).setForeground(Color.WHITE);  
labelListTitle.get(i).setFont(new Font("HY전고딕", Font.PLAIN, 12));  
labelListTitle.get(i).setBounds(135, 20, 400, 15);  
panelList.get(i).add(labelListTitle.get(i));
```

// 글 작성자 라벨

```
labelListUser.add(new JLabel(boardList.get(i).getBoard_username()));  
labelListUser.get(i).setForeground(Color.WHITE);  
labelListUser.get(i).setFont(new Font("HY전고딕", Font.PLAIN, 12));  
labelListUser.get(i).setBounds(540, 20, 57, 15);  
panelList.get(i).add(labelListUser.get(i));
```

// 글 작성시간 라벨

```
labelListTime.add(new JLabel(boardList.get(i).getBoard_wtime()));  
labelListTime.get(i).setForeground(Color.WHITE);  
labelListTime.get(i).setFont(new Font("HY전고딕", Font.PLAIN, 12));  
labelListTime.get(i).setBounds(620, 20, 150, 15);  
panelList.get(i).add(labelListTime.get(i));
```

// 글 조회수 라벨

```
labelListHit.add(new JLabel(Integer.toString(boardList.get(i).getBoard_hit(  
labelListHit.get(i).setForeground(Color.WHITE);  
labelListHit.get(i).setFont(new Font("HY전고딕", Font.PLAIN, 12));  
labelListHit.get(i).setBounds(827, 20, 57, 15);  
panelList.get(i).add(labelListHit.get(i));
```

// 패널 클릭 이벤트

```
panelList.get(i).addMouseListener(new MouseAdapter() {  
    public void mouseEntered(MouseEvent e) {  
        panelList.get(num).setBackground(Color.black);  
    }  
}
```

```
public void mouseExited(MouseEvent e) {  
    panelList.get(num).setBackground(Color.DARK_GRAY);  
}
```

// 패널 클릭 이벤트

```
panelList.get(i).addMouseListener(new MouseAdapter() {  
    public void mouseEntered(MouseEvent e) {  
        panelList.get(num).setBackground(Color.black);  
    }  
  
    public void mouseExited(MouseEvent e) {  
        panelList.get(num).setBackground(Color.DARK_GRAY);  
    }  
}
```

public void mouseReleased(MouseEvent e) {

```
    writeBtnEvent();  
    boardMain.mainApp.headerIssue.selectBoard();  
    boardMain.scrollWrite.setBounds(112, 88, 686, 311);  
    boardMain.delBtn.setBounds(260, 420, 86, 23);  
    boardMain.prevBtn.setBounds(410, 420, 86, 23);  
    boardMain.updBtn.setBounds(560, 420, 86, 23);  
    // 수정, 삭제등에 사용할 board_id를 담아둔다!  
    boardMain.board_id = boardList.get(num).getBoard_id();  
    // 조회수 올리기 메서드.
```

updHit();

// 댓글 조회 메서드

selectCmmt();

boardMain.title.setText(boardList.get(num).getBoard_title());

boardMain.content.setText(boardList.get(num).getBoard_content());

// placeholder 기능으로 인해 회색으로 글씨가 보여 하얀색으로 초기화 시킴..

boardMain.title.setForeground(Color.white);

boardMain.content.setForeground(Color.white);

// 삭제 수정 버튼을 보여줄지 여부..

```
if(boardList.get(num).getBoard_username().equals(boardMain.mainApp.getRegistMemberVO().getMember_id())) {  
    boardMain.delBtn.setVisible(true);  
    boardMain.updBtn.setVisible(true);  
}
```

boardMain.registBtn.setVisible(false);

// 클릭 시 검은 배경이 남는 현상 발생으로 색깔 변경..

```
panelList.get(num).setBackground(Color.DARK_GRAY);  
}
```


게시판 개발 주안점(페이지 추가)

```
// board의 갯수를 조회한다.
public void countRowBoard() {
    String sql = "select board_id from board where board_status = 0 and board_group_id = ?";
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    try {
        pstmt = boardMain.con.prepareStatement(sql, ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        pstmt.setInt(1, boardMain.board_group_id);
        rs = pstmt.executeQuery();
        rs.last();
        ALLCOUNT = rs.getRow();
        PAGECOUNT = (rs.getRow() + (boardMain.GRIDCOUNT - 1)) / boardMain.GRIDCOUNT;
        // page 갯수 계산을 위해 넣어준다.. 10개 이상이될때 분리 시키는 작업..
        PAGECOUNTSPLIT = PAGECOUNT;
        if(PAGECOUNT < 10) {
            PAGECOUNTSPLITNUM = PAGECOUNT;
            if(PAGECOUNT == 0) {
                PAGECOUNTSPLITNUM = 1;
            }
        } else {
            PAGECOUNTSPLITNUM = 10;
        }
        rs.beforeFirst();

        addPage();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        boardMain.dbCon.close(pstmt, rs);
    }
}
```

```
// page수가 많아지면 x값을 증가시켜 이동시키기 위해..
x += 20;
boardMain.boardPanel_north.add(boardMain.labelPageIndex.get(i));
boardMain.labelPageIndex.get(i).setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
// page쪽수 라벨을 누르면 작동.. 페이지 이동
boardMain.labelPageIndex.get(i).addMouseListener(new MouseAdapter() {
    public void mouseReleased(MouseEvent e) {
        boardMain.labelPageIndex.get(boardMain.pagenum).setForeground(Color.white);
        boardMain.pagenum = num;
        boardMain.labelPageIndex.get(num).setForeground(Color.black);
        pageSelect();
    }
});
}

boardMain.labelPageNext = new JLabel("▶");
boardMain.labelPageNext.setFont(new Font("HY견고딕", Font.PLAIN, 12));
boardMain.labelPageNext.setForeground(Color.WHITE);
boardMain.labelPageNext.setBounds(x + 10, 7, 16, 15);
boardMain.labelPageNext.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
boardMain.boardPanel_north.add(boardMain.labelPageNext);

boardMain.labelPagePrev.addMouseListener(new MouseAdapter() {
    public void mouseReleased(MouseEvent e) {
        // 최소 페이지 수를 넘기면 안넘어 가도록 조정
        if (boardMain.pagenum > 0) {
            if (boardMain.pagenum % 10 == 0) {
                boardMain.pageForINum = (boardMain.pageForINum - 10);
                delPageLabelPrev();
                if (PAGECOUNTSPLITNUM == PAGECOUNT) {
                    PAGECOUNTSPLITNUM -= (boardMain.pageForINum + 10);
                } else {
                    PAGECOUNTSPLITNUM -= 10;
                }
                addPage();
            }
            if (boardMain.labelPageIndex.size() != boardMain.pagenum) {
                boardMain.labelPageIndex.get(boardMain.pagenum).setForeground(Color.white);
            }
            boardMain.pagenum = boardMain.pagenum - 1;
            boardMain.labelPageIndex.get(boardMain.pagenum).setForeground(Color.black);
            pageSelect();
        }
    }
});

boardMain.labelPageNext.addMouseListener(new MouseAdapter() {
    public void mouseReleased(MouseEvent e) {
        // 최대 페이지 수를 넘기면 안넘어 가도록 조정
        if (PAGECOUNT - 1 > boardMain.pagenum) {
            if (boardMain.pagenum == boardMain.labelPageIndex.size() - 1) {
                boardMain.pageForINum = (boardMain.pageForINum + 10);
                delPageLabelNext();
                PAGECOUNTSPLITNUM += 10;
                if (PAGECOUNTSPLITNUM > PAGECOUNT) {
                    PAGECOUNTSPLITNUM = PAGECOUNT;
                }
                addPage();
            }
            boardMain.labelPageIndex.get(boardMain.pagenum).setForeground(Color.white);
            boardMain.pagenum = boardMain.pagenum + 1;
            boardMain.labelPageIndex.get(boardMain.pagenum).setForeground(Color.black);
            pageSelect();
        }
    }
});
}
```

채팅 개발 주안점

줄 바꿈

```
String msg = chatTextArea.getText();
chatTextArea.getInputContext();
chatTextArea.getLineCount();
String lineChangedText = chat_lib.seperateTextLine();
mainAppChatSocket.mainAppchatThread.send(lineChangedText);
chatTextArea.setText("");
```

채팅창에 반영될때 명령어를 \n으로 치환



2

노예

11/30 09:33

해당처럼 채팅줄이
바뀝니다.

채팅 개발 주안점

채팅방의 정보를 스트링 형태로 소켓으로 전달

```
//소켓으로 채팅방에 관한 명령문 전달시, 해당 채팅방의 정보를 전달하는 메소드
public void sendClientInfo(int chat_id, int member_no, String member_name ) {
    String clientInfo = chat_id+","+member_no+","+member_name;
    mainApp.mainAppChatSocket.mainAppchatThread.send(clientInfo);
    System.out.println("sendClientInfo 발동 인포는 : "+clientInfo);
}
```

서버에서 받은 정보를 정리하여, 클라이언트의 소켓으로 전달

```
public void send(String msg) {
    try {
        System.out.println("서버트래드에서 send 발동됨");
        for (int i = 0; i < myServerSocket.threadList.size(); i++) {
            MyServerThread myServerThread = myServerSocket.threadList.get(i);
            myServerThread.buffw.write(chat_id + "," + current_time + "," + member_no + "," + current_messageId + "," + current_memberName + "," + msg + "\n");
            System.out.println(chat_id + "," + current_time + "," + member_no + "," + current_messageId + "," + current_memberName + "," + msg + "\n");
            myServerThread.buffw.flush();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

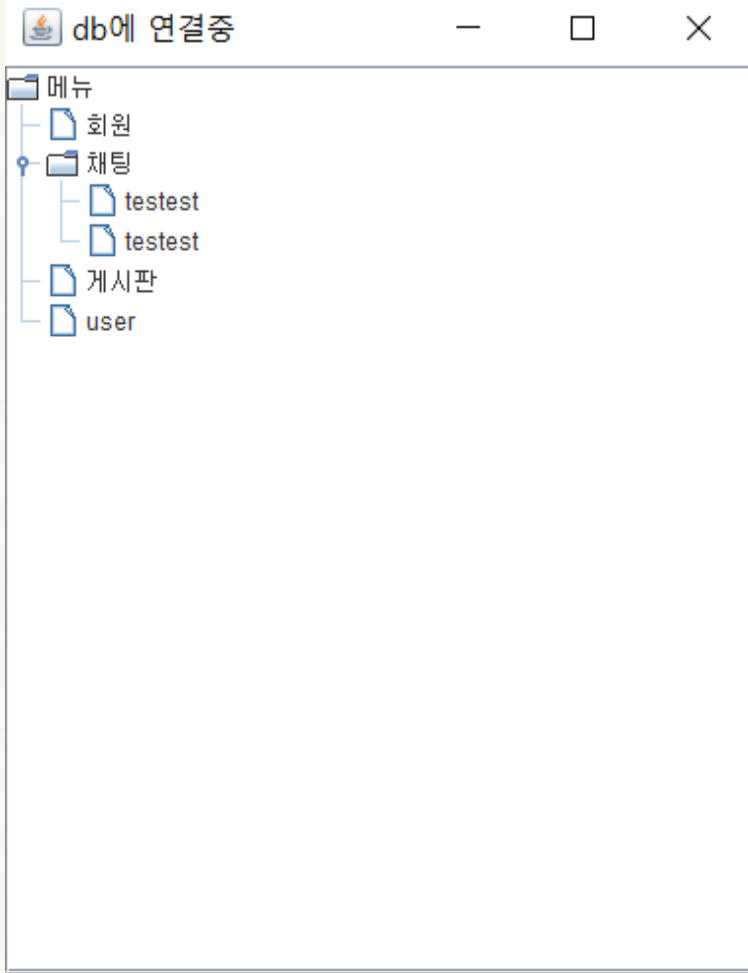
채팅 개발 주안점

```
if(decryptN[0].equals("#newChat:931006")) {
    if (chat_lib.memberComparing(mainApp.messageVO.getChat_id(),
        mainApp.getRegistMemberVO().getMember_no())) {
        chat_lib.refreshChatList(mainApp.con);
    }
} else if(decryptN[0].equals("#oneLeft:931006")) {
    if(mainApp.chatVO.getChat_id() == mainApp.messageVO.getChat_id()) {
        mainApp.chatPanel.p_north_chat_member_panel.removeAll();
        JLabel membernameLabel = new JLabel("참여자가 없는 채팅방입니다.");
        membernameLabel.setFont(new Font("HY전고딕", Font.PLAIN, 20));
        membernameLabel.setForeground(Color.WHITE);
        membernameLabel.setPreferredSize(new Dimension(300, 60));
        mainApp.chatPanel.p_north_chat_member_panel.add(membernameLabel);
        mainApp.chatTextArea.setEnabled(false);
        System.out.println("#oneLeft 종료");
        mainApp.chatTextArea.updateUI();
        mainApp.chatPanel.p_north_chat_member_panel.updateUI();
    }
} else if(decryptN[0].equals("#outFromChat:931006")) {
    if(mainApp.chatVO.getChat_id() == mainApp.messageVO.getChat_id()) {
        chat_lib.setCurrentChatVO(mainApp.chatVO.getChat_title());
        chat_lib.loadChatPanel();

        String m_name = chat_lib.changeMemberNotoName(mainApp.messageVO.getMember_no());
        ChatOutPopPanel chatOutPopPanel = new ChatOutPopPanel(m_name, mainApp.chatOut_pop);
        mainApp.chatOut_pop = mainApp.popupFactory.getPopup(mainApp.frame, chatOutPopPanel, mainApp.frame.getLocationOnScreen().x+450, mainApp.frame.getLocationOnScreen().y+450);
        chatOutPopPanel.close_button.addActionListener((e)->{
            mainApp.chatOut_pop.hide();
            if(mainApp.messageVO.getChat_id() == mainApp.chatVO.getChat_id() && mainApp.messageVO.getMember_no() == mainApp.getRegistMemberVO().getMember_no()) {
                mainApp.p_center_south.setVisible(false);
                mainApp.p_chat_south_center.setPreferredSize(new Dimension(240, mainApp.chatSmallPanels.size()*45));
                mainApp.p_center_center.removeAll();
                mainApp.p_center_center.updateUI();
                mainApp.p_chat_south_center.updateUI();
            }
        });
    }
}
```

클라이언트 소켓에서 클라이언트앱에 이프문을 통해 명령문 마다 다른 기능을 수행

Jtree 주안점



```
tree.addTreeSelectionListener(new TreeSelectionListener() {  
    public void valueChanged(TreeSelectionEvent e) {  
        DefaultMutableTreeNode node1;  
        DefaultMutableTreeNode node2;  
  
        node1 = (DefaultMutableTreeNode) tree.getLastSelectedPathComponent();  
  
        String name = (String) node1.getUserObject();  
        System.out.println(name + "눌림");  
  
        node2 = (DefaultMutableTreeNode) node1.getParent();  
        String parent = (String) node2.getUserObject();  
  
        if (parent.equals("채팅")) {  
            removeAll("chat", "chat_title", name);  
            node2.remove(node2.getIndex(node1));  
        } else if (parent.equals("회원")) {  
            removeAll("RegistMember", "member_id", name);  
            node2.remove(node2.getIndex(node1));  
        } else if (parent.equals("게시판")) {  
            removeAll("RegistMember", "member_id", name);  
            node2.remove(node2.getIndex(node1));  
        } else if (parent.equals("user")) {  
            removeAll("RegistMember", "member_id", name);  
            node2.remove(node2.getIndex(node1));  
        }  
    }  
});
```

```
public void createNode(DefaultMutableTreeNode top) {  
    //넘겨받은 탑 노드에 원하는 하위 노드를 생성하여 자식 노드로 붙이자  
    DefaultMutableTreeNode[] node=new DefaultMutableTreeNode[4];  
    node[0]=new DefaultMutableTreeNode("회원");  
    node[1]=new DefaultMutableTreeNode("채팅");  
    node[2]=new DefaultMutableTreeNode("게시판");  
    node[3]=new DefaultMutableTreeNode("user");  
  
    for(int i=0;i<memList.size();i++) {  
        node[0].add(new DefaultMutableTreeNode(memList.get(i)));  
    }  
  
    for(int i=0;i<chatList.size();i++) {  
        node[1].add(new DefaultMutableTreeNode(chatList.get(i)));  
    }  
  
    for(int i=0;i<editList.size();i++) {  
        node[2].add(new DefaultMutableTreeNode(editList.get(i)));  
    }  
  
    for(int i=0;i<userList.size();i++) {  
        node[3].add(new DefaultMutableTreeNode(userList.get(i)));  
    }  
}
```

시연

느낀점

-이동열

뭐가 어렵고 어디로 접근을
해야하는가 방향이 잡힌 느낌이고,
성취감과 흥미를 느끼고, 아직
어렵지만 나름대로 자신감이 생긴 것
같습니다.

-김재성

스윙으로 디자인을 하면서 너무
시간과 노력이 치중된 감이 있었다.
기능을 우선적으로 진행하는게
맞았으나, 기획구상단계에서 놓친것
같아 아쉬웠다. 하지만 팀원전체가
본 프로젝트를 통해 이루고자했던
목표를 달성할 수 있어 한결으론
뿌듯하다.


-서진호

더 하고 싶은 기능 구현이 많았으나,
개인기량의 미비로 안타깝게 초기 구상한
아이디어를 모두 실현하지 못한 부분이
아쉽다. 하지만 협업을 통해 프로젝트를
진행함에 있어, 정보교류, 워크플로우등
배울점이 많아 아쉽지만, 뜻깊은
경험이었다.

-윤빈

구상에서는 모든게 쉽게 느껴졌지만, 막상
직접 코드를 짜보니 쉽지 않은 부분이
 많다는걸 알게 되었다. 부족한 기량으로
스파게티 코딩이 계속되다보니 기능추가,
버그수정 시에 산출물에비해 쏟아지는
에너지와 시간이 너무 많았었다 실현하진
못했지만 앞으로 어떠한 방향으로
진행할지에대해 감이 잡힌것 같고 덕분에
자신감을 잃고 또 얻는 중요한 시간이
되었던 것 같다.

QnA



김범진

https://github.com/bum89452/js_workspace

김재성

https://github.com/kjsj20/java_workspace/tree/master/TeamProject

윤빈

<https://github.com/vini1006/TeamProject>

이동열

<https://github.com/leedongyoul>

서진호

<https://github.com/seojinho5819>