

니 소식?

– Conceptualization



Student No	22212029
Name	박대영
E-mail	pdy000726@naver.com

[Revision history]

Revision date	Version #	Description	Author
2024/03/22	1.0	First Writing	박대영
2024/04/18	1.1	(좋아요 기능)기능 추가 및 내용 수정	박대영

= Contents =

1. Business purpose	
2. System context diagram	
3. Use case list	
4. Concept of operation	
5. Problem statement	
6. Glossary	
7. References	

1. Business purpose

- Project background, motivation

현대 사회에서는 정보력 싸움이다. 정보력이 없다면 남들보다 앞서 나갈 수 없다. 대학교에서의 정보력 싸움은 취업에 대한 세미나, 교환학생, 자격증 지원, 정부 지원 프로그램 등이 있다. 이러한 소식은 보통 대학 홈페이지에서 쉽게 접할 수 있다. 하지만, 많은 학생은 대학 홈페이지에 들어가지 않아 수많은 정보를 놓치게 된다. 나 또한 놓친 정보가 많았고, 주변에서도 내가 찾은 정보에 대해서 알지 못하는 경우가 다반사다. 이것은 편리함을 추구하는 현대 사회의 폐해라고 생각한다. 그렇다면, 이것을 해결할 수 있는 방법이 무엇일까? 주변에서 쉽게 찾아볼 수 있었다. 직접 찾아보지 않는 가장 큰 이유는 '알림 서비스'이다. 광고 메시지, 알림 메시지 등을 통해 온 정보들은 몇 번의 클릭으로 바로 볼 수 있다. 그리고, 현재 대학 소식 및 학과 소식은 '영남대학교 스마트 출결' 어플로 오긴 하지만, 일부만 올 뿐. 새로운 글마다 알려주지 않는다. 그래서, 대학 홈페이지에 올라오는 새로운 글에 대해서 사람들에게 내용에 대해 전달해주는 프로젝트를 구축해 보고자한다.

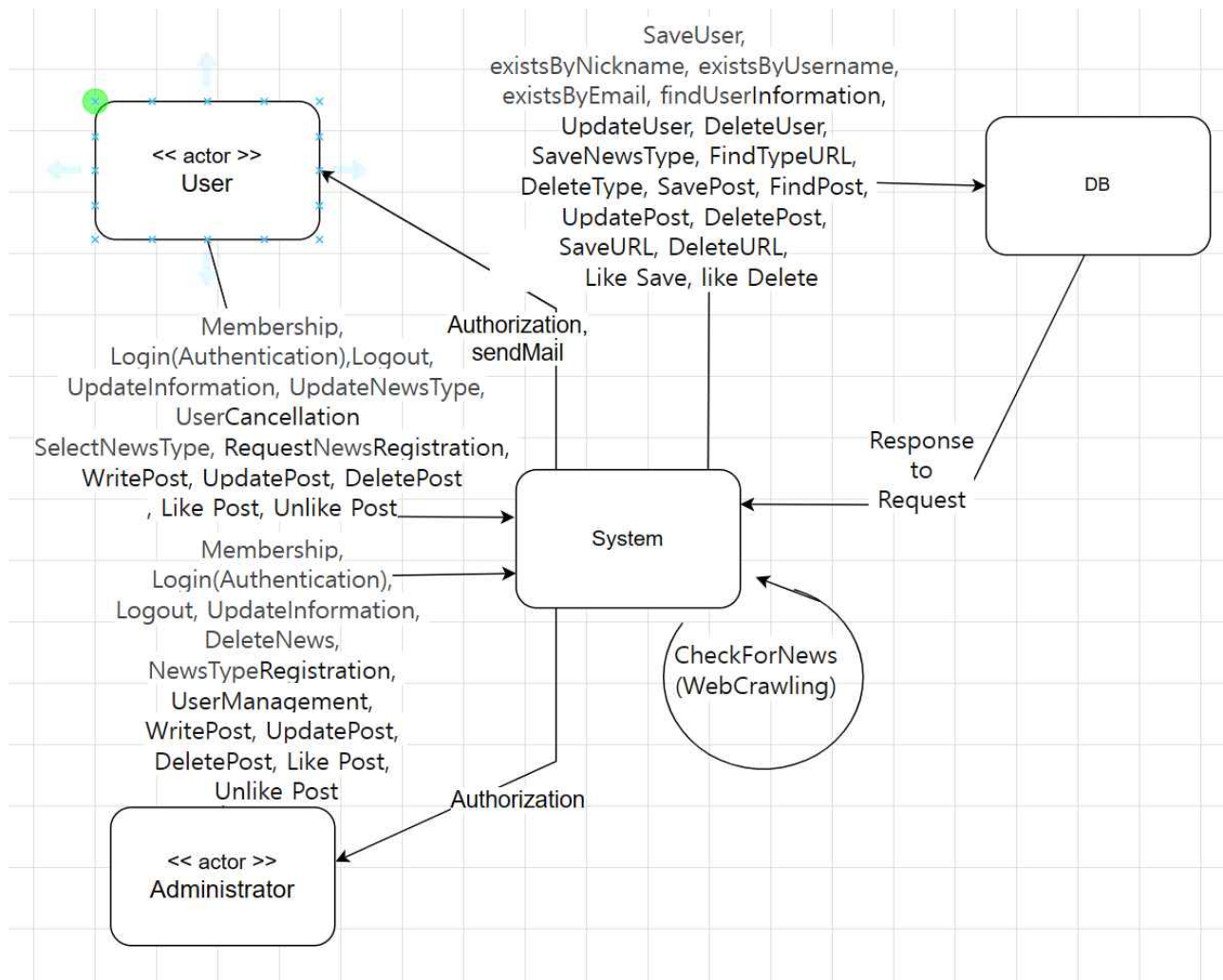
- Goal

1. 사용에 있어서의 편리함. 필요한 기능만 깔끔하게 사용.
2. PC, 모바일 모두가 쉽게 접근할 수 있는 웹으로 구현.
3. 철저한 리팩터링을 통해 서버의 과부하 방지.
4. 최종적으로 SMS 전송을 위한 SMS 서비스 구매.

- Target

이 프로젝트의 주요 사용자로는 대학생으로 생각하고 있다. 영남대학교의 영대소식, 학과 홈페이지에 대한 새로운 글을 전달하고자 하기 때문이다.

2. System context diagram



- MemberShip 회원가입
- Login(Authentication) 로그인(인증)
- Logout 로그아웃
- Authorization 로그인 유지(인가)
- UpdateInformation 회원 정보 업데이트
- UserCancellation 회원 탈퇴
- UpdateNewsType 받고싶은 소식 수정(추가 및 삭제)
- DeleteNews 소식 자체를 삭제
- SelectNewsType 받고 싶은 소식 선택
- RequestNewsRegistration 소식 추가 요청
- RegistrationNewsType 소식 등록
- WritePost 게시글 작성

• UpdatePost	게시글 업데이트
• DeletePost	게시글 삭제
• LikePost	게시글 좋아요 등록
• UnlikePost	게시글 좋아요 취소
• UserManagement	유저 관리
• sendMail	메일 보내기
• SaveUser	DB에 회원 저장
• existsByNickname	DB에 저장된 User 닉네임 중복 확인
• existsByUsername	DB에 저장된 User 아이디 중복 확인
• existsByEmail	DB에 저장된 User 이메일 중복 확인
• findUserInformation	DB에서 User 정보 찾기
• UpdateUser	DB에 저장된 User 업데이트
• DeleteUser	DB에 저장된 User 정보 삭제
• SavePost	DB에 게시글 저장
• FindPost	DB에서 게시글 찾기
• UpdatePost(DB)	DB에 저장된 게시글 업데이트
• DeletePost	DB에 저장된 게시글 삭제
• SaveNewsType	DB에 소식 종류 저장
• FindTypeURL	DB에 저장된 소식 주소 찾기
• DeleteType	DB에 저장된 소식 삭제
• SaveURL	DB에 글 URL 저장
• DeleteURL	DB에 글 URL 삭제
• LikeSave	DB에 글 좋아요 저장
• LikeDelete	DB에 글 좋아요 삭제
• Response to Request	요청에 대한 응답
• CheckForNews(WebCrawling)	새로운 소식이 있는지 확인

3. Use case list

1) Membership

Actor	User
Description	시스템을 사용하기 위해 회원가입 과정을 한다.

2) Login

Actor	User, Administrator
Description	시스템을 사용하기 위해 로그인 과정을 거친다.

3) Logout

Actor	User, Administrator
Description	시스템에서 로그아웃한다.

4) User Management

Actor	Administrator
Description	사용자를 관리한다. 사용자 탈퇴, 사용자가 선택한 소식 등을 관리한다.

5) Find User Information

Actor	User, Administrator
Description	로그인 한 사용자의 회원 정보를 불러온다.

6) Update Information

Actor	User, Administrator
Description	로그인 한 사용자의 회원 정보를 업데이트한다.

7) User Cancellation

Actor	User
Description	사용자의 회원 탈퇴한다.

8) Select News Type

Actor	User
Description	받고 싶은 게시판 소식을 선택한다. (최초 선택)

9) Request News Registration

Actor	User
Description	받고 싶은 게시판 소식 추가를 요청한다.

10) Save Post

Actor	User, Administrator
Description	사용자는 요청할 내용을, 어드민은 공지사항을 글을 통해 올린다.

11) Read Post

Actor	User, Administrator
Description	사용자는 요청할 내용을, 어드민은 공지사항을 글을 통해 올린다.

12) Delete Post

Actor	User, Administrator
Description	올린 글을 삭제한다.

13) Update Post

Actor	User, Administrator
Description	올린 글을 수정한다.

14) Like Post

Actor	User, Administrator
Description	글에 대해 좋아요를 누른다.

15) Unlike Post

Actor	User, Administrator
Description	좋아요한 글에 대해 좋아요를 취소한다.

16) Update News Type

Actor	User
Description	현재 받고있는 게시판 소식에서 추가 및 삭제 과정을 거친다.

17) News Type Registration

Actor	Administrator
Description	사용자가 등록 요청을 한 게시판 소식에 대해 검토 후 추가한다.

18) Delete News

Actor	Administrator
Description	현재 등록되어 있는 게시판 소식을 삭제한다.

4. Concept of operation

1) Membership

Purpose	시스템을 사용하기 위한 인증을 위해 회원가입 진행.
Approach	사용자가 아이디, 비밀번호, 전화번호, 이메일, 받고 싶은 게시판 소식 등을 작성하고 서버에 정보를 보내면 서버에서 DB에 저장을 시킴.
Dynamics	시스템을 최초로 사용할 시 자신의 정보를 저장하기 위한 경우.
Goals	회원가입 기능을 구현한다.

2) Login

Purpose	시스템을 사용하기 위해 서버에 등록된 사용자인지 확인.
Approach	사용자가 아이디, 비밀번호를 입력하여 로그인 요청을 보내면, 서버에서 DB에 있는 정보를 가져와 입력된 정보와 비교를 함. 성공할 경우, 로그인 상태를 유지하기 위해 쿠키에 세션ID를 보냄.
Dynamics	시스템을 실행 시 로그인할 경우.
Goals	로그인 기능을 구현한다.

3) Logout

Purpose	사용자가 시스템 사용 상태를 변경하기 위해 로그아웃을 함.
Approach	사용자가 로그아웃을 하면 서버에 요청을 해 로그인 유지 상태를 제거함. 세션ID를 삭제하여 더 이상 인증되지 않는 상태가 됨.
Dynamics	시스템에서 로그아웃을 할 경우.
Goals	로그아웃 기능을 구현한다.

4) User Management

Purpose	Admin이 사용자들의 정보를 관리하기 위한 목적.
Approach	Admin이 사용자들의 정보를 관리 시도를 하면 서버에서 알맞게 DB에 있던 내용을 수정 혹은 삭제함.
Dynamics	사용자의 요청 또는, 올바르게 못한 방법으로 서버에 데이터를 저장할 경우 (ex. 불건전한 닉네임), 원할한 서버 관리 등과 같은 경우
Goals	사용자의 정보를 관리할 수 있도록 DB에서 값을 불러와 페이지를 구성한다.

5) Find User Information

Purpose	사용자 혹은 Admin이 자신의 정보를 가지고 옴.
Approach	자신에 대한 회원 정보를 조회하면 DB에 저장되어 있는 자신의 정보를 가지고 옴.
Dynamics	자신의 정보에 대해서 알고자 하는 경우.
Goals	DB에서 사용자의 정보를 조회해 가지고올 수 있도록 한다.

6) Update Information

Purpose	사용자 혹은 Admin이 자신의 정보를 수정.
Approach	자신에 대한 회원 정보에 대해 변경사항이 있다면 서버에 요청한다. 그리고, DB에 저장되어 있는 자신의 정보를 요청에 맞게 수정한다.
Dynamics	자신의 정보를 변경하고자 하는 경우.
Goals	DB에서 사용자의 정보를 조회해 사용자의 요청에 맞게 Update를 한다.

7) User Cancellation

Purpose	사용자가 자신의 정보를 삭제.
Approach	사용자가 회원 탈퇴 버튼을 누르면, 서버에 요청을 보내 DB에 저장되어 있는 사용자 정보를 삭제.
Dynamics	시스템의 사용을 그만하고 싶을 때 사용자가 서버에서 자신의 정보에 대해 삭제를 원하는 경우.
Goals	DB에서 사용자의 정보를 조회해 사용자의 정보를 삭제한다.

8) Select News Type

Purpose	사용자가 받고자하는 게시판 소식에 대해서 선택.
Approach	사용자가 원하는 게시판 소식(영대 소식, 원하는 학과 등)을 선택하여 서버에 요청을 보내면, DB에 요청에 맞게 수정.
Dynamics	시스템을 통해 받고자하는 게시판 소식에 대해 추가 또는 삭제를 원하는 경우.
Goals	DB에서 사용자의 정보를 조회해 요청에 맞는 게시판 소식을 추가 또는 삭제한다.

9) Request News Registration

Purpose	사용자가 원하는 게시판 소식에 대해 추가를 원하는 경우.
Approach	사용자가 원하는 게시판 소식에 대한 내용을 글을 통해 추가하면, 서버에서 DB에 사용자의 id로 Post를 추가.
Dynamics	현재는 서버에 소식이 없지만, 사용자가 게시판 소식에 대해 받고자 원하는 경우, Admin에게 요청을 할 때.
Goals	게시판을 만들어 게시판에 게시판 소식 추가 요청에 대한 글을 작성할 수 있도록 구성한다.

10) Save Post

Purpose	사용자는 게시판 소식에 대한 요청 등, Admin은 공지사항에 대해 알리기 위함.
Approach	사용자, Admin이 원하는 내용을 작성하여 서버에 요청을 보내면, 서버에서 DB에 해당 작성자의 id에 맞게 Post를 저장.
Dynamics	글의 작성을 원하는 경우.
Goals	글을 작성할 수 있는 게시판을 만들어 글을 작성할 수 있도록 구성한다.

11) Read Post

Purpose	사용자와 Admin이 작성되어 있는 글을 읽기 위함.
Approach	사용자와 Admin이 읽고자 하는 글을 선택하면, 서버에서 DB에 저장되어 있는 글을 가지고 와 글을 반환해줌.
Dynamics	글을 읽기를 원하는 경우.
Goals	읽고자 원하는 글을 선택하면 화면에 그 글에 대한 내용을 보일 수 있도록 한다.

12) Delete Post

Purpose	글을 올린 작성자가 글을 삭제하기 위함.
Approach	글을 올린 작성자가 자신의 올린 글에 대해 삭제 요청을 서버에 보내면, 서버에서 DB에 해당 Post를 삭제.
Dynamics	자신의 글에 대해 삭제를 원하는 경우.
Goals	글 삭제 버튼을 눌러 자신의 작성한 글에 대해서만 삭제를 할 수 있도록 한다.

13) Update Post

Purpose	글을 올린 작성자가 글을 업데이트하기 위함.
Approach	글을 올린 작성자가 자신의 올린 글에 대해 업데이트 요청을 서버에 보내면, 서버에서 DB에 해당 Post를 요청에 맞게 업데이트.
Dynamics	자신의 글에 대해 변화를 주고자 업데이트를 하는 경우.
Goals	업데이트를 원하는 부분에 대해 작성을 한 후 버튼을 누르면 업데이트가 가능하게 한다.

14) Like Post

Purpose	글에 대해 좋아요 표시를 하기 위함.
Approach	사용자가 글에 대해 '좋아요'를 누르면, 서버에서 해당 글에 대한 좋아요 개수를 업데이트.
Dynamics	특정 글에 대해 좋아요 표시를 넣으려는 경우.
Goals	좋아요를 하고 싶은 글에 대해 좋아요 버튼을 누르면 좋아요가 적용되도록 한다.

15) Unlike Post

Purpose	좋아요한 글에 대해 좋아요를 삭제하기 위함.
Approach	사용자가 좋아요한 글에 대해 좋아요를 취소하고 하면, 서버에서 해당 좋아요를 삭제.
Dynamics	사용자가 좋아요한 글에 대해 좋아요를 취소하고자 하는 경우.
Goals	좋아요 취소를 하면, 좋아요 했던 기록이 사라지도록 한다.

16) Update News Type

Purpose	사용자가 현재 받고 있는 게시판 소식에서 추가 혹은 삭제를 한다.
Approach	사용자가 받는 소식에 대해 변화를 줄 때 서버에게 요청을 하면, 서버에서 DB에 저장된 사용자 게시판 소식 정보를 업데이트함.
Dynamics	사용자가 현재 받고 있는 게시판 소식 정보에 대해서 추가 혹은 삭제 과정을 거치는 경우.
Goals	회원 정보 수정 페이지에서 게시판 소식 타입을 변경할 수 있도록 만든다.

17) News Type Registration

Purpose	사용자가 원하는 게시판 소식에 대해 Admin이 추가를 함.
Approach	사용자가 원하는 게시판 소식을 Admin이 해당 소식에 대한 이름과 사이트 주소를 서버에 보냄. 그리고, 서버에서는 그 내용을 DB에 등록하여 소식 사용 추가.
Dynamics	사용자가 올린 게시판 소식 추가 요청사항 글을 본 후, Admin이 필요하다고 판단하여 등록을 하는 경우.
Goals	Admin만 사용할 수 있는 게시판 소식 관리 페이지를 통해 사용자가 원하는 소식을 추가할 수 있도록 한다.

18) Delete News

Purpose	Admin이 서버에 지정된 게시판 소식에 대해서 삭제를 함.
Approach	Admin이 게시판 소식에 대해 삭제 요청을 보내면, 서버에서 DB에 소식 정보에 대한 내용을 삭제.
Dynamics	Admin이 서버의 과부하 방지를 위해 특정 게시판 소식에 대한 더 이상의 소식 전달이 필요없다고 판단하는 경우.
Goals	Admin만 사용할 수 있는 게시판 소식 관리 페이지를 통해 특정 소식에 대해 삭제를 할 수 있도록 한다.

5. Problem statement

이 프로그램은 새로운 글이 올라온다면, 주기적으로 프로그램이 확인을 하여 등록된 사용자에게 등록된 글의 제목과 URL을 메일로 보내준다. 여기서 볼 수 있는 문제는 아래와 같다.

첫 번째로, 중복된 글의 전달 가능성이다. 서버를 재시작한다면, 이미 보냈던 새로운 글에 대해서도 프로그램이 새로운 글로 인식하여 보내게 된다. 또한, 영남대학교에서 올라오는 글은 7일간 새로운 글이라는 표시인 'N'표시가 있다. 그래서, 사용자가 중복된 글의 전달로 인해 불편함을 느낄 수 있다. 이러한 문제는 DB에 저장하거나, 인메모리 데이터 저장소인 Redis에 저장해서 위와 같은 문제가 발생하지 않도록 해결해야 한다. 이 프로그램은 다른 컴퓨터에서의 실행을 확인하여야 하기 때문에 DB를 통해 문제를 해결하겠다.

두 번째로, 새로운 글에 대한 내용을 DB에 저장할 시, 메모리의 용량이 증가하는 문제이다. 이런 문제가 생긴다면, DB에서 작업을 처리하는 속도가 늦어질 것이고, 메모리 부족 문제가 생길 수 있다. 위에서 영남대학교의 글은 7일간 새로운 글이라는 것을 인식한다고 했는데, 7일 후 DB에서 글에대한 정보가 삭제될 수 있도록 처리하여야 한다.

세 번째로, 프로그램의 오버헤드 가능성이다. 대학 홈페이지를 분석하니 09:00~16:00 시간대가 글이 가장 활발한 시간대이다. 또한, 글도 위 시간대에 계속 올라오는 것이 아니기 때문에 프로그램의 오버헤드 가능성이 있다. 오버헤드를 없애는 것은 불가능하기에 적절한 시간을 설정하여 오버헤드를 최소화하여야 한다.

네 번째로, 프로그램의 비용 생성 가능성이다. 세 번째 이유와 비슷하다. 보통 영남대학교 홈페이지에 올라오는 '영대소식' 같은 경우에는 글이 많이 올라오지만, 학과 게시판은 비교적 적게 올라온다. 이것을 해결하기 위해서 사용자와 소통을 하여 적절하게 생성하거나, 글의 빈도에 따라서 시간 설정을 다르게 해 프로그램을 작동시키도록 해야한다.

마지막으로, 알림 시스템의 문제이다. 이 프로그램은 메일 API를 사용해 사용자에게 새로운 글에 대한 내용을 메일로 보내준다. 하지만, 메일의 알림을 켜지 않는다면, 사용자는 주기적으로 메일을 확인하지 않으면 새로운 글에 대한 소식을 확인 못할 것이다. 또한, 확인한다고 하더라도 직접 클릭하며 확인하기에 이 프로그램을 사용하는 의미가 없어질 것이다. SMS 서비스는 비용이 든다. 그래서, 이 프로그램은 메일을 통한 전송을 일차적으로 하고, 비용을 들어서 SMS 서비스도 활성화하여 영남대학교 학생들의 사용을 할 수 있도록 하는 것이 목표이다.

6. Glossary

용어	설명
User	프로그램을 사용하는 일반 사용자이다. 프로그램에 자신의 정보를 등록하고 원하는 페이지에 대한 소식을 받는다.
Administrator	서버를 제외한 프로그램을 관리하는 사람이다. 회원 관리, 회원이 원하는 게시판 소식 페이지를 등록 및 삭제 등의 역할을 한다.
DB	디스크 기반의 영구적인 데이터 저장소이다. 회원의 정보, 프로그램에서 동작하는 소식 게시판, 사용자가 작성한 글, 새로운 글에 대한 URL을 저장한다.
Authentication	프로그램을 사용하기 위해서 사용되는 동작이다. 사용자가 로그인 시도를 하면, 서버에서 사용자의 정보를 찾아 분석을 하고, 올바르다면 서버에 저장(정확히는 세션ID 저장소)을 한다.
Authorization	서버에 저장된 사용자의 정보를 보고 인증이 필요한 페이지에 접근할 수 있도록 서버에서 확인하는 과정이다.
WebCrawling	페이지의 html을 보고 원하는 정보를 가져올 수 있도록 하는 기술이다. 이 기술을 통해서 새로운 글을 확인하고 사용자에게 보내주는 과정을 하게 된다.
메일 API	사용자가 입력한 이메일 주소로 원하는 내용을 보낼 수 있게 해주는 API이다. API는 소프트웨어 어플리케이션 간에 상호작용을 할 수 있도록 도와주는 것이며, 이 프로그램은 메일과의 상호작용을 위해 메일 API를 쓴다.

7. References

메일 API

<https://docs.spring.io/spring-framework/reference/integration/email.html>

Web Crawling

<https://jsoup.org/>