

КОРОТКИЙ ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Загальна характеристика

Система, яка розроблялась у рамках дипломного проекту, спрямована на використання у вищих навчальних закладах задля того, щоб підвищити ефективність і якість самостійної роботи студентів, їх взаємодії з викладачами та між собою поза аудиторією. Підхід, який було застосовано, об'єднує існуючу систему навчання з сучасними технологіями, надаючи інструменти як уже реалізовані, так і потенційно впроваджені.

Головною ідеєю проекту, яка одразу створила зручний механізм оголошення цільової аудиторії, був обраний принцип «публікація-підписка» (Pub/sub). Він був покладений в основу процесів взаємодії викладачів і студентів, відкриваючи можливість досягати одразу всіх студентів одного факультету, потоку, навчального курсу або групи, в одну мить сповіщаючи обране коло про актуальну для нього інформацію, чи то зміна в розкладі, нове домашнє завдання або інша новина.

При розробці додатку значну увагу було приділено користувацькому інтерфейсу. Важливим завданням було створити його якомога простим у використанні і в той самий час багатфункціональним і зручним, адже ним користуватимуться кожного дня. Основні функції системи винесено в рядок меню, інші згруповано в списки. Всі графічні елементи і кольорові гами було використано відповідно до загальноприйнятих, і тепер навіть тому, хто вперше користується сервісом, буде інтуїтивно зрозуміло, як виконати бажану дію.

Веб-орієнтований додаток складається з окремих модулів, при чому, кожний, згідно з конвенцією Django про модулі багаторазового використання,

незалежний від інших. Разом створено шість модулів, а саме: модуль авторизації користувачів (Authorization), модуль каналів новин (Channel), модуль оголошень (Post), модуль курсів (Course), модуль діалогів (Dialogue) та модуль статистики (Statistics). Призначення кожного модуля та опис застосованих алгоритмів описано в наступних підрозділах, а лістинг вихідного коду моделей, з яких формуються таблиці БД, знаходиться в Додатку 2.

3.2. Модуль Authorization

3.2.1. Авторизація

Модуль Authorization відповідає за авторизацію користувача та налаштування його анкети і профілю. Майже всі базові функції сервісу, за виключенням, звісно, реєстрації і самої авторизації, потребують, щоб користувач був авторизований, тому перед кожною view-функцією можна побачити декоратор `login_required`, як продемонстровано у фрагменті коду на рис. 3.1.

```
1 from django.contrib.auth.decorators import login_required
2
3
4 @login_required
5 def view_function():
6     pass
```

Рисунок 3.1 – Декоратор перевірки авторизації для view-функції

Це означає, що кожного користувача, який на сайті ще не авторизований, буде перенаправлено за адресою `/account/login/` (встановлено за замовчуванням), за яким посиланням він не намагався б перейти. Якщо користувач авторизований, цей декоратор буде пропущено. Аналогічна ситуація відбувається в разі, коли view написано у вигляді класу.

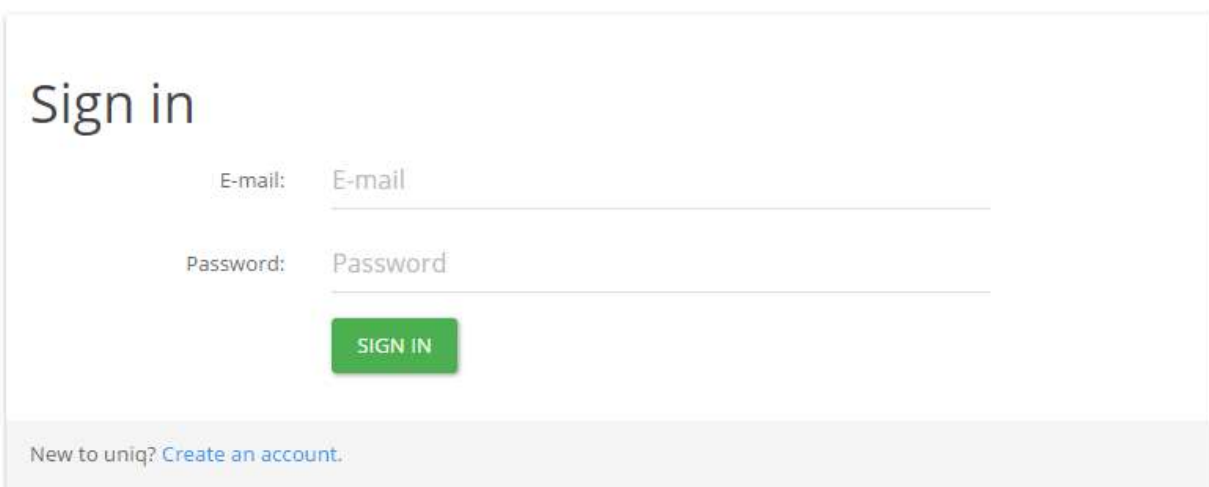
Успадковується так званий «домішок» (mixin) `LoginRequiredMixin`, що демонструє фрагмент коду на рис. 3.2.

```
1 from django.contrib.auth.mixins import LoginRequiredMixin
2
3
4 class ViewClass(LoginRequiredMixin):
5     pass
```

Рисунок 3.2 – Домішок перевірки авторизації для view-класу

Авторизації користувачів передує ідентифікація, яка відбувається за допомогою email-адреси та пароля, вказаних при реєстрації. Якщо введені дані у відповідні поля на сторінці авторизації, показаної на рис. 3.3, співпадають із тими, що зберігаються в базі даних на сервері, користувач авторизується в системі, в іншому випадку його буде повідомлено про помилку авторизації. На випадок, коли обліковий запис ще не було створено, під формою пропонується перейти на сторінку реєстрації за гіперпосиланням, або за адресою `/account/register/` (встановлено за замовчуванням), і пройти просту реєстрацію, що включає в себе заповнення певних полів анкети.

Рисунок 3.3 – Фрагмент сторінки авторизації користувача



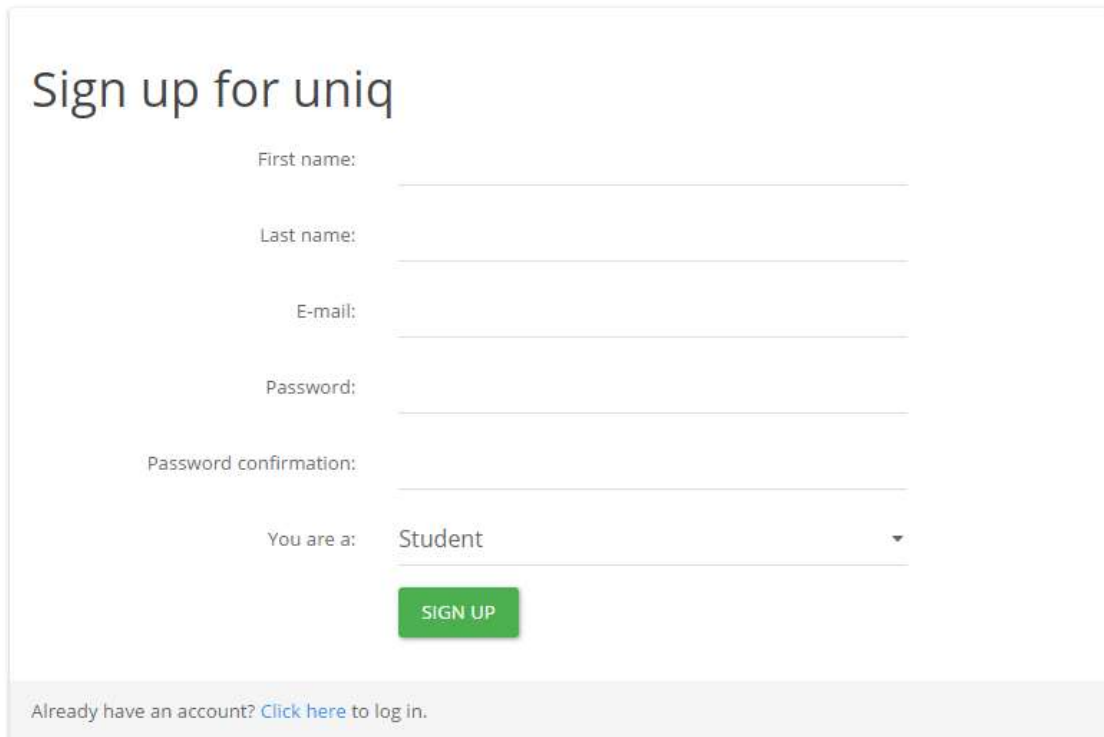
The image shows a 'Sign in' form with the following elements:

- Title: **Sign in**
- Field 1: **E-mail:** followed by an input field containing the placeholder text 'E-mail'.
- Field 2: **Password:** followed by an input field containing the placeholder text 'Password'.
- Button: A green button with the text **SIGN IN**.
- Footer: A light gray bar at the bottom containing the text 'New to uniq? [Create an account.](#)'.

3.2.2. Реєстрація

Реєстрація користувача відбувається після того, як він коректно заповнить анкету на сторінці реєстрації (рис. 3.4), що складається з наступних полів:

- ім'я (та по батькові) користувача;
- прізвище користувача;
- email-адреса, за допомогою якої буде здійснюватись авторизація користувача і відновлення облікового запису в разі втрати до нього доступу;
- пароль, необхідний для входу в обліковий запис;
- підтвердження введеного пароля;
- вибір статусу користувача в університеті з таких варіантів: студент (за замовчуванням), викладач, працівник кафедри чи факультету.



The image shows a registration form with the following fields and elements:

- Title: Sign up for uniq
- First name: [text input]
- Last name: [text input]
- E-mail: [text input]
- Password: [text input]
- Password confirmation: [text input]
- You are a: [dropdown menu with 'Student' selected]
- SIGN UP button (green)
- Footer: Already have an account? [Click here](#) to log in.

Рисунок 3.4 – Фрагмент сторінки реєстрації користувача

Статус користувача – важлива інформація для сервісу. Він указує додатку, яку інформацію відображати саме для цього користувача, надає особистий доступ до окремих функцій і розділів, створює привілеї щодо створення нової та зміни існуючої інформації. В наступних розділах описано різницю між можливими типами користувачів.

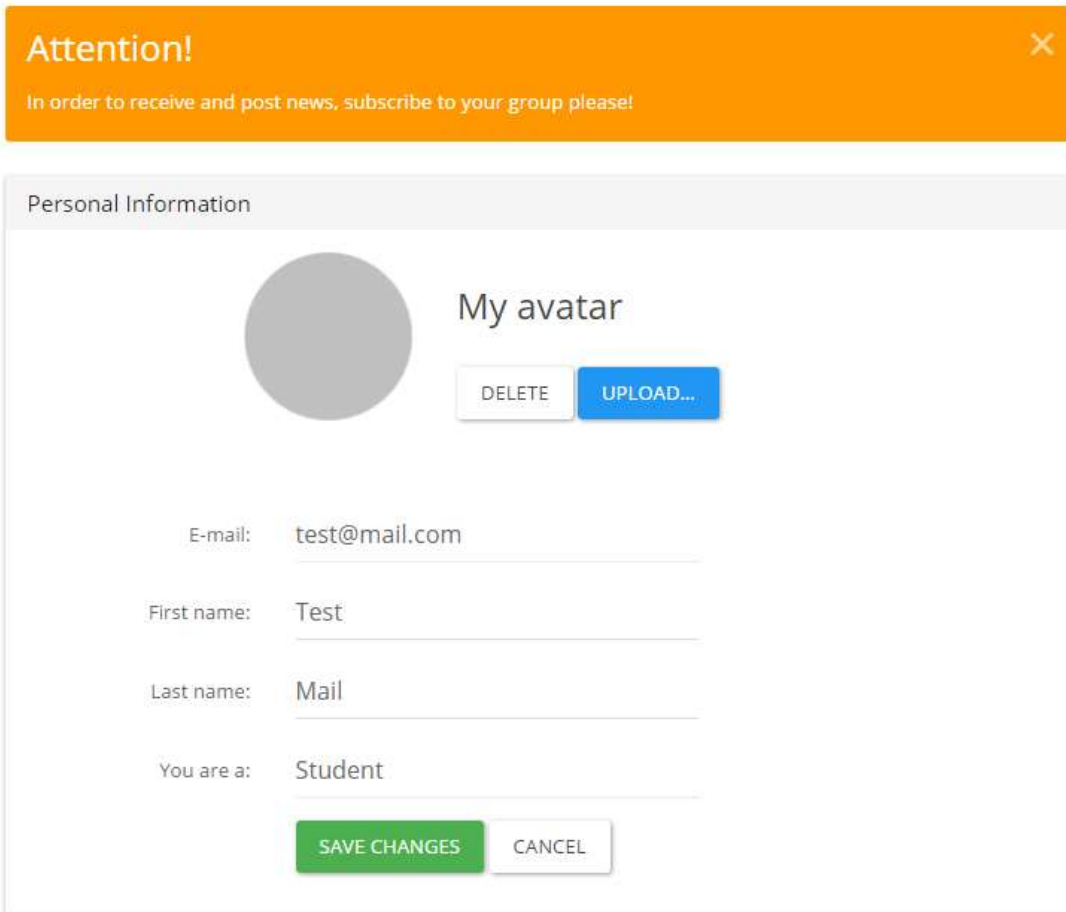
В модулі авторизації описана модель користувача, яка зберігається в базі даних. Через унікальні потреби сервісу, вона була створена власноруч і відрізняється від стандартної моделі, яку пропонує фреймворк. До моделі користувача були додані поля дати створення облікового запису, шляху до каталогу з файлами користувача на файловому сервері, імені файлу аватара користувача та основні канали, на які користувач може бути підписаний.

Необхідно зауважити, що персональні дані, що користувачі вводять, а саме імена та прізвища, зводяться до нормального виду перед збереженням, тобто кожне слово з великої літери, а всі наступні – малі.

Під час створення облікового запису користувача Django зберігає модель в базі даних, встановивши поточну дату створення і шлях до каталогу з файлами користувача, де зберігатиметься його аватар тощо. Але на цей час на файловому сервері вже має існувати такий каталог, тому в першу чергу Django його створює. Для цього необхідно знайти унікальний ідентифікатор каталогу, що складається з рядка «user» та id, причому id – унікальне число, що раніше для цього не використовувалось. Також до моделі автоматично додається первинний ключ, тобто атрибут, за допомогою якого можна однозначно ідентифікувати користувача або обрати відповідний запис з БД.

3.2.3. Профіль

Рис. 3.5 демонструє сторінку профілю користувача, на яку його буде автоматично перенаправлено після успішної реєстрації. Тут він може встановити або видалити свій аватар, змінити деякі персональні дані та вказати основні канали університету (модуль Channels), на які він хоче підписатись. В тому разі, якщо підписка ще не створена на жодний канал, на сторінці профілю користувачу буде запропоновано виправити ситуацію, обравши канал його закладу, щоб отримувати актуальні оновлення. В цей розділ також можна потрапити авторизованому користувачу, знайшовши відповідний розділ у випадяючому списку під ім'ям користувача в правому куті навігаційної панелі.



The screenshot displays a user profile page. At the top, there is an orange attention banner with the text "Attention!" and a close button (X). Below the banner, the page title "Personal Information" is visible. The main content area includes a circular placeholder for an avatar labeled "My avatar", with "DELETE" and "UPLOAD..." buttons. Below the avatar, there are input fields for "E-mail:" (test@mail.com), "First name:" (Test), "Last name:" (Mail), and "You are a:" (Student). At the bottom, there are "SAVE CHANGES" and "CANCEL" buttons.

Рисунок 3.5 – Фрагмент сторінки профілю користувача

3.3. Модуль Channels

Модуль Channels виконує функції, що стосуються маніпуляцій з каналами. Канал – це об’єкт в базі даних, з яким пов’язують новини (модуль Post), оголошення, сповіщення задля того, щоб саме коло користувачів, які створили підписку на цей канал, отримували актуальну для себе інформацію.

3.3.1. Ієрархія каналів

Канали бувають п’яти типів, утворюючи ієрархію, зображену на рис.

3.6:

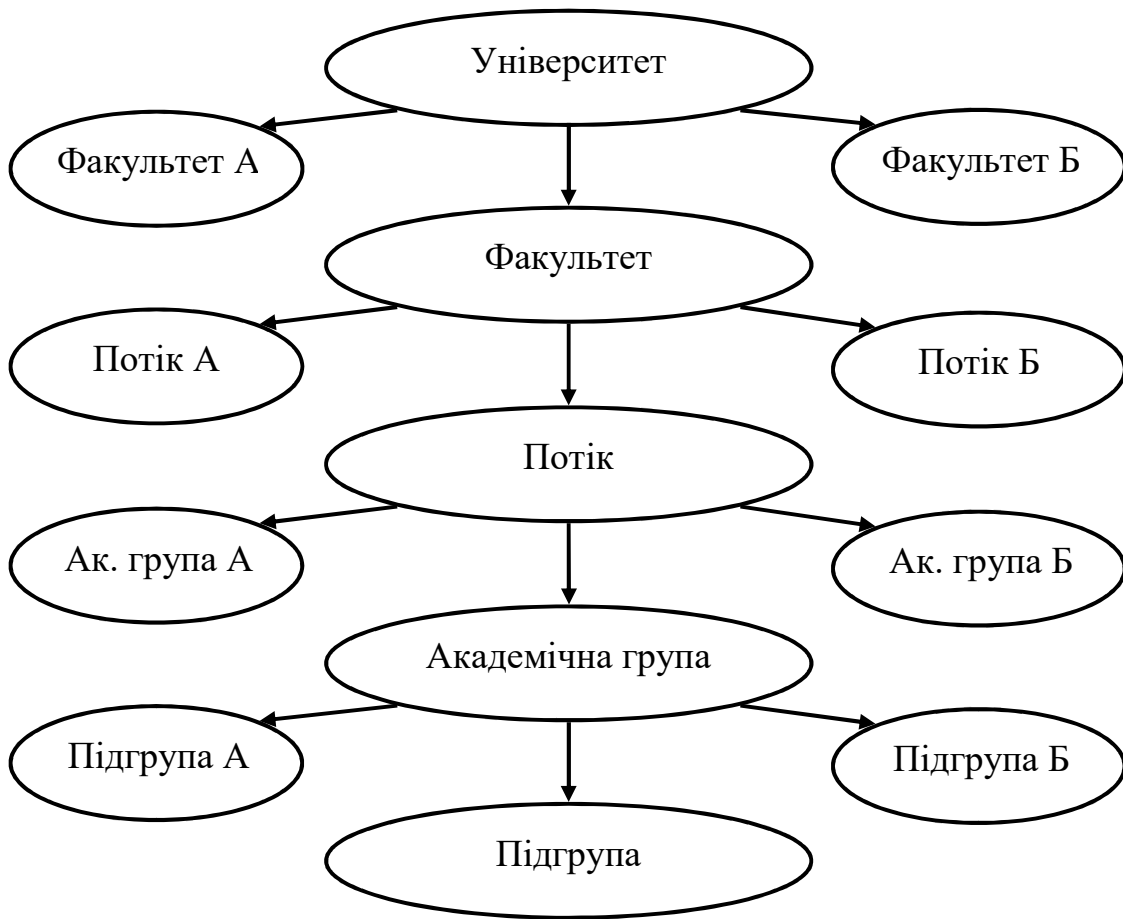


Рисунок 3.6 – Еталонна структура ієрархії каналів

- університетський канал – канал першого рівня, необхідний для створення усіх похідних від нього. Лише адміністратор сервісу може створити цей канал за допомогою спеціальної форми, як показано на рис. 3.7, де вводить інформацію про навчальний заклад у відповідні поля. Необхідними є назва навчального закладу, скорочені загальноприйняті назви, країна та місто, де заклад знаходиться. Додати персонал, який володітиме правом оновлювати (тобто створювати, змінювати і видаляти) інформацію каналу може теж він або вже зареєстрований користувач з перерахованими привілеями;

Add university

| | |
|---------------|---|
| Title: | Національний технічний університет України |
| Short titles: | нтуу, <small>Input short names, dividing by coma</small> |
| Country: | Україна |
| City: | Львів |

Рисунок 3.7 – Фрагмент форми для створення каналу університету

- канал факультету – канал наступного рівня, який може бути створений лише робітником університету з привілеями оновлення інформації, вказавши у відповідній формі, як показано на рис. 3.8, назву факультету, та обравши університет, в якому він знаходиться. Оновлювати інформацію на цьому рівні можуть користувачі, які

отримали привілеї від привілейованих користувачів цього або вищого рівня;

Add faculty

| | |
|-------------|---|
| Title: | <input type="text" value="Факультет біотехнологій"/> |
| University: | <input type="text" value="Bogomolets National Medical University"/> ✎ + |

Рисунок 3.8 – Фрагмент форми для створення каналу факультету

- канал потоку, тобто набору академічних груп, що навчаються за одним напрямом паралельно, може бути створений привілейованими користувачами верхнього рівня. Для цього в спеціальній формі, продемонстрованої на рис. 3.9, необхідно вказати назву потоку, тобто спеціальність, за якою навчаються академічні групи, факультет, додаткову інформацію, якщо необхідно, рік вступу та випуску. Створювати оновлення на цьому рівні можуть користувачі як попереднього, так і наступного рівнів, але підписатись на цей канал жодний не може;

Add group stack

| | |
|----------------|---|
| Title: | <input type="text" value="Системний аналіз"/> |
| Faculty: | <input type="text" value="IASA"/> ✎ + |
| Year enter: | <input type="text" value="2017"/> |
| Year graduate: | <input type="text" value="2021"/> |

Рисунок 3.9 – Фрагмент форми створення каналу потоку

- канал академічної групи створюється також у спеціальній формі, як показано на рис. 3.10, привілейованим підписником каналу факультету, який може створювати, переглядати і редагувати оновлення користувачів тільки свого рівня. Дивитись усі оновлення, редагувати власноруч додані та підписуватись на канал вказаного рівня можуть лише ті користувачі, які встановили статус «студент». Окрім створення новин у своїй академічній групі, студенти можуть обрати варіант поділитись нею з усім потоком, отже, підписники всіх груп на зазначеному потоці отримають це оновлення;



Add group

Title:

Group stack:  

Рисунок 3.10 – Фрагмент форми для створення каналу академічної групи

- канал підгрупи – нижчий рівень ієрархії, може бути створений будь-яким студентом, підписаним на певний канал академічної групи. Підписатись на цей канал є можливість лише за запрошенням студентів тієї самої групи.

Така система існує для того, щоб користувачі, які підписуються на канал, мали доступ і отримували сповіщення про всі оновлення каналів, що

знаходяться вище за ієрархією, а автори оновлень могли легко сповіщати усіх користувачів, які слідкують за оновленнями каналів, що нижче за ієрархією.

3.3.2. Керування підписками

На сторінці профілю користувача присутня форма, представлена на рис. 3.11, яка призначена для зручного управління підписками на основні канали, тобто головний – новини, що стосуються всього університету, факультетський або канал академічної групи, якщо користувач – студент. Відписатись від каналу або підписатись на будь-який з основних можна одразу тут, натиснувши на червону або зелену кнопки відповідно, розташовані праворуч від назви каналу. Функція відписки діє наступним чином: коли користувач обирає відписатись від каналу певного рівня, розташовані нижче за ієрархією канали також скидаються. Щоб обрати канал для підписки, треба натиснути на зелену кнопку, яка перенаправить на сторінку пошуку каналу певного рівня, але не нижчого за ієрархією, ніж перший вільний. Такий сценарій логічний, бо зазвичай студент навчається в одному університеті, факультеті та групі.

Subscription

You are subscribed to:

| | | | |
|-------------|--|--------------------------|--------|
| University: | Igor Sikorsky Kyiv Polytechnic Institute | <input type="checkbox"/> | CHANGE |
| Faculty: | Faculty is not specified yet | <input type="checkbox"/> | CHANGE |
| Group: | Group is not specified yet | <input type="checkbox"/> | CHANGE |

Рисунок 3.11 – Фрагмент форми підписки на основні канали

Можливості користувачів на всіх рівнях ієрархії, відповідно до їх статусу, детально продемонстровано на Схемі взаємодії користувачів в додатку 1. У схемі зроблено наголос на те, що користувачі зі статусом «викладач» та «робітник університету» мають змогу підписуватись лише на канали рівнів університет та факультет, отже, на каналах рівнів потік і нижче за ієрархією взаємодія відбувається виключно між студентами.

3.3.3. Стрічка новин

На сайті сторінка, на яку заходить користувач сервісу, щоб передивитись оновлення, називається «стрічка». До речі, сюди буде перенаправлено користувачів одразу після їх успішної авторизації. В цьому розділі користувач за замовчуванням переглядає список оновлень, приклад яких наведено на рис. 3.12, сформованих автоматично модулями сервісу або власноруч студентами, з усіх каналів, на які він підписаний. Тобто, якщо користувач – студент і вказав свою академічну групу, він буде отримувати оновлення каналу його академічної групи та всіх каналів, що знаходяться вище за ієрархією.

The screenshot displays a news feed interface. On the left, there is a vertical list of channels with their respective post counts:

- All together (3)
- Ihor Sikorsky Kyiv Polytechnic Institute (2)
- FCIT (0)
- Computer Science (1)
- КС-42 (0)

The main content area shows a news post from 'Миколай Наушко | Student' dated 'June 17, 2017, 8:56 p.m.'. The post title is 'Нова школа' and the text discusses expert evaluations of the job market and the importance of continuous learning and critical thinking. A green button labeled '+ ADD NEW POST' is visible in the top right corner. Below the post, the word 'University' is partially visible.

Рисунок 3.12 – Фрагмент сторінки зі стрічкою новин

Всі новини, які користувач бачить у стрічці, можуть бути позначені як важливі (★) або неактуальні (✓). Важливі оновлення отримують кольорову зірочку і зберігаються, щоб користувач міг переглянути їх у відповідному розділі, а от неактуальні новини доступні для перегляду лише на вкладці певного каналу: з загальної стрічки вони приховуються.

На сторінці стрічки новин користувач має можливість переключитись на вкладку оновлень окремого каналу. Обравши певний зі списку підписок, користувач побачить новини, сформовані лише для цього каналу. Ця функція зручна у випадку, коли розшукується інформація за певним джерелом. В правому куті панелі навігації відображається кількість нових непереглянутих сповіщень, тому важливу інформацію буде складно пропустити.

3.3.4. Типи оновлень

Оновлення зі стрічки новин бувають таких типів:

- оголошення, яке користувачі створюють на каналі (описаний в модулі Post);
- сповіщення, сформоване системою під час оновлень курсу (модуль Course), який вивчає студент;
- сповіщення, яке формує модуль Statistics;
- системне сповіщення, що стосується користувача.

Незалежно від типу, оновлення відповідають правилам матеріального дизайну, які диктують певний зовнішній вигляд, і мають декілька типових ознак. Під кожним оновленням зображується час та день, коли воно було

створено, а також дата, до якої воно актуальне, якщо це вказано. Неактуальні оновлення з загальної стрічки зникають. Окрім цього, присутні кнопки для збереження оновлення як важливого та неактуального.

Оголошення створюються користувачами і доступні для перегляду всім, хто вказаний в колі отримувачів. В стрічці новин демонструються їх назви, повний текст (або частина, якщо він перевищує ліміт в 250 літер, і кнопка, щоб показати текст повністю), три останніх коментаря (описано в модулі Post) з можливістю переглянути всі, ім'я та прізвище автора.

Сповіщення від модуля Course у різний спосіб допомагають студентам постійно покращувати свою успішність. Звідси користувачі дізнаються про отримання відповіді на своє запитання, надходження чергового завдання або закінчення терміну виконання попередніх. Студенти розраховуватимуть свій час краще, коли побачать весь об'єм роботи.

Модуль Statistics також формує оголошення. Механізм створення рейтингу надає виняткові можливості для заохочення студентів систематично перемагати себе і досягати кращих результатів, адже відомо з економіки [10], що конкуренція підтримує динамізм у розвитку конкурентів і підвищує продуктивність праці.

Системні сповіщення формуються задля того, щоб попередити користувача про зміну в послугах, що надаються, або ознайомити з новим функціями.

3.3.5. Пошук

На сайті в панелі навігації розміщена форма пошуку, якої можна дістатись із будь-якої сторінки. Пошук здійснюється по всіх оголошеннях

каналів, на які підписаний користувач, а також збережених важливих, на випадок, якщо від цього каналу була здійснена відписка; по назвах курсів і завдань у кожному. Знайдені результати згруповуються за розділами, як показано на рис. 3.13, надаючи можливість переглянути повний список за посиланням.

Found 1 result(s) by search of "Нова школа":

Section: Channels' posts



Section: Subjects' posts

Рисунок 3.13 – Фрагмент сторінки пошуку оголошень

3.4. Модуль Post

Модуль Post, мабуть, є головною складовою того принципово нового механізму взаємодії викладачів зі студентами. Коли користувач створює оголошення (таблиця Post в БД) для якого-небудь кола людей, це оголошення бачать усі користувачі з цього кола і мають можливість залишити під ним коментар (таблиця Comment в БД), що дозволяє обом бокам шаблону Pub/sub,

видавцям і підписникам, обмінюватись інформацією стосовно саме започаткованої теми видавця. Отже, кожне оголошення містить список коментарів, які створюють підписники в окремому призначеному для цього місці, унеможливаючи появі плутанини з адресацією інформації, часом створення відповіді (коментаря), метою тощо. Поява відповіді до оголошення буде супроводжуватись сповіщенням того, хто це оголошення надіслав, тому пропустити важливу інформацію буде складно, адже це може бути не лише новина про якусь подію, а й питання щодо виконання чергового завдання і від швидкості отримання відповіді буде залежати термін виконання цього завдання.

Функція для створення оголошення існує як в модулі Channel, так і в модулі Course. Ідея надання студентам інструменту допомагати один одному, знаходячи разом вірне рішення проблеми, запозичена в інформаційних спільнотах. Кожний може оголосити проблему, пов'язану з навчанням, на сторінці відповідного курсу (модуль Course) або на каналі академічної групи, якщо вона загального характеру. Тоді всі інші користувачі з аналогічними привілеями побачать це оголошення і зможуть запропонувати своє рішення або підтримати існуюче, віддавши йому свій голос. Така схема роботи в команді сприятиме підвищенню рівня знань у всіх студентів, бо ці оголошення доступні всім, а також позитивно вплине на товариські стосунки в групі.

Додавати свої коментарі можуть не тільки студенти, а й викладачі, якщо оголошення з'явилося у розділі деякого курсу. Звісно, відповідь викладача – значно авторитетніша, але змога підтримати коментар інших студентів відкриває безліч можливостей для заохочення поділитися знаннями з іншими, отримавши за це певну винагороду в якості балів, наприклад.

В модулі Post існує дві моделі, які зберігаються в окремих таблицях БД:

- модель Post, яка описує інформацію про оголошення;
- модель Comment, яка містить інформацію про коментар.

3.4.1. Модель Post

Модель Post складається з низки полів, деякі з них користувач заповнює власноруч у спеціальній формі, наведеній на рис. 3.14, а решта генерується автоматично. Користувач вводить такі поля:

- заголовок оголошення, за яким зручно його шукати. Поле має обмеження в 150 символів і показується у стрічці повністю;
- поле тексту оголошення. Воно також має обмеження за довжиною, але значно більше – 5000 символів;
- поле для завантаження файлів. Максимальна кількість – 10;
- канал, на який поширяться створене оголошення;
- день, включно до якого це оголошення буде актуальним (необов'язкове поле).

The image shows a form for creating a new announcement. The fields are as follows:

- Title:** Дистанційна освіта
- Text:** Дистанційна освіта - це можливість навча
- Author:** super@user.com
- Is subject:**
- University:** -----
- Faculty:** -----
- Group set:** Computer Science

Рисунок 3.14 – Фрагмент форми для створення нового оголошення

До них автоматично додаються наступні поля:

- первинний ключ цього запису;
- зовнішній ключ, який вказує на автора коментаря;
- дата створення оголошення.

Створене та поширене оголошення може бути видалено або змінено в тій самій формі. Для редагування доступні ті самі поля, що і під час створення, за винятком каналу, який отримує це оголошення.

На кожному оголошенні вказано час створення, автора, текст та назву, за якою можна цей пост знайти. Під кожним оголошенням показано максимум три коментаря, що отримали найбільшу підтримку користувачів та є можливість додати ще один.

3.4.2. Модель Comment

Модель Comment схожа на модель Post. Користувач друкує коментар під певним оголошенням, тому зникає необхідність додатково вказувати канал, а також коментар не має заголовку. В моделі Comment є такі поля:

- зовнішній ключ, який вказує на пост, якого коментар стосується;
- зовнішній ключ, який вказує на автора коментаря;
- текст коментаря, обмежений 1000 символів;
- дата створення;
- кількість користувачів, які підтримали цей коментар.

Коментар може видалити лише його автор.

3.5. Модуль Course

Модуль Course – важлива складова сервісу, яка забезпечує підтримку курсів: створення, керування, підписку на них окремих користувачів або одразу академічної групи. На сайті, звісно, окремий розділ присвячений курсам, який доступний всім користувачам, але з різними привілеями. На цій сторінці студенту показано всі курси, з якими він має справу, тобто активні курси. Неактивні курси, які закінчились, або користувач від них відписався, ховаються в спеціальній вкладці Archive. Окрім перегляду існуючих курсів, у викладачів є можливість створити новий курс, а студентів – знайти будь-який і підписатись. Ця опція надає шанс самостійно обирати додаткові або непрофільні предмети, бо обов’язкові предмети пов’язані з академічною групою, на них не потрібно підписуватись окремо і від них не можна відписатись.

3.5.1. Курс

Курс – аналог каналу, але надає додаткові функції як для студентів, так і викладачів. Щоб створити курс, викладачу необхідно заповнити певну форму, показану на рис. 3.15, яка складається з наступних полів:

Add subject

| | |
|----------------------------------|--|
| Title: | <input type="text" value="Дискретна математика"/> |
| Description: | <div><p>Мета – сформувати у студентів знання, вміння і навички, необхідні для засвоєння курсу програмування, побудови дискретних математичних моделей реальних об’єктів, проектування систем обробки інформації з використанням алгебричного підходу, розробки ефективних алгоритмів та їх аналізу.</p><p>Завдання вивчення дисципліни – навчити студентів використовувати апарат дискретної математики для розв’язування практичних задач, що пов’язані з розробкою програмних комплексів для EOM та створенням алгоритмів вирішення прикладних проблем.</p><p>У результаті вивчення навчального курсу студент повинен знати:</p></div> |
| Channel: | <input type="text" value="-----"/> |
| Date start: | <input type="text" value="2017-09-01"/> Today |
| Date finish: | <input type="text" value="2018-01-16"/> Today |
| <input type="checkbox"/> Is open | |
| Max points: | <input type="text" value="100"/> |

Рисунок 3.15 – Фрагмент форми для створення нового курсу

- назва курсу, за якою цей курс можна знайти;
- опис курсу, який включатиме головні теми обговорення, кількість кредитів за курс, дещо про самих викладачів;
- дату початку курсу та його тривалість;
- тип курсу, що вказує, чи зможуть на нього підписатись окремі студенти або лише академічні групи;
- кількість балів, які можна заробити упродовж курсу, виконуючи передбачені завдання;
- викладачі, окрім того, що створює курс. Після створення курсу вони отримають такі самі привілеї щодо керування курсом і додаванням матеріалів.

Наведені поля зберігатимуться в БД у спеціальній моделі Course разом з автоматично створеними:

- зовнішній ключ, що вказує на викладача курсу;
- розміщення сховища з матеріалами курсу, яке створюється на сервері разом із курсом.

Коли курс створено, в ньому можна додавати матеріали, до яких матимуть доступ усі студенти цього курсу. Разом з цим викладач зазначає кількість балів, які заробить студент, виконавши завдання вчасно та повністю, і спосіб, у який завдання розподіляються між студентами: вони обирають

самостійно з доступних варіантів, або отримують випадкове. Виконане завдання студент завантажує на сервіс і, натиснувши на кнопку «надіслати», не може скасувати цю дію. Потім викладач отримує сповіщення про надходження роботи і може його перевірити. Унікальність такого механізму відносно існуючого порядку здачі очно полягає в тому, що викладач не знає, чию роботу він перевіряє, отже, на його оцінку не впливатимуть різні суб'єктивні чинники. Все, що викладачу відомо – варіант завдання роботи. Перевіривши надісланий варіант, викладач виставляє оцінку або залишає відгук щодо роботи з рекомендаціями дещо виправити, після чого студент має змогу надіслати дороблену версію. Усі версії відправлених робіт будуть збережені на сервері назавжди без можливості видалення – це має замінити необхідність їх друкувати та приносити в паперовому вигляді, щоб залишити в архіві кафедри. Це позбавить студентів від марного витрачання часу на візит до типографій, покращить екологію в країні і звільнить кабінет з-під архіву для більш корисних потреб.

Студент переходить на певний курс, і в нього відкривається сторінка з усіма завданнями, які він уже зробив і які залишились відкритими. Незроблені завдання також мають таймер, що показує кількість днів до кінцевого терміну здачі. Обравши завдання, студент може завантажити його собі на комп'ютер або переглянути на сайті. У разі, якщо виникають запитання, звідси їх можна поставити, тоді вони з'являться на каналі курсу і міститимуть тему поточного завдання. Коли завдання виконано і готове до перевірки, його завантажують на сайт і надсилають викладачеві.

Студент знає, скільки балів можна заробити за кожне завдання та вже отримані бали. Для того, щоб допомогти користувачу слідкувати за своєю успішністю, в кожному курсі підраховується сума отриманих балів за

пройдений час і обчислюється відсоток закінченості курсу. Її можна відслідкувати окремо на курсі чи одразу по всіх у розділі Progress.

Після того, як курс закінчується, у студентів залишається до них доступ, але на них більше не можна підписуватись і завантажувати файли на сервер, а от зберегти собі матеріал можна завжди. Також можна поставити запитання на каналі курсу або залишити коментарі. Викладач може відновити курс після його закінчення, що дозволяє залишити доступ до активного курсу для попередніх студентів і створені оголошення для нових. В цьому разі, перші матимуть привілеї лише передивлятися і створювати оголошення, а всі завдання та прогрес залишаться попередніми.

3.6. Модуль Dialogue

Окрім публічних каналів, напевно, виникає ситуація, коли необхідно особисто щось обговорити. Це може бути безпосередньо пов'язано з навчанням або якимось чином стосуватись академічної групи, в будь-якому разі, користувачів необхідно забезпечити інструментом спілкування один-на-один, для чого і створено модуль Dialogue.

Головною в цьому модулі є модель Message, яка вказує спосіб зберігання інформації стосовно повідомлень. В ній містяться наступні поля:

- первинний ключ; він додається автоматично і є унікальним ідентифікатором для кожного повідомлення;
- поле адресанта повідомлення – зовнішній ключ, який посилається на модель користувача в модулі Authorization;
- поле адресата повідомлення – зовнішній ключ, який посилається на модель користувача в модулі Authorization;

- поле дати, в якому зберігається час і день надсилання повідомлення.

В діалозі передбачено участь двох користувачів, тому для створення діалогу достатньо натиснути на ім'я користувача в списку підписників чи викладачів, перейшовши на сторінку опису каналу або курсу. Механізму для пошуку користувачів іншим шляхом не існує, бо ця опція виходить за межі ідеї проекту. На сервісі також є спеціальний розділ, у якому користувач переглядає список діалогів і відкриває необхідний. Зображення цього розділу з екрану показано на рис. 3.16. Цей розділ Вибір повідомлень для того, щоб показати користувачеві, виконується запитом до БД, які наступним кроком сортуються за датою створення, як продемонстровано у фрагменті коду на рис. 3.17.

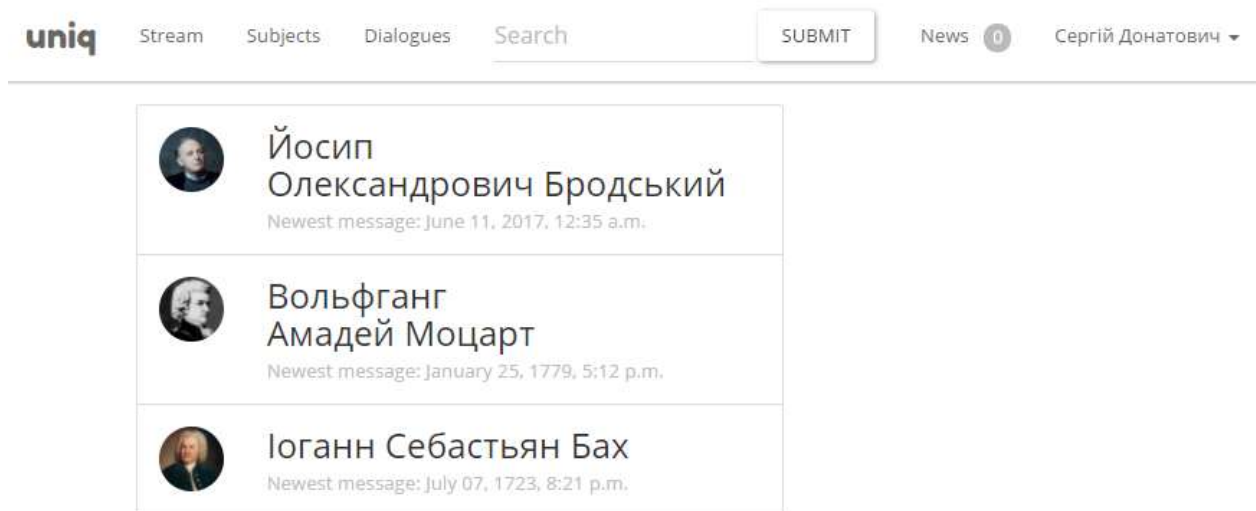


Рисунок 3.16 – Фрагмент сторінки зі списком діалогів

```
1 from .models import Message
2
3
4 def get_messages(dialogue_pk):
5     messages = Message.objects.filter(dialogue=dialogue_pk)
6     return messages.order_by('date')
```

Рисунок 3.17 – Фрагмент коду пошуку повідомлень з діалогу

Знизу сторінки з діалогом знаходиться форма для створення нового повідомлення. В полі необхідно надрукувати текст повідомлення та натиснути на кнопку Send (надіслати), щоб співбесідник його отримав. Надсилання та отримання повідомлень здійснюється за допомогою AJAX-запитів, тому користувачу не потрібно оновлювати сторінку, щоб побачити нове повідомлення.

3.7. Модуль Statistics

Особливістю розробленого сервісу став механізм, який обчислює рейтинг усіх студентів, базуючись на їхній поточній успішності в межах певного рівня ієрархії. З періодичністю в місяць буде підраховуватись відсоток отриманих балів студентів за кожний курс (дисципліну), виходячи з максимальної суми балів за ті завдання, кінцевий термін яких знаходився в проміжку часу з початку семестру і до поточного місяця включно. Відсоток отриманих балів за всі дисципліни буде підсумовуватись, щоб знайти середній. Результат перерахується у стобальну систему, відповідно до якого і буде складено загальну рейтингову систему по потоку, факультету та університету. Всі результати зберігатимуться в БД і будуть доступні в розділі Progress.

Рейтинг окремого студента формується у сповіщення саме для нього і з'являється в його стрічці новин. А от студентів, які потрапили в список топ-10 людей з найкращими балами на кожному рівні еталонної схеми каналів, публічно привітають на відповідних рівнях. Очікується, що такий механізм статистики спонукатиме студентів приділяти увагу всім дисциплінам та навчить їх ефективніше планувати свій час.