

Филиал Московского Государственного Университета
имени М.В. Ломоносова в городе Ташкенте

Факультет прикладной математики и информатики
Кафедра прикладной математики и информатики

Абдуллаева Евгения Гасановна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
на тему: «Разработка модулей оповещения и статистики для
системы дистанционного обучения»

по направлению 01.03.02 «Прикладная математика и информатика»

Выпускная квалификационная работа рассмотрена и рекомендована к защите

Руководитель Филиала МГУ имени М.В. Ломоносова в г. Ташкенте
к.ф.-м.н., доцент Строгалов Александр Сергеевич

Научный руководитель
к.ф.-м.н., в.н.с. Алисейчик Павел Александрович

«_____» _____ 2021 г.

Ташкент 2021 г.

Аннотация

В настоящей работе приведены результаты, достигнутые в процессе расширения функциональности системы дистанционного обучения «МГУ Контест»: разработке модуля оповещения пользователей и модуля статистики, а также усовершенствовании раздела помощи пользователям.

В модуле оповещения реализованы возможность подписки пользователя на определенные сущности, оповещение пользователей о событиях на сайте, интерфейс настройки оповещений; переработано отображение списка оповещений. В модуле статистики усовершенствована страница профиля студента, добавлено отображение сведений об активности и успеваемости отдельного студента, а также рейтинг студентов; в список курсов внедрено отображение сведений о сложности курсов. Реализована основная страница раздела помощи пользователям.

Abstract

This paper presents the results achieved in the process of expanding the functionality of the “MSU Contest” distance learning system: developing a user notification module and a statistics module, as well as improving the user help section.

The notification module implements the ability to subscribe a user to certain entities, notify users about events on the site, an interface for configuring notifications; redesigned display of the list of notifications. In the statistics module, the student profile page has been improved, the display of information about the activity and progress of an individual student, as well as the rating of students has been added; the display of information about the complexity of courses has been implemented in the list of courses. The main page of the user help section has been implemented.

Содержание

1	Введение	4
2	Общая структура и организация работы	5
3	Модуль оповещения	6
3.1	Подписка на курсы и разделы	6
3.2	Оповещения о событиях на сайте	6
3.3	Список оповещений	7
3.3.1	Бесконечная прокрутка списка оповещений	7
3.3.2	Динамическая отметка оповещений как прочтенных	7
3.4	Интерфейс настройки оповещений	7
3.4.1	Выбор получаемых оповещений	7
3.4.2	Список подписок на курсы и разделы	8
4	Модуль статистики	9
4.1	Профиль студента	9
4.1.1	Персонализированные изображения профиля	9
4.1.2	Отображение сведений об активности	9
4.1.3	Результаты курсов	9
4.2	Рейтинг студентов	9
4.2.1	Подсчет общей успеваемости и точности студента	9
4.2.2	Рейтинг студентов по факультетам, потокам и курсам	9
4.3	Сведения о сложности курсов	9
5	Раздел помощи	10
5.1	Адресаты вопросов	10
5.2	Отображение списков вопросов и сообщений об ошибках	10
6	Заключение	11
7	Приложения	12
7.1	Модель подписки	12
8	Список использованных источников и литературы	15

1 Введение

2 Общая структура и организация работы

Данная работа содержит две основные части, описанные в главах 3 и 4. В главе 3 представлено описание работы, проделанной над модулем оповещений, в главе 4 – описание работы над модулем статистики системы дистанционного обучения «МГУ Контест». Система реализована с использованием фреймворка для веб-приложений Django [4], написанного на языке программирования Python [5], в связи с этим расширение функциональности системы также производится с использованием указанных технологий.

В модуле статистики для реализации графического отображения собранных сведений используется инструмент Chart.js [6].

3 Модуль оповещения

3.1 Подписка на курсы и разделы

Персонализация оповещений о событиях на сайте требует реализации механизма подписки пользователей на определенные сущности, о состоянии которых пользователь желает получать оповещения. Поэтому для предоставления преподавателям возможности получать оповещения о новых комментариях и посылках лишь к определенным курсам и разделам была реализована подписка пользователя на курсы и разделы.

На детальной странице каждого курса реализован интерфейс, доступный преподавателям, предоставляющий возможность подписаться на данный курс или отписаться от курса, а также подписаться на данный курс и все его разделы или отписаться от них. На детальной странице каждого раздела реализован интерфейс, доступный преподавателям, позволяющий подписаться на данный раздел или отписаться от него.

Все пользователи имеют возможность подписаться на оповещения об объявлениях и о расписании. Такую возможность предоставляет интерфейс настройки оповещений, реализация которого описана ниже.

3.2 Оповещения о событиях на сайте

Для пользователей сайта реализована возможность получать оповещения о следующих событиях:

- о добавлении комментария к курсу, разделу, задаче или посылке – оповещения получают преподаватели, подписанные на курс или раздел, к которым написан данный комментарий, если же комментарий написан к задаче или посылке к задаче, то оповещения получают преподаватели, подписанные на раздел, включающий данную задачу;
- о новой посылке к задаче – оповещения получают преподаватели, подписанные на раздел, включающий задачу, к которой отправлена посылка;
- о публикации объявления – оповещения получают все пользователи, находящиеся в группе, которой адресовано объявление, и подписанные на оповещения об объявлениях;
- о добавлении расписания – оповещения получают все пользователи, подписанные на оповещения о расписании;
- о создании вопроса – оповещение получает преподаватель, которому адресован вопрос, при переадресации вопроса или при появлении ответа на вопрос, оповещение об этом получает пользователь, задавший данный вопрос;

- о сообщении об ошибке – оповещения получают администраторы сайта, при изменении статуса сообщения оповещение получает пользователь, отправивший сообщение об ошибке.

3.3 Список оповещений

3.3.1 Бесконечная прокрутка списка оповещений

Для списка оповещений пользователя реализован механизм динамического раскрытия списка при прокручивании. При переходе на страницу списка оповещений отображаются лишь последние 30 полученных пользователем оповещений, при прокручивании страницы к концу отображаемого списка подгружаются следующие 30 более давних оповещений.

3.3.2 Динамическая отметка оповещений как прочтенных

В списке оповещений еще не прочтенные пользователем оповещения помечаются специальным цветом. При их просмотре в списке, то есть при нахождении таких оповещений в области видимости пользователя, они динамически помечаются как прочтенные, а также динамически обновляется значение счетчика непрочтенных оповещений. При просмотре объектов непрочтенных оповещений, например, комментария или объявления, о добавлении которых были получены данные оповещения, данные оповещения также помечаются как прочтенные.

3.4 Интерфейс настройки оповещений

3.4.1 Выбор получаемых оповещений

Пользователям предоставляется возможность выбора типов получаемых оповещений. Для этого реализован интерфейс настройки оповещений, где пользователь может отметить некоторые из следующих пунктов:

- объявления – доступен всем пользователям;
- расписание – доступен всем пользователям;
- комментарии – доступен преподавателям;
- послышки – доступен преподавателям.

При подписке на оповещения о комментариях или послышках при отсутствии у пользователя подписок на курсы или разделы, пользователь подписывается на все курсы и разделы. При отписке от оповещений о комментариях и послышках пользователь, соответственно, отписывается от всех курсов и разделов.

3.4.2 Список подписок на курсы и разделы

Интерфейс настройки оповещений включает также интерактивный список курсов и разделов, на которые подписан пользователь. Данный раздел интерфейса настройки оповещений реализован так, что по нажатию на элемент списка (курс или раздел) пользователь имеет возможность отписаться от выбранного курса или раздела. Список доступен преподавателям.

4 Модуль статистики

4.1 Профиль студента

4.1.1 Персонализированные изображения профиля

4.1.2 Отображение сведений об активности

4.1.3 Результаты курсов

4.2 Рейтинг студентов

4.2.1 Подсчет общей успеваемости и точности студента

4.2.2 Рейтинг студентов по факультетам, потокам и курсам

4.3 Сведения о сложности курсов

5 Раздел помощи

5.1 Адресаты вопросов

5.2 Отображение списков вопросов и сообщений об ошибках

6 Заключение

В рамках настоящей работы был реализован модуль оповещения пользователей и модуль статистики для системы дистанционного обучения «МГУ Контест», а также усовершенствован раздел помощи пользователям системы. Таким образом была расширена функциональность системы дистанционного обучения «МГУ Контест».

7 Приложения

7.1 Модель подписки

```
class SubscriptionManager(models.Manager):
    def make_subscription(self, user, object, cascade=False):
        Course = apps.get_model('contests', 'Course')
        Contest = apps.get_model('contests', 'Contest')
        Submission = apps.get_model('contests', 'Submission')
        Subscription(object=object, user=user).save()
        if isinstance(object, Course) and cascade:
            subscribed_contest_ids = \
                Subscription.objects.for_user_model(user, Contest) \
                    .values_list('object_id', flat=True)
            new_contest_ids = \
                Contest.objects.filter(course=object) \
                    .exclude(id__in=subscribed_contest_ids) \
                    .values_list('id', flat=True)
            contest_subscriptions = (Subscription(
                object_type=ContentType.objects.get_for_model(Contest),
                object_id=contest_id, user=user
            ) for contest_id in new_contest_ids)
            Subscription.objects.bulk_create(contest_subscriptions)
        if isinstance(object, (Course, Contest)):
            if not Subscription.objects. \
                for_user_models(user, Comment, Submission).exists():
                Subscription(object_type=ContentType.objects. \
                    get_for_model(model=Comment), user=user).save()
                Subscription(object_type=ContentType.objects. \
                    get_for_model(model=Submission), user=user).save()

    def delete_subscription(self, user, object, cascade=False):
        Course = apps.get_model('contests', 'Course')
        Contest = apps.get_model('contests', 'Contest')
        Submission = apps.get_model('contests', 'Submission')
        Subscription.objects.for_user_object(user, object).delete()
        if isinstance(object, Course) and cascade:
            course_contest_ids = \
                Contest.objects.filter(course=object) \
                    .values_list('id', flat=True)
            Subscription.objects.for_user_model(user, Contest) \
                .filter(object_id__in=course_contest_ids).delete()
```

```

        if not Subscription.objects \
            .for_user_models(user, Course, Contest).exists():
            Subscription.objects \
                .for_user_models(user, Comment, Submission).delete()

class SubscriptionQuerySet(models.QuerySet):
    def for_user_model(self, user, model):
        return self.filter(user=user,
            object_type=ContentType.objects.get_for_model(model))

    def for_user_models(self, user, *models):
        return self.filter(user=user,
            object_type__in=ContentType.objects \
                .get_for_models(*models).values())

    def for_user_object(self, user, object):
        return self.filter(user=user,
            object_type=ContentType.objects \
                .get_for_model(object._meta.model),
            object_id=object.id)

class Subscription(models.Model):
    object_type = models.ForeignKey(ContentType, on_delete=models.CASCADE)
    object_id = models.PositiveIntegerField(null=True, blank=True)
    object = GenericForeignKey(ct_field='object_type')
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    objects = SubscriptionManager.from_queryset(SubscriptionQuerySet)()

    class Meta:
        verbose_name = "Подписка"
        verbose_name_plural = "Подписки"
        unique_together = ('user', 'object_type', 'object_id')

    def validate_unique(self, exclude=None):
        if Subscription.objects.exclude(id=self.id).filter(
            user=self.user,
            object_type=self.object_type,
            object_id__isnull=True).exists():
            raise ValidationError("Подписка с такими значениями полей \
                User, Object type и Object id уже существует.")
        super().validate_unique(exclude)

```

```
def __str__(self):  
    return '%s, %s%s' % (self.user.account.get_full_name(),  
        self.object_type.model,  
        ': ' + self.object.title if self.object_id is not None else '')
```

8 Список использованных источников и литературы

- [1] Моделирование процесса обучения / В. Б. Кудрявцев, П. А. Алисейчик, К. Вашик, Ж. Кнап, А. С. Строгалов, С. Г. Шеховцов // Интеллектуальные системы. – 2006, т. 10, вып. 1-4, стр. 189-270.
- [2] О дистанционном образовании – пример реализации и перспективы / П. А. Алисейчик, А. С. Строгалов, Р. А. Бекташев // Интеллектуальные системы. Теория и приложения. – 2016, т. 20, вып. 3, стр. 127-133.
- [3] Репозиторий проекта «МГУ Контест»
<https://github.com/ruslanbektashev/contest>
- [4] Документация фреймворка Django
<https://docs.djangoproject.com/en/3.0>
- [5] Документация языка Python
<https://docs.python.org/3>
- [6] Документация инструмента Chart.js
<https://www.chartjs.org/docs/latest/>