

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ПРОГРАММНЫХ КОМПОНЕНТ ДЛЯ СИСТЕМЫ
ДИСТАНЦИОННОГО ОБУЧЕНИЯ IPSILONUNI**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Пикляева Михаила Романовича

Научный руководитель
доцент, к. ф.-м. н.

А. С. Иванова

Заведующий кафедрой
к. ф.-м. н., доцент

С. В. Миронов

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Анализ задачи	5
1.1 Функционал ленты новостей	5
1.2 План улучшения	5
2 Инструментарий разработки	6
2.1 Фреймворк Ruby on Rails	6
2.2 Шаблон проектирования MVC	6
2.3 jQuery и AJAX	6
2.4 HTML Slim	7
2.5 PostgreSQL	7
2.6 RSpec	7
2.7 Docker	7
3 Процесс разработки	8
3.1 Анализ процесса создания новости	8
3.2 Развертывание	8
3.3 Способы отладки кода	9
3.4 Создание контроллера и представления	9
3.5 Создание таблицы получателей новостей	9
3.6 Обновление вывода новостей	10
3.7 Интеграция	10
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	13

ВВЕДЕНИЕ

Актуальность темы. Современное высшее образование невозможно представить без информационных технологий. Профессии, связанные с умственным трудом, в наше время практически подразумевают под собой умение работать с информацией. Многие дисциплины (в том числе и не технической направленности) прибегают к помощи цифровых технологий в процессе обучения, будь то наличие творческих заданий в формате презентаций, онлайн тесты или автоматические средства проверки заданий.

Даже от студентов, только поступающих на первый курс, ожидается грамотность в использовании ресурсов в сети интернет. С другой стороны, переход от традиционных методик обучения к цифровым не всегда легко осуществить. Во многом за счет малого количества подходящих инструментов.

В последние годы одним из основных инструментов, используемых в Саратовском государственном университете как преподавателями, так и студентами, является образовательный портал «Система дистанционного обучения IpsilonUni». Ipsilon помогает перевести в цифровую среду процесс контроля учебными планами и образовательной траектории обучающихся. Также он позволяет фиксировать ход образовательного процесса с использованием балльно-рейтинговой системы, работать с учебно-методическими курсами и тестами и помогает обучающимся формировать портфолио.

Разработка системы дистанционного образования на платформе Ipsilon ведется с 2008 года сотрудниками Института электронного и дистанционного обучения СГУ [1].

Текущая мировая эпидемиологическая обстановка подняла актуальность использования системы дистанционного обучения. Существование специальной платформы, на которой студенты могут поддерживать связь с преподавателями и получать информацию об учебном процессе, позволило преодолеть сложности, возникшие в результате резкого перехода от очного к дистанционному образованию.

В то же время актуальность модернизации системы Ipsilon также повысилась. С ростом активности использования портала возникла и потребность в повышении эффективности и удобства работы пользователей с ним.

Система Ipsilon включает в себя множество независимых компонент. В частности, можно выделить компоненту новостной ленты, которая стала

неотъемлемой частью обязательного дистанционного обучения. Ipsilon предоставляет преподавателям возможность оповещать студентов о различных событиях внутри дисциплины с помощью встроенных новостей.

Цель бакалаврской работы — улучшение пользовательского опыта при взаимодействии с лентой новостей. Поставленная цель определила **следующие задачи**:

- Анализ сценариев использования компоненты новостной ленты и выявление потенциальных вариантов ее улучшения.
- Ознакомление со внутренней архитектурой портала Ipsilon, а также с основными используемыми инструментами его разработки: Ruby on Rails, jQuery, PostgreSQL,RSpec, Docker, Gitlab.
- Реализация найденных улучшений и интеграция их в систему портала.

Практическая значимость бакалаврской работы. Практическая значимость работы заключается в повышении удобства работы преподавателей с платформой Ipsilon.

Структура и объём работы. Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы — 44 страницы, из них 37 страниц — основное содержание, включая 12 рисунков и 18 листингов кода, цифровой носитель в качестве приложения, список использованных источников информации — 22 наименования.

1 Анализ задачи

1.1 Функционал ленты новостей

Основной функционал портала, на котором было сконцентрировано внимание, — это лента новостей. Его возможности заключаются в следующем.

У каждого пользователя сайта есть доступ к разделу «Новости». На главной странице портала, а также на странице новостей, отображается список новостей, доступных пользователю.

Новости в ленту могут публиковаться двумя способами:

- Администратор может отправить новость с использованием фильтра получателей.
- Любой из преподавателей конкретной дисциплины в семестре может отправить новость всем группам, которые обучаются данной дисциплине.

1.2 План улучшения

Рассмотрим сценарии использования новостей преподавателем. Преподаватель может сообщить всем группам особенности проведения зачета или экзамена по предмету. В рамках дистанционного обучения преподаватели также оповещали студентов о предстоящих видеоконференциях.

Нередка ситуация, когда группа или группы, изучающие один предмет, разделяются на подгруппы. Возникает проблема, что подгруппы могут заниматься как в разное время, так и у разных преподавателей. Тогда имеющегося функционала портала для публикации новостей становится недостаточно для комфортной работы преподавателей. В связи с этим была поставлена задача улучшить фильтры публикации новостей. В частности, позволить преподавателю выбирать список студентов из групп данной дисциплины.

Для реализации нового функционала потребовалось предварительно ознакомиться с используемым инструментарием.

2 Инструментарий разработки

2.1 Фреймворк Ruby on Rails

Основным инструментом в разработке системы Ipsilon является язык Ruby с фреймворком Ruby on Rails.

Ruby — динамический и высокоуровневый язык, лаконичность которого дает преимущество по сравнению с другими языками, предлагающими веб-фреймворки. Ruby on Rails является веб-ориентированным фреймворком с открытым исходным кодом [2]. Отличительной особенностью Rails разработчики называют тот факт, что почти всегда есть «лучший» способ реализовать задуманное. Фреймворк так спроектирован, чтобы поощрять реализацию схожих идей одинаковым способом. Поэтому можно выделить общие шаблоны разработки на Rails, и если их изучить, то код на Rails будет легко как писать, так и читать [3] [4].

2.2 Шаблон проектирования MVC

Фреймворк Rails следует шаблону проектирования MVC. MVC (Model-View-Controller, Модель-Представление-Контроллер) — это шаблон проектирования, используемый для разработки пользовательских интерфейсов, а также логики управления данными. Акцент делается на разделение бизнес логики приложения и отображения пользователю. Три основных части шаблона проектирования MVC можно описать так [5]:

- Модель определяет, какую информацию может содержать приложение.
- Представление управляет «внешним видом» приложения: планировкой и отображением.
- Контроллер направляет команды моделям и частям представлений.

2.3 jQuery и AJAX

Скриптовые возможности JavaScript используются в проекте Ipsilon для связи между представлениями и контроллерами в интерактивном режиме. Хотя представления в Rails и используют Ruby вставки в HTML код, ERB не предполагает интерактивности на странице в браузере. Примеры использования JavaScript можно обнаружить в форме создания новости от лица администратора. На странице есть поля для выбора направлений или групп. Поиск внутри этих полей реализован с помощью jQuery, а также библиотеки select2 на основе jQuery. Запросы на поиск отправляются с помощью AJAX на сервер, где

контроллер может их обработать и вернуть результаты в представление.

2.4 HTML Slim

HTML формы в проекте Ipsilon написаны с использованием специального шаблонизатора. Шаблонизатор — программное обеспечение, позволяющее использовать HTML-шаблоны для генерации конечных HTML-страниц [6]. Основная цель использования шаблонизаторов — это отделение представления данных от исполняемого кода. В данном проекте для упрощения написания HTML представлений используется шаблонизатор Slim.

2.5 PostgreSQL

В проекте Ipsilon существует множество экземпляров объектов, для хранения и обработки которых требуется база данных. Более того, особенности связей между объектами, а также типы хранимых данных накладывают определенные требования на выбор СУБД. Данным требованиям наиболее удовлетворяет СУБД PostgreSQL.

В проекте Ipsilon для изменения схемы базы данных используются миграции — удобный и надежный способ обновления структуры базы данных от одной версии до другой. В Rails реализован так называемый «метод инкрементных изменений» — каждой миграции присваивается временная метка ее создания, а автоматический скрипт позволяет выполнить все миграции после последней ранее примененной [7].

2.6 RSpec

Для автоматического тестирования кода в проекте Ipsilon используется библиотека RSpec, созданная для behavior-driven разработки тестов (разработка через поведение) [8]. Тесты в RSpec — это фактически инструкции описанные на обычном английском языке [9].

2.7 Docker

Развертывание портала происходит с помощью инструмента Docker — платформы для контейнеризации с открытым исходным кодом. Docker позволяет разработчикам упаковывать приложения в контейнеры — стандартизированные компоненты, которые объединяют исходные коды приложения со всеми библиотеками и зависимостями операционной системы, чтобы запускать код в любой среде [10] [11].

3 Процесс разработки

3.1 Анализ процесса создания новости

В соответствии с текущим процессом создания или редактирования новости можно выделить этапы разработки. Перед отрисовкой формы необходимо получить в контроллере не только список групп, но и список их студентов. Далее в форме необходимо добавить форму выбора студентов. Список студентов-получателей после отправки необходимо сохранить в базе данных.

После формализации требований и анализа архитектурных изменений приложения список подзадач выглядит следующим образом:

- Создать контроллер, позволяющий получить список студентов по списку групп.
- Подготовить представление, отвечающее за создание и редактирование новости — отобразить форму выбора студентов в стиле checkbox списка.
- Создать таблицу в базе данных для хранения списка студентов, которым предназначена новость.
- Обновить контроллер и представление, отвечающие за получение актуальных новостей для студентов.
- Добавить сохранение списка получателей каждой новости в базу данных.

3.2 Развертывание

Начало процесса разработки включает в себя конфигурацию среды для внесения изменений в код приложения и легкого доступа к просмотру результатов изменений. Для этого необходимо развернуть локальную версию приложения.

Развертывание портала Ipsilon на локальном устройстве сильно облегчено за счет наличия Docker контейнера. Под операционной системой на основе Linux для работы с контейнерами используется инструмент `docker-compose`. При помощи двух команд можно получить полностью готовое к разработке приложение. Команда `docker-compose run` запускает контейнер и предоставляет к нему внутренний доступ. Команда `docker-compose up` запускает все необходимые сервисы, включая серверную часть.

После этого доступ к локальной версии портала Ipsilon можно получить, обратившись в любом браузере по адресу `localhost:5000`.

3.3 Способы отладки кода

Перед тем, как комментировать рабочий исходный код, стоит уделить внимание тому, как его получить. Отладка кода занимает значительную часть времени при разработке. И чем сложнее код, тем больше усилий требуется при исправлении всех ошибок.

Самый важный инструмент при отладке кода — это возможность просматривать журнал ошибок. В проекте Ipsilon существует три возможных канала вывода предупреждений и ошибок: специальный интерфейс в окне браузера для ошибок в Ruby, консоль разработчика в браузере для ошибок в JavaScript и журнал ошибок и информационных сообщений для серверных операций.

3.4 Создание контроллера и представления

Первой подзадачей было создание действия контроллера, при помощи которого можно по списку групп получить список объектов студентов из всех групп. По способу использования данной функции было решено добавить в нее наполнение двух списков. В аргументы функции приходит два списка идентификаторов: список получателей новости и список групп. По списку идентификаторов получаются списки объектов.

Представление для результата этой функции — это частичный шаблон формы. В него входит список студентов в формате checkbox. Во внешнем уровне содержится список групп, принадлежащий данному семестру дисциплины. При нажатии на группу разворачивается список ее студентов. Для удобства также реализована кнопка «Выбрать всех», которая переключает состояние выбора студентов текущей группы либо на «все выбраны», либо на «не выбран ни один».

Данная частичная форма создается после загрузки страницы создания и редактирования новости. Для этого был реализован AJAX запрос к функции `load_students`, результат которого подменяет собой внутренний HTML на странице.

3.5 Создание таблицы получателей новостей

Тип связей между новостями и студентами — так называемый «многие-ко-многим». Каждая новость может иметь множество получателей. С другой

стороны, каждый студент может быть получателем множества новостей. Такой тип связей принято реализовывать следующим образом. Создается новая таблица, в каждой строке которой записывается пара: идентификатор объекта из первой таблицы и идентификатор объекта из второй таблицы. В качестве первой таблицы служит таблица `newsfeed_news`, каждый объект которой — это одна новость. В качестве второй таблицы выбрана таблица со всеми пользователями. На то, является ли пользователь студентом, указывает выставленная ему роль.

Также была создана модель `Newsfeed::Receiver` для хранения связи между новостями и получателями.

3.6 Обновление вывода новостей

Для получения старого и нового списков получателей были реализованы вспомогательные функции `new_receivers` и `old_receivers`. Функционал `new_receivers` зависит от роли автора новости — администратор и преподаватель реализованы по-разному ввиду особенностей фильтров.

Выбор подходящих групп сводится к последовательным запросам по каждому фильтру и объединению результатов.

Если происходит создание новости, то список новых получателей передается вместе с параметрами новости в функцию для создания объекта новости и записи его в базу данных. В таблицу новостей добавляется сама новость, а в таблицу связей новость-студент добавляются все получатели. Если происходит редактирование новости, то во вспомогательную функцию передается как прежний список получателей новости, так и список новых получателей. Внутри происходит формирование двух новых списков: списки связей новость-студент, подлежащие удалению из таблицы и добавлению в таблицу.

В обоих случаях все взаимодействия с базой данных обернуты внутрь транзакции. Транзакция позволяет объединить несколько операций с базой данных в одну атомарную. Таким образом, если хотя бы одна из операций не успешна, то все операции будут отменены.

3.7 Интеграция

В разработке приложения системы `Ipsilon` применяется стратегия `Continuous Integration and Continuous Delivery (CI/CD)`, непрерывная интеграция и непрерывное развертывание). Непрерывная интеграция подразумевает авто-

матический запуск конвейера скриптов для построения, тестирования и валидации изменений кода при каждом добавлении небольших частей кода в кодовую базу приложения, но до объединения его с главной веткой. Непрерывное развертывание предполагает шаг, следующий за CI, — автоматическое развертывание приложения на месте при каждом изменении главной ветки репозитория [12].

Автоматизация CI/CD предоставляется встроенными возможностями платформы GitLab, на которой базируется система Ipsilon. Инструментами непрерывной интеграции служат два основных компонента:

- RSpec — инструмент для behavior-driven разработки тестов.
- Rubocop — статический анализатор кода на языке Ruby.

Первоначальный статический анализ кода помог выявить ошибки в стиле оформления и сложности выполнения некоторых функций. Ошибки были исправлены так, чтобы код соответствовал заявленным требованиям приложения.

На CD-диске, прикрепленном к работе, предоставлен исходный код приложения с проделанными изменениями, а также полный список проделанных изменений в исходном коде в формате вывода команды `git diff`.

ЗАКЛЮЧЕНИЕ

В рамках данной работы были произведены изменения в компоненте новостной ленты портала IpsilonUni, направленные на улучшение пользовательского опыта. Данные изменения включают в себя как переработку пользовательского интерфейса, так и внутреннего устройства системы: добавлена форма выбора отдельных студентов при отправке новости, а также инфраструктура для поддержки данного функционала.

Также было проведено ознакомление с архитектурой проекта системы Ipsilon и получен опыт в работе с сопутствующими инструментами: Ruby on Rails, jQuery, RSpec, Docker.

Данные изменения могут послужить основой дополнительным опциям новостной ленты. Например, потенциальной задачей является возможность сохранения списка студентов-получателей внутри дисциплины или преподавателя для использования в последующих новостях. Актуальность задачи становится понятна при исследовании способов использования новостей. В частности, отдельный список студентов — это скорее всего подгруппа внутри дисциплины, которая не изменяется на протяжении семестра.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Институт электронного и дистанционного обучения | СГУ - Саратовский государственный университет [Электронный ресурс].— URL: <https://www.sgu.ru/structure/openedu> (Дата обращения 10.04.2021). Загл. с экр. Яз. рус.
- 2 Ruby on Rails | A web-application framework that includes everything [Электронный ресурс].— URL: <https://rubyonrails.org/> (Дата обращения 10.04.2021). Загл. с экр. Яз. англ.
- 3 Getting Started with Rails [Электронный ресурс].— URL: https://guides.rubyonrails.org/getting_started.html (Дата обращения 10.04.2021). Загл. с экр. Яз. англ.
- 4 *Фернандес, О.* Путь Rails. Подробное руководство по созданию приложений в среде Ruby on Rails / О. Фернандес.— Санкт-Петербург: Символ-Плюс, 2008.
- 5 Ruby on Rails github repository [Электронный ресурс].— URL: <https://github.com/rails/rails> (Дата обращения 10.04.2021). Загл. с экр. Яз. англ.
- 6 Slim [Электронный ресурс].— URL: <http://slim-lang.com/about.html> (Дата обращения 12.04.2021). Загл. с экр. Яз. англ.
- 7 RailsGuide - Active Record Migrations [Электронный ресурс].— URL: https://edgeguides.rubyonrails.org/active_record_migrations.html (Дата обращения 15.04.2021). Загл. с экр. Яз. англ.
- 8 About RSpec [Электронный ресурс].— URL: <https://rspec.info/about/> (Дата обращения 20.04.2021). Загл. с экр. Яз. англ.
- 9 RSpec github repository [Электронный ресурс].— URL: <https://github.com/rspec/rspec-rails> (Дата обращения 20.04.2021). Загл. с экр. Яз. англ.
- 10 What is Docker? | IBM [Электронный ресурс].— URL: <https://www.ibm.com/cloud/learn/docker> (Дата обращения 23.04.2021). Загл. с экр. Яз. англ.

- 11 *Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл. — Москва: ДМК Пресс, 2020.*
- 12 GitLab CI/CD [Электронный ресурс]. — URL: <https://docs.gitlab.com/ee/ci/> (Дата обращения 03.05.2021). Загл. с экр. Яз. англ.