

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

Кафедра математической кибернетики и компьютерных наук

**ИНТЕГРАЦИЯ IPSILONUNI С ЭЛЕКТРОННОЙ БИБЛИОТЕКОЙ**  
**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студентки 4 курса 411 группы  
направления 02.03.02 — Фундаментальная информатика и  
информационные технологии  
факультета компьютерных наук и информационных технологий  
Гношевой Алены Александровны

Научный руководитель:

доцент, к. ф.-м. н. \_\_\_\_\_

И. А. Батраева

подпись, дата

Зав. кафедрой:

к. ф.-м. н. \_\_\_\_\_

С. В. Миронов

подпись, дата

Саратов 2022

## **ВВЕДЕНИЕ**

### **Актуальность темы.**

Разработка нового модуля в платформе IpsilonUni для создания списка учебно-методического обеспечения дисциплин (практик) является актуальной, так как сможет облегчить и ускорить процесс создания, проверки, редактирования и утверждения списка рекомендуемой литературы для рабочей программы.

### **Цель бакалаврской работы.**

Интеграция системы дистанционного обучения IpsilonUni с электронной библиотекой СГУ.

Поставленная цель определила **следующие задачи**:

- 1) изучить язык Ruby on Rails;
- 2) научиться проектировать Rails - приложения;
- 3) освоить архитектуры MVC и REST, применить их на практике;
- 4) изучить внутреннее устройство платформы IpsilonUni, понять структуру и логику его работы;
- 5) разработать новый принцип взаимодействия с учебно-методическим обеспечением рабочей программы;
- 6) организовать интеграцию системы IpsilonUni с электронной библиотекой СГУ;

### **Практическая значимость бакалаврской работы.**

Одним из параметров аккредитации вуза является показатель обеспеченности учебного процесса учебниками. При составлении рабочих программ преподавателю необходимо предоставить список рекомендуемых учебников. Вуз должен обеспечить всех обучающихся доступом к соответствующим электронным учебникам или предоставить необходимое число печатных учебников. Как правило, при аккредитации этот раздел проверяется. Доступ к электронным учебникам проверяется сотрудником библиотеки непосредственно по каждой позиции, наличие печатных изданий и их количество проверяется по каталогу библиотеки. Именно поэтому добавление в платформу IpsilonUni нового модуля создания списка литературы является практически значимым, так как сможет облегчить и ускорить процесс создания и утверждения списка литературы рабочей программы.

**Структура и объём работы.** Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и четырех приложений. Общий объем работы – тридцать девять страниц, из них тридцать одна страница – основное содержание, включая восемнадцать рисунков, список использованных источников информации – двадцать наименований.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

**Первый раздел «Теоретическая часть»** посвящен описанию используемых технологий для разработки.

Приведено подробное теоретическое изложение понятий гибкой разработки веб-приложений, архитектуры MVC (Model-View-Controller), Rails-приложений, миграций, REST (Representational State Transfer), AJAX и описание микроструктуры Slim.

Rails — это веб-фреймворк написанный на языке программирования Ruby, в свою же очередь фреймворк - это набор приложений и различных средств для создания веб-приложения. Все моменты взаимодействия системы фреймворка очень продуманы и структурированы. Ruby on Rails является средой, облегчающей разработку, развертывание и обслуживание веб-приложений.

Все Rails-приложения выполняются с использованием архитектуры Модель-Представление-Контроллер (Model-View-Controller, MVC).

Модель отвечает за поддержку состояния приложения. Модель — это не просто данные, в ней прописаны все бизнес-правила, применяемые к этим данным.

Представление отвечает за формирование пользовательского интерфейса, который обычно основан на данных модели.

Контроллеры организуют работу приложения. Они воспринимают события внешнего мира (обычно ввод данных пользователем), взаимодействуют с моделью и отображают соответствующее представление для пользователя.

Стиль RESTful разработки помогает разработчикам Rails-приложений решить, какие контроллеры и методы необходимы: достаточно структурировать приложение, используя ресурсы, которые могут создаваться, извлекаться, изменяться и удаляться.

Технология Ajax переопределила базовую модель использования браузеров, заключающуюся в единовременной визуализации страницы. Ajax дала возможность браузеру взаимодействовать с сервером в промежутках между обновлениями страницы, что дает большее удобство для пользователя, хотя и за счет повышения сложности. Ajax передает XML между сервером и клиентом, используя клиентские

JavaScript-библиотеки. Ajax-разработчики могут посылать асинхронные запросы от клиента в любое время, а пользователь может продолжать свою работу в браузере во время обработки запросов.

Миграции в Rails помогают развивать схему базы данных приложения без уничтожения и нового создания базы после каждого изменения. Означает это то, что появившиеся в процессе разработки данные – не теряются. При выполнении миграции достаточно описать изменения, необходимые для перехода от одной версии схемы к другой – следующей или предыдущей.

Slim — это быстрый и легкий шаблонизатор с поддержкой Rails 3 и 4. Это микроструктура РНР, которая помогает писать веб-приложения и API.

В теоретической части описаны все используемые технологии и приведены все необходимые предварительные требования для практической реализации.

**Второй раздел «Практическая часть»** посвящен разработке новой функциональности для платформы системы дистанционного обучения IpsilonUni.

На сайте Ipsilon есть множество возможностей — работа с учебными планами и образовательной траекторией обучающегося, фиксация хода образовательного процесса (балльно-рейтинговая система), работа с учебно-методическими курсами и тестами, формирование портфолио обучающегося и система дистанционного обучения .

В первую очередь приводится обзор рабочих программ в системы IpsilonUni, для того, чтобы описать логику работы системы с учебно-методическим обеспечением дисциплин (практик). В настоящий момент времени создания и редактирование компонентов структуры рабочей программы состоит из взаимодействия с текстовым редактором.

Основной задачей практической работы являлось избавление от взаимодействия с текстовым поле в разделе с учебно-методическим обеспечением путем добавления новой функциональности, содержащей в себе интеграцию с библиотекой.

Для решения поставленной задачи, первоначально было проведенного несколько встреч с Крутихином Игорем Валерьевичем, заместителем директора ЗНБ СГУ. Целью общения было выявления трудностей со стороны сотрудников библиотеки при составлении учебно-методических обеспечений дисциплин (практик) для рабочих

программ университета. Результатом стало выявление следующей проблемы – при проверке составленного списка учебных материалов сотрудникам библиотеки необходим быстрый и легкий способ проверки наличия и количества необходимых методических обеспечений.

Решение было предложено следующее: так как у сотрудников библиотеки есть аккаунт в IpsilonUni, для их работы на сайте предусмотрена роль «Администратор библиотеки», то при просмотре составленного списка учебных материалов для рабочей программы, отдельным полем отображать ссылку, при нажатии на которую, сотрудник перенаправляется на страницу с учебным материалом на сайте библиотеки. После проверки наличия нужного количества экземпляров учебников, сотруднику необходимо предоставить возможность оставить комментарий и выбрать статус для учебного материала. Такое решение упрощает работу сотрудника библиотеки и преподавателя, сокращая временной процесс утверждения рабочей программы.

Согласовав предложенное решение с Крутихиным Игорем Валерьевичем, следующим этапом было обсуждение нового модуля в рабочих программах с сотрудниками со стороны сервиса IpsilonUni. Итогом всех обсуждений стало полное постановление задачи и описанные ниже этапы разработки для их решения.

В практической части работы подробно описаны этапы разработки:

1. Настройка программного обеспечения.
2. Обзор нового модуля рабочих программ в системе IpsilonUni.
3. Изменение структуры базы данных, добавление в базу данных сервиса IpsilonUni новых таблиц, отвечающих за хранение информации об учебно-методических пособиях.
4. Добавление функциональности для доступа к внешнему серверу, необходимой для организации интеграции с электронной библиотекой СГУ.
5. Описание функций, отвечающих за добавление учебных материалов из базы данных библиотеки СГУ на стороне сервиса IpsilonUni.
6. Добавление нового визуального представления, а именно удобного и понятного интерфейса для составления рабочей программы в приложении IpsilonUni.

Для работы с приложением IpsilonUni была установлена операционная система Ubuntu версии 20.0.

Установка Ruby, Rails и всего сопутствующего программного обеспечения поддержки была осуществлена с помощью Docker Compose. Сделано это было для того, чтобы разрешить сложную задачу, созданную большим разнообразием окружений: разные операционные системы, версии.

Для того, чтобы организовать интеграцию приложения Ipsilon с системой автоматизации библиотек ИРБИС, первоначально были написаны функции для доступа к внешним сервисам. Для решения этой задачи использовался класс Net::HTTP, который предоставляет обширную библиотеку для создания пользовательских HTTP-клиентов. Протокол передачи гипертекста ( HTTP ) — это прикладной протокол для распределенных, совместных информационных систем гипермедиа. HTTP является основой передачи данных для World Wide Web.

Новый разработанный интерфейс представляет из себя следующее — пользователь, при переходе в раздел с учебно-методическим обеспечением в рабочей программе будет видеть два текстовых поля. Первое для поиска необходимого учебника в базе данных библиотеки СГУ с кнопкой автоматического добавления библиографического описания пособия в список учебных материалов. Второе, при желании, для самостоятельного ввода информации о методическом обеспечении. У второго текстового поля так же имеется кнопка добавления введенного описания в таблицу со списком учебников.

Интерфейс содержит в себе два элемента «Добавить материал из ИРБИС» и «Добавить материал вручную». При нажатии на любой из выбранных, пользователю открывается форма, предлагающая добавить новую позицию в список учебных материалов. Для удобства взаимодействия с полем ввода, на нем есть текст, который объясняет пользователю, что данное поле предназначено для поиска учебного материала из базы данных библиотеки или для ввода описания вручную. Стоит отметить, что при реализации поиска была задействована технология AJAX. При её использовании нет необходимости каждый раз обновлять страницу, так как она использует динамическое обращение к серверу при отправке запроса.

Таблица имеет несколько столбцов, при просмотре которой пользователь будет видеть количество учебников в списке, описание, ссылку на внешний ресурс библиотеки, комментарий и статус. У пользователя с ролью «Преподаватель» не будет возможности изменить

статус и комментарий в составленном списке, такая возможность предоставлена пользователям с ролью «Администратор библиотеки».

В отдельном разделе наглядно продемонстрирована работа итогового проекта. Преподавателю предоставлен удобный и понятный интерфейс для составления рабочей программы, для этого добавление учебных материалов из базы данных библиотеки производится на стороне сервиса Ipsilon. Сотрудник библиотеки при просмотре составленного списка имеет возможность одобрить или нет предложенную книгу и оставлять комментарии к каждому учебному материалу из списка.

## **ЗАКЛЮЧЕНИЕ**

Результатом практической работы стала готовая к внедрению и использованию новая функциональность для работы с учебно-методическим и информационным обеспечением дисциплины (практик). Разработан модуль для интеграции системы дистанционного обучения IpsilonUni с библиотекой СГУ. Благодаря чему у преподавателей и сотрудников библиотеки теперь упрощен процесс взаимодействия и сокращено время для утверждения списка рекомендуемой литературы рабочей программы.

Для успешного разрешения поставленной задачи был изучен Ruby on Rails фреймворк, архитектурный шаблон MVC и CRUD - методология.

### **Основные источники информации:**

1. Руби, С. Rails 4. Гибкая разработка веб-приложений / С. Руби, Д. Томас, Д. Хэннсон. — ООО Издательство «Питер», 2014. — С. 448.
2. Хартл, М. Ruby on Rails для начинающих / М. Хартл. — Москва: ДМК, 2017. — С. 57.
3. Фултон, Х. Путь Ruby. Третье издание / Х. Фултон. — Москва: ДМК, 2015. — С. 657.
4. Макгаврен, Д. Head First. Изучаем Ruby / Д. Макгаврен. — Санкт-Петербург – Москва: Питер, 2016. — С. 544.
5. Фернандес, О. Путь Rails. Подробное руководство по созданию приложений в среде Ruby on Rails. Оби Фернандес / О. Фернандес. — Санкт-Петербург – Москва: ИМВО, 2009. — С. 768.
6. Метц, С. Ruby. Объектно-ориентированное проектирование / С. Метц. — Санкт-Петербург: Питер, 2017. — С. 304.