

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Рубцовский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования
«Алтайский государственный университет»

Кафедра Математики и прикладной информатики

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)**

Тема: Разработка программного интерфейса интеграции
программных приложений с корпоративной информационной
системой Рубцовского института (филиала) АлтГУ

Выпускную квалификационную
работу (бакалаврскую работу)
выполнил студент
4 курса, группы 1245
Костенко В.В.

(подпись)

Научный руководитель:
канд. техн. наук, доцент
Жданова Е.А.

(подпись)

Выпускная квалификационная
работа (бакалаврская работа)
защищена

Допустить к защите
Зав. кафедрой
канд. техн. наук, доцент
Жданова Е.А.

(подпись)

« 26 » июня 2018 г.

Оценка отлично

Председатель ГЭК
д-р техн. наук, профессор
Пятковский О.И.

(подпись)

« 18 » июня 2018 г.

Рубцовск 2018

РЕФЕРАТ

Дипломный проект: 77 страниц, 30 рисунков, 3 таблицы, 18 источников, 3 приложения.

СИСТЕМНЫЙ АНАЛИЗ, КОРПОРАТИВНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, ПРОГРАММНЫЙ ИНТЕРФЕЙС, API, REST, ZEND, .NET CORE 2, МИКРОСЕРВИСНАЯ АРХИТЕКТУРА, OAUTH2.0, UML, UML Designer.

Объект исследования – корпоративная информационная система Рубцовского института (филиала) АлтГУ.

Предмет исследования – процесс интеграции с информационной системой института корпоративных и сторонних программных приложений.

Цель исследования – разработка программного интерфейса интеграции программных приложений с корпоративной ИС Рубцовского института.

Методы решения поставленных задач: системный анализ, моделирование предметной области, объектно-ориентированная методология.

Результаты проекта: проанализированы данные о предметной области, проведен системный анализ предметной области, UML диаграмма объектов и действий корпоративной ИС, на основе объектной модели выявлены цели создания проекта, требования к нему, пользователи и разработано API.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Аналитическая часть	7
1.1 Техничко-экономическая характеристика предметной области	7
1.2 Анализ функционирования объекта исследования.....	30
1.3 Определение цели и задач проектирования API	34
1.4 Обзор и анализ существующих разработок, выбор технологии проектирования	35
1.4.1 Обзор и анализ существующих разработок	35
1.4.2 Выбор технологии проектирования информационной системы	36
1.5 Выбор и обоснование проектных решений	39
1.5.1 Выбор фреймворков, технологий разработки и технологических платформ.....	39
1.5.2 Выбор средств разработки и проектирования	43
2 Проектная часть	47
2.1 Разработка функционального обеспечения	47
2.2 Разработка информационного обеспечения	47
2.2.1 Используемые классификаторы и системы кодирования	47
2.2.2 Характеристика нормативно-справочной и входной оперативной информации.....	48
2.2.3 Характеристика результатной информации	49
2.2.4 Структура результатной информации	50
2.2.5 Информационная модель и ее описание	53
2.3 Разработка программного обеспечения	55
2.3.1 Описание серверных программных модулей	55
2.3.2 Описание клиентских программных модулей.....	58
2.3 Компьютерно-сетевое обеспечение.....	59
2.4 Обеспечение информационной безопасности.....	59
3 Оценка эффективности внедрения информационных систем	67
3.1 Общие положения.....	67
3.2 Показатели эффективности	69
3.3 Расчет экономической эффективности	71
3.4 Расчет управленческой эффективности	74
ЗАКЛЮЧЕНИЕ	75
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	76
ПРИЛОЖЕНИЕ А	78
ПРИЛОЖЕНИЕ Б.....	80
ПРИЛОЖЕНИЕ В	83

ВВЕДЕНИЕ

На современном этапе развития информационных технологий ни одна организация не может полноценно функционировать без использования различных информационных систем. Задачи применяемых систем могут начинаться от планировщиков задач для сотрудника и заканчиваться системами помощи принятия управленческих, экономических и других решений для директора организации.

В процессе развития подобных систем они стали усложняться и процесс их внедрения или разработки для организации становился сложнее и сложнее. По сути они функционировали как отдельные единицы, а это было практически не применимо к деятельности предприятий. Поэтому на определенном этапе все предприятия стали выстраивать целые корпоративные информационные системы (КИС), которые работают в комплексе друг с другом.

Но данный подход имел достаточно большую проблему для функционирования всей системы в целом:

- сильная взаимосвязанность компонентов приводила к тому что выход одной системы приводил к ошибкам в работе в других ИС;
- дублируемость функционала в системах приводила к сложности их проектирования, создания и дальнейшего поддержания в работоспособном состоянии.

Для решения этих двух проблем была разработана новая микросервисная архитектура корпоративных информационных систем.

Преимущество данной архитектуры заключается в том, что каждый сервис (микросервис) является одной атомарной и независимой единицей системы в целом, поэтому функциональность одного сервиса никак не влияла на другую. Так же такое разграничение помогло решить проблему дублирования кода (функционала), что положительно влияет на сложность поддержки систем. Самым частым проявлением данной архитектуры

является API – (программный интерфейс приложения) – набора готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом).

API – позволяют избежать чрезмерного дублирования кода в программах, а также использовать повторяющиеся модули в разных ИС и даже на разных платформах – ПК, WEB, мобильные ОС и др.

На сегодняшний день, почти каждая корпоративная ИС позволяет интегрировать в себя новые внутренние и сторонние приложения, без особых затрат на написание программного кода при помощи API.

Внедрение подобной архитектуры на базе корпоративной информационной сети Рубцовского института (филиал) АлтГУ было бы целесообразным развитием всей ИС в целом. Поэтому разработка программного интерфейса интеграции программных приложений с корпоративной информационной системой института является актуальной задачей развития КИС.

В связи с этим, объектом исследования данного проекта является корпоративная информационная система Рубцовского института (филиала) АлтГУ.

Предметом исследования является процесс интеграции с информационной системой института корпоративных и сторонних приложений.

Цель исследования – разработка программного интерфейса интеграции программных приложений с корпоративной ИС Рубцовского института.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- провести подробный анализ предметной области;
- построить объектную модель существующей организации доступа к корпоративной информационной системе института;
- выявить недостатки существующей системы и на их основании предложить объектную модель «Как должно быть»;

- разработать программный интерфейс (API);
- оценить эффективность от внедрения API.

Методы, используемые при написании проекта: системный анализ, моделирование предметной области.

1 Аналитическая часть

1.1 Техничко-экономическая характеристика предметной области

Рубцовский институт (филиал) АлтГУ – обособленное структурное подразделение Алтайского Государственного университета, расположенное в городе Рубцовск, Алтайского края и осуществляющее постоянно его функции в рамках выданной институту лицензии на образовательную деятельность.

Предприятие является государственной, некоммерческой организацией, имеет право на ведение образовательной деятельности и на льготы, установленные законодательством РФ, в соответствии с лицензией, выданной Министерством образования РФ.

Институт реализует образовательные программы полного и неполного высшего профессионального образования. Также институт обладает правом реализации образовательных программ по хозрасчетным договорам в рамках установленного перечня специальностей.

Основными функциями института являются:

- реализация образовательных программ в сфере высшего профессионального образования по очной, заочной и очно-заочной (вечерней) формам обучения по направлениям подготовки (специальностям) Алтайского Государственного Университета в соответствии с лицензией, выданной институту;
- организация учебного процесса в институте по образовательным программам высшего профессионального образования регламентируется учебным планом по направлению подготовки специальности и расписанием учебных занятий для каждой формы обучения, которые разрабатываются институтом самостоятельно и утверждаются на основе государственного

образовательного стандарта высшего профессионального образования, примерных образовательных программ, учебных планов по направлению подготовки специальности и программам дисциплин, утверждаемых федеральным органом управления образованием, при этом примерный учебный план и программы дисциплин имеют рекомендательный характер;

- реализация программ дополнительного образования (по полной и ускоренной программам) на базе очного или заочного обучения;

- осуществление гибких программ для курсов переподготовки и повышения квалификации работников предприятий, организаций, учреждений и органов административного управления городов и поселков юга Алтайского края;

- разработка и внедрение инновационных технологий обучения;

- исследования в области предпринимательской деятельности, экономики, региональных и муниципальных реформ.

Содержание образования в институте ориентировано на:

- обеспечение самоопределения личности, создание условий для ее саморазвития;

- развитие гражданского общества;

- укрепление и совершенствование правового государства.

Профессиональное образование в институте обеспечивает получение обучающимися, профессий соответствующей их квалификации. При этом гарантируется государственный образовательный стандарт.

Предприятие было создано для удовлетворения потребности в высококвалифицированных специалистах и для повышения уровня образования в регионе. В соответствии с заданными целями, предприятие вправе осуществлять следующие виды деятельности:

- предоставление услуг высшего образования;

- проведение курсов повышения квалификации;

- оказание платных библиотечных услуг;

- услуги передачи данных, информационные услуги, в том числе на основе организуемых компьютерных классов открытого доступа;
- услуги по разработке, производству и поставке программных продуктов, информационных систем, программных и методических средств;
- сервисных работ и услуг по техническому обслуживанию средств вычислительной техники, оргтехники;
- организация и проведение физкультурно-оздоровительных и спортивных мероприятий, создание спортивных секций;
- организация и предоставление культурно-бытовых услуг.

Финансирование деятельности института осуществляется в установленном порядке за счет средств:

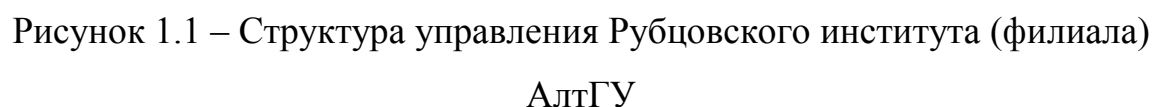
- федерального бюджета;
- полученных от выполнения хозяйственных договоров;
- полученных от подготовки студентов по хозрасчетным договорам;
- спонсоров, добровольных взносов, пожертвований граждан, организаций;
- из других законных источников.

Организационная структура – совокупность звеньев, расположенных в строгой соподчиненности и обеспечивающих взаимосвязь между другими подразделениями организации, а также распределение между ними ответственности и прав, которая проявляется через разделение труда, создание специализированных подразделений, иерархию должностей и является необходимым элементом эффективной организации, так как придает ей внутреннюю стабильность и позволяет добиться определенного порядка в использовании ресурсов.

Организационная структура предприятия представляет собой строгую иерархию, которая представлена на рисунке 1.1.

Филиал располагается в двух учебных корпусах. Численный состав

Создана система управления всеми образовательными процессами института, которая является очень гибкой и динамичной.



Управление учебным процессом института осуществляется за счет информационных систем: «Учебные планы», «Поручения ППС», «Графики учебного процесса», «Абитуриенты», «Студенты», «Кадры», «Редактор расписания занятий», подсистемы «Анализ учебных планов в соответствии с ФГОС (система менеджмента качества», подсистемы «Анализ расписания занятий», «Паспорт компетенции», «Редактор рабочей программы учебной

дисциплины», «Редактор учебно-методического комплекса дисциплины», «Электронная научная библиотека института», «Система электронного обучения на платформе Moodle».

Все вышеперечисленные информационные системы являются исключительно разработкой института и отвечают всем предъявляемым в настоящее время требованиям к подобным ИС.

ИС «Учебные планы»

В данной системе на каждый учебный год создаются или обновляются рабочие учебные планы направлений подготовки и специальностей в соответствии с ФГОС ВО и ФГОС СПО, а также учебные планы дополнительного профессионального образования (ДПО) (рисунок 1.2).

Предварительный просмотр

План одобрен Ученым советом вуза
Протокол № _____
" ____ " _____ 20__

РАБОЧИЙ УЧЕБНЫЙ ПЛАН

УТВЕРЖДАЮ
Директор Рубцовского института (филиала) АлтГУ
К.Г. Анисимов
" ____ " _____ 20__ г.

Направление: Прикладная информатика
Форма обучения: дневная (бакалавриат)
Срок обучения: 4 года
Год поступления: 2015

№ п/п	Наименование циклов и дисциплин	Общий объем			Объем аудиторных занятий				Распределение по курсам и семестрам								Итоговые формы контроля		
		ЭЗ	Всего	Сам. раб.	Всего	Лекц.	Сам.	Лаб.	1 курс		2 курс		3 курс		4 курс		Курс.	Контр.	Экзам.
									1	2	1	2	1	2	1	2			
Б1	Дисциплины (модули)																		
Б	Дисциплины (модули) \ Базовая часть																		
Б.2.1	История	3	108	60	48	20	28		48									1.1	
Б.2.2	Иностранный язык	10	360	198	162			162	40	42	40	40						1.1	2.2
Б.2.3	Философия	3	108	44	64	26	38		64									1.2	1.1
Б.2.4	Физическая культура	2	72		72		72		36	36								1.2	
Б.2.5	Дифференциальные уравнения	5	180	114	66	32	34				66								2.1
Б.2.6	Линейная алгебра	4	144	80	64	36	28		64										1.1
Б.2.7	Математический анализ	6	216	126	90	42	48		42	48								1.1	1.2
Б.2.8	Дискретная математика	4	144	64	80	36	44		80										1.2
Б.2.9	Теория вероятностей и математическая статистика	8	288	238	50	24	26				50								2.1
Б.2.10	Теория систем и системный анализ	3	108	28	80	36		44					80					3.1	
Б.2.11	Информатика и программирование	5	180	100	80	20		60	80										1.1
Б.2.12	Физика	3	108	58	50	18	18	14		50									1.2
Б.2.13	Безопасность жизнедеятельности	2	72	40	32	16	16			32									1.2
Б.2.14	Вычислительные системы, сети и телекоммуникации	5	180	98	82	36		46	40	42							1.2	1.1	1.2
Б.2.15	Операционные системы	3	108	58	50	24		26		50									1.2

1

Рисунок 1.2 – Информационная система «Учебные планы»

В ИС «Учебный план» методисты кафедр вносят рабочие учебные планы по ВО, СПО и ДПО, после чего можно сформировать множество

отчетов: «Базовый учебный план», «План – график набора» по каждому курсу, который выдается лично на руки каждому студенту на кафедре и т.д.

После внесения учебного плана в данную информационную систему происходит проверка плана на соответствие ФГОС.

Процедура проверки осуществляется в отдельной подсистеме «Анализ учебных планов в соответствии с ФГОС (система менеджмента качества)» (рисунок 1.3), в которой внесены все требования ФГОС по направлениям подготовки ВО и специальностям.

Учебные планы ВО и СПО обновляются раз в год, а учебные планы ДПО вносятся все время в течение года по мере формирования групп или индивидуального обучения (рисунок 1.4).

Учебный план - Полушина Юлия Сергеевна

ВПО, СПО, ДПО

Год: 2013 Специальности: Государственное и муниципальное управление (ДО (Бак.), 4) 2013

Форма обучения: дневная (бакалавриат) Лет: 4 Мес.: 0 Код специальности: 081100.62 Сокр. название: ГМУ Станда

Кафедра: Кафедра «Менеджмент» Примечание: Год 2013

Учебный план Система менеджмента качества

Проверить Проверить всё Критерии Параметры Копировать критерии

Критерий	Описание	Дата проверки	Результат запроса	Результат
Требования к загрузке студентов учебным временем	критерии выражены час/неделя			
Максимальный объем учебной нагрузки студентов (аудитор. + самостоят. работа студентов)	Число часов общего теоретического обучения / Количество недель теоретического обучения (включая экзаменационные сессии)			
Не более 10 экзаменов и 12 зачетов в год	не считаем занятия по физической культуре	28.04.2014 16:15:18		Выполняется
Общий объем аудиторной нагрузки (не считаем занятия по физической культуре)	Общее количество аудиторной нагрузки - физическая культура / Количество недель теоретического обучения (включая экзаменационные сессии)	28.04.2014 16:15:16	23	Выполняется
Требования к расписанию учебных занятий				
Второстепенные методические требования к расписанию учебных занятий				
Второстепенные организационные требования к расписанию учебных занятий				
Требования к содержанию учебного плана				
Требования к срокам освоения основной образовательной программы	Критерии выражены в зачетных единицах и неделях			
Общая трудоемкость ООП	Рассчитывается в зачетных единицах как сумма трудоемкости всех циклов и разделов ООП	28.04.2014 16:25:04	240	Выполняется
Общий объем каникулярного времени в год	рассчитывается в неделях	28.04.2014 16:14:23		Выполняется
Общий объем каникулярного времени в зимний период в год	рассчитывается в неделях	28.04.2014 16:14:24		Выполняется
Трудоемкость ООП за год	в ЗЕ			
Трудоемкость ООП за курс	в ЗЕ			
Учебная и производственные практики	в неделях			

Рисунок 1.3 – Информационная система «Учебные планы» подсистема «Анализ учебных планов (система менеджмента качества)»

Название курсов	Кол-во часов	Кафедра
1С: Предприятие: 8 редакция 3.0 (1С: Бухгалтерия 8; 1С: Зарплата и Управление)	72	Кафедра "Экономики"
1С: Предприятие 8.2 (1С: Бухгалтерия 8.2., 1С: Управление торговлей 8.2., 1С: Зарплата и Управление торговлей 8.2)	72	Кафедра "Экономики"
1С: Предприятие 8.2 (1С: Управление торговлей 8.2)	16	Кафедра "Экономики"
1С: Предприятие 8.2 (1С: Бухгалтерия 8.2)	12	Кафедра "Экономики"
1С: Предприятие 8.2 (1С: Бухгалтерия 8.2., 1С: Управление торговлей 8.2., 1С: Зарплата и Управление торговлей 8.2)	72	Кафедра "Экономики"
Бухгалтерский финансовый учет	10	Кафедра "Экономики"
Изучение торговых операций и применения ККТ	2	Кафедра "Экономики"
Информационные технологии. Microsoft Excel	16	Кафедра «Математики и прикладной информатики»
Всего: 21 курсов		

Дисциплина	Всего	Лекций	Лабораторных	Семинаров	Зачёт	Экзамен
1С: Предприятие: 8 редакция 3.0 (1С: Бухгалтерия 8; 1С: Зарплата и Управление торговлей 8.2)	72		70		☐	☒

Рисунок 1.4 – Информационная система «Учебный план», вкладка ДПО

В ИС «Студенты» сформированы личные дела обучающихся, медицинская карта и т.д., а также формируется индивидуальный учебный план для каждого студента, в зависимости от базового учебного плана набора и выбранных студентом элективных дисциплин и курсовых работ.

Информационная система «Студенты» представляет собой базу, в которой хранится информация о каждом обучающемся, выпускнике, слушателе курса ДПО, отчисленном студенте, студенте в академическом отпуске. В ИС «Студенты» пользователь имеет возможность посмотреть анкетные данные студента, информацию об успеваемости, документообороте и т.д. (Рисунок 1.5, 1.6 и 1.7).

Студенты (РИ АлтГУ) Версия 3.1.0.502

Группа: **12146** Долгов: **0**

Анкетные данные | Учебный процесс | Документооборот | Безопасность | Медик

Основные сведения | Контактная информация | Документы | Военный учет | Трудоустройство

Фамилия И.О.
Агарков Сергей Сергеевич
Букреев Юрий Александрович
Геберт Андрей Иванович
Кимосова Дарья Валерьевна
Крец Яна Эдуардовна
Лысенко Денис Дмитриевич
Нефёдова Лада Евгеньевна
Носкова Анна Васильевна
Пак Виктория Альбертовна
Пивцайкин Владислав Владимирович
Плетнева Ирина Алексеевна
Попов Андрей Аркадьевич
Располов Роман Викторович
Сизиков Алексей Андреевич
Тюляева Вера Сергеевна
Федякина Ксения Сергеевна
Цуканова Екатерина Витальевна
Шилиева Анастасия Александровна
Юрченко Мария Александровна

Расшифровка набора
Факультет
Экономический
Специальность
вентное и муниципальное управление (дневная (бакалавриат))
Форма обучения
дневная (бакалавриат)
Год поступ.
2011
Лет обучения
4
Задолженность по оплате
0
☐ Является старостой
☐ Общежитие

Фамилия
Цуканова
Имя
Екатерина
Отчество
Витальевна
Родился
05.10.1993 20 Женский
Группа
12146
План набора
ГМУ (ДО (Бак.), 4) 2011
Вид набора
ПН
Статус
Обучается
Профиль "Муниципальное управление"
Ин. язык
Английский язык
Физ. группа
Физическая культура для обычных групп
Студ. билет
12146/19
Куратор
Шихалева Наталья Юрьевна
Льгота
Малообеспеченная семья
☐ Обучается индивидуально
☐ Свободное посещение

Выбрано 19 Save All

Рисунок 1.5 – Информационная система «Студенты»

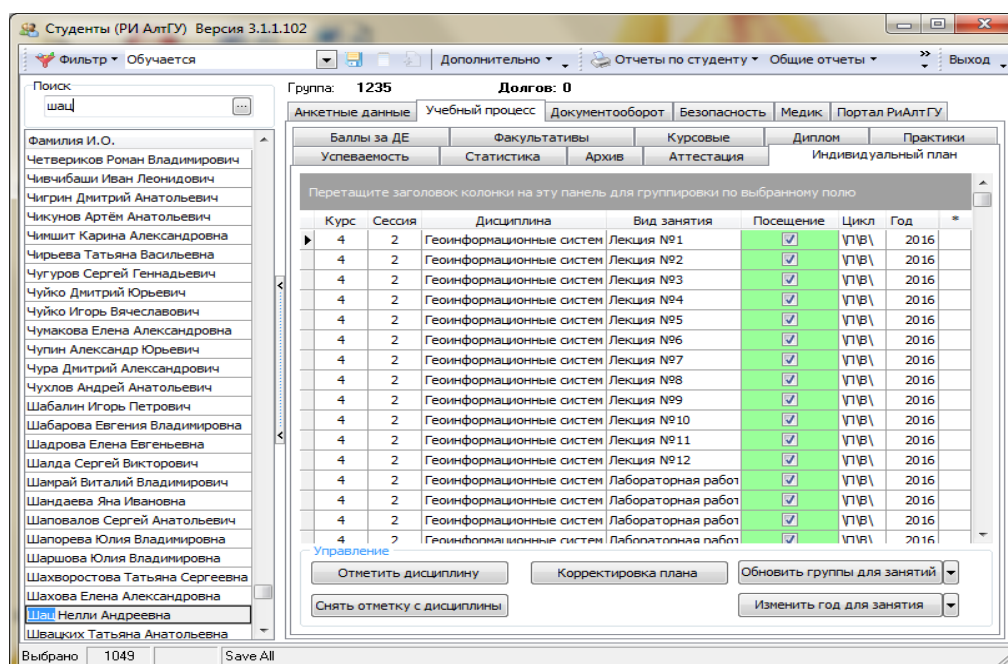


Рисунок 1.6 – Информационная система «Студенты», формирование индивидуального учебного плана обучающегося

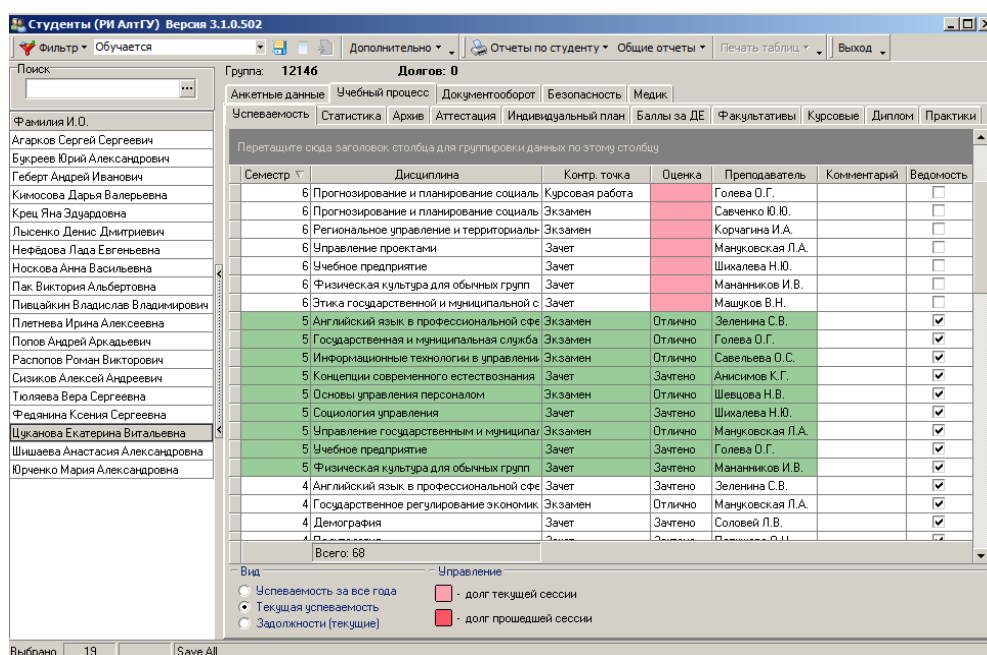


Рисунок 1.7 – Информационная система «Студенты»

На основании сформированных индивидуальных учебных планов студентов, количества студентов в группах и информации из ИС «Кадры» (Рис 1.8.) (должность преподавателя, количество часов на ставку и т.д.), формируются учебные поручения преподавателей на учебный год, имеется

возможность закреплять каждого преподавателя отдельно за видами занятий: лекцией, практическим занятием, семинарским занятием, лабораторной работой и промежуточной аттестацией в семестре (Рисунок 1.9).

Кадры - Савельева Ольга Сергеевна

Фильтрация Отчёты

Поиск: а

ФИО

- Абрамчук Ольга Яковлевна
- Андрусенко Любовь Анатольевна
- Анисимов Константин Геннадьевич**
- Анисимова Елена Александровна
- Аничкин Евгений Сергеевич
- Антропов Александр Васильевич
- Аристи Наталья Сергеевна
- Афиногенов Сергей Сергеевич
- Балин Алексей Петрович
- Барышников Евгений Николаевич
- Барышникова Юлия Андреевна
- Бахирева Анна Алексеевна
- Бахметова Елена Павловна
- Безгинов Владимир Иванович
- Безгинова Татьяна Капитоновна
- Белюсова Яна Николаевна
- Беляев Виктор Иванович
- Беспечный Олег Васильевич
- Бирюкова Тамара Дмитриевна
- Блошкин Алексей Николаевич
- Боков Геннадий Федорович
- Бордачева Марина Викторовна
- Борисенко Светлана Николаевна
- Боровков Александр Михайлович
- Букшина Светлана Владимировна

Количество сотрудников: 218

Основные сведения | Паспортные данные | Образование | Ученые звания и степени

Фото: [Portrait of Konstantin Anisimov]

Фамилия: Анисимов
Имя: Константин
Отчество: Геннадьевич
Дата рождения: 25.04.1974 39 Мужской

Вид: [Blank]
Телефон физ. лица: [Blank]
Телефон служебный: +7 (38557) 4-20-04, доб. 11-11
Мегафон: [Blank]

Вид занятости: [Blank]
Основное место работы: Директор
Внутреннее совместительство: Доцент

Подразделение: Дирекция
Кафедра «Математики и прикладн

Дата приема на работу: 17.07.2000
Испытательный срок: 0
№ трудового договора: 166
Занимаемых ставок: 1
Дата начала договора: 01.11.2008
График работы: Основной пятидневный гра
Дата окончания договора: 30.11.2013
Вид расчета: Оклад по дням
Дата увольнения: [Blank]

Рисунок 1.8 – Информационная система «Кадры»

Поручения - Савельева Ольга Сергеевна

Учебный год: 2013 Форма обучения: Все (15) Кафедра: Все (7) Курс: Все (6)

Отчеты: Разное Настройки

Отчеты по преподавателю
Отчеты по кафедре

Курс	Семестр	Форма обуч.	Специальность	Лет обу	Группа	Часов	Кол-во	Пре
4	2	ДО	МО	5	1201; 1204	20	18	Жданова Е. А.
		ЗО	ПМУ	6	1204б	10	12	Жданова Е. А.
4	2	ДО	МО	5	1201; 1204	12	18	Жданова Е. А.
		ЗО	ПМУ	6	1204б	2	12	Жданова Е. А.
	3	ЗО	ПМУ	6	1204б	2	13	Жданова Е. А.
4	2	ДО	МО	5	1201; 1204	4,5	18	Жданова Е. А.
		ЗО	ПМУ	6	1204б	3,25	13	Жданова Е. А.
4	3	ЗО	ПМУ	6	1204б	9,75	13	Жданова Е. А.

Изменить распределение Объединить в одну группу Распределить по учебному

В пр-м	№	Занятие	Семинар	Группа	Кол.	Преподаватели	Ст. до	Расписание
1	1	Занятие	Семинар №1	1201; 1204	18	Жданова Е. А.	1	2014, 3 пара, 21
2	2	Занятие	Семинар №2	1201; 1204	18	Жданова Е. А.	1	2014, 3 пара, 21
3	3	Занятие	Семинар №3	1201; 1204	18	Жданова Е. А.	1	2014, 3 пара, 21
4	4	Занятие	Семинар №4	1201; 1204	18	Жданова Е. А.	1	2014, 3 пара, 21
5	5	Занятие	Семинар №5	1201; 1204	18	Жданова Е. А.	1	2014, 3 пара, 21
6	6	Занятие	Семинар №6	1201; 1204	18	Жданова Е. А.	1	2014, 3 пара, 21

Рисунок 1.9 – Информационная система «Поручения»

ИС «Поручения» позволяет формировать множество отчетов: учебное поручение на каждого ППС (Рисунок 1.10), сводное поручение на кафедру и т.д.

Информационные системы «Редактор рабочей программы учебной дисциплины» и «Редактор учебно-методического комплекса дисциплины» предназначены для создания рабочих программ и учебно-методических комплексов дисциплин вместе с фондами оценочных средств по дисциплинам учебных планов.

Эти ИС связаны с ИС «Учебные планы», ИС «Компетенции», ИС «График учебного процесса» и с электронно-тематическими планами дисциплин. Из ИС «Учебные планы» выбирается направление подготовки, профиль и название дисциплины.

Затем в ИС «Редактор УМК» и ИС «Редактор РПД» отправляется номер семестра, в котором преподается дисциплина и итого часов за семестр (Рисунок 1.11).

Из электронно-тематического плана в эти ИС отправляется тематический план, с разбивкой дисциплины на дидактические единицы, темы, с указанием аудиторных и самостоятельных часов и вид промежуточной аттестации.

X	Форма обучения	Лет обучения	Всего часов	Лек.	Лаб.	Сем.	Зач.	Экз.	Курс.
☒	очная	4 года	432	90	86	0	5 сем.	6,7 сем.	7 сем.
☒	заочная	5 лет	432	20	24	0		4,5 курс	5 курс
☒	очная	3 года	432	98	96	0		7,8 сем.	8 сем.
☒		3 г. 6 мес.	432	24	26	0	3 курс	2 курс	2 курс
☒		3 года	432	24	26	0	3 курс	2 курс	2 курс

Рисунок 1.10 – «Редактор создания Учебно-методического комплекса дисциплины»

Заместитель директора по учебной работе _____ /Жданова Е. А./
Заведующий кафедрой _____

17

Из ИС «Компетенции» автоматически переносится индекс компетенции, формулировка, перечень компонентов (знать, уметь, владеть) и уровни их освоения (пороговый и повышенный).

Информационная система позволяет экспортировать учебно-методический комплекс и готовую рабочую программу дисциплины в файл Microsoft Word и выводить печатную форму (Рисунок 1.12).

Мастер создания УМК

УМК

Учебные планы

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Алтайский государственный университет»
Рубцовский институт (филиал)

УТВЕРЖДАЮ
Директор Рубцовского института
(филиала) АлтГУ
_____ К.Г. Анисимов
М.П. _____
«___» _____ 20__ г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО ДИСЦИПЛИНЕ
ЭКОНОМИКО-МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИ**

Уровень основной профессиональной образовательной программы –
бакалавриат
Направление подготовки – 38.03.01 Экономика
Профиль - Финансы и бухгалтерский учет
Форма обучения – очная, заочная
Кафедра - математики и прикладной информатики

Рубцовск
2015

При разработке учебно-методического комплекса учебной дисциплины в
основу положены:

1) ФГОС ВО по направлению подготовки 080100.62 Экономика,
утвержденный приказом Министерства образования и науки РФ от «21» декабря
2009 г., №747

2) Учебный план по направлению подготовки 38.03.01 Экономика,
утвержденный решением Ученого совета Рубцовского института (филиала)
АлтГУ от «17» февраля 2014 г., протокол № 6

Учебно-методический комплекс одобрен на заседании кафедры математики
и прикладной информатики от «28» августа 2014 г., протокол № 1

Заведующий кафедрой МПИИ, к.т.н. доцент _____ Е.А. Жданова

Разработчик:
Преподаватель _____ М.В. Досымова

Работодатель:
Председатель Комитета по
финансам, налоговой и _____ В.И. Пыльников
кредитной политике
Администрации г. Рубцовска

Рисунок 1.12 – Формируемый отчет учебно-методического комплекса дисциплины

В начале каждого учебного года в ИС «Компетенции» обновляются все паспорта компетенций направлений подготовки высшего образования и среднего специального образования в соответствии с ФГОС (Рисунок 1.13).

Ответственными за эту работу являются заведующие кафедрами, методисты и преподаватели, обеспечивающие данную компетенцию.

Обновленный паспорт компетенций выставляется на образовательный портал для студентов.

Компетенции - Торопова Ирина Сергеевна

Год: 2015 Код направления: 38.03.01

Экономика

Паспорт компетенции Копировать Вырезать Вставить Копировать всё Вставить всё Удалить всё

Код	Описание	Кафедра
ОК -1	Владеет культурой мышления, способен к обобщению, анализу, восприятию информации, постановке проблем и самостоятельному нахождению способов их решения;	Кафедра «Общественных дисциплин»
ОК -3	Способен понимать движущие силы и закономерности исторического процесса; события и процессы в истории, их значение, особенности развития;	Кафедра «Общественных дисциплин»
ОК -10	Способен критически оценивать свои достоинства и недостатки, наметить пути и выбрать средства их развития;	Кафедра «Общественных дисциплин»
ОК -11	Осознает социальную значимость своей будущей профессии, обладает высокой мотивацией к выполнению профессиональных обязанностей;	Кафедра "Экономики"
ОК -12	Способен понимать сущность и значение информации в развитии современного информационного общества, обладает навыками критического анализа информации;	Кафедра «Математики и прикладной информатики»

Знать Уметь Владеть Удалить

Знать, уметь, владеть.	Описание
Знать	основные методы научного познания, формы и законы рационального мышления
Знать	психологию практического и социального интеллекта
Знать	основы социального взаимодействия
Знать	социологию и психологию управления
Знать	философскую постановку проблемы познания (как возможно познание, критерии достоверности знания)
Уметь	использовать различные методы научного познания
Уметь	мыслить логически непротиворечиво
Уметь	осуществлять социальное действие
Уметь	принимать управленческие решения
Владеть	культурой логического мышления
Владеть	навыками взаимодействия в коллективе и группе, управления ими

Уровни сформированности компетенций Учебные дисциплины

Пороговый уровень Повышенный уровень Удалить

Уровень сформированности компетенции	Основные признаки уровня
Пороговый уровень	Знает методы эмпирического и теоретического исследования, а также смежные методы
Пороговый уровень	Знает основные формы мышления: понятие, суждение, умозаключение
Пороговый уровень	Знает законы тождества, противоречия, исключенного третьего и достаточного основания
Повышенный уровень	Знает способы проверки правильности дедуктивных умозаключений
Повышенный уровень	Знает методы научной индукции
Повышенный уровень	Понимает взаимосвязь основных законов логики

Рисунок 1.13 – Информационная система «Компетенции»

В информационную систему «График учебного процесса» вносятся графики, утвержденные Ученым советом института (Рисунок 1.14).

В графике отражаются специальности и направления подготовки с указанием количества лет обучения и номера групп. Для каждой группы указывается период аудиторного обучения, практик, сессий и каникул.

Информационная система позволяет экспортировать график учебного процесса в файл Microsoft Excel. При подаче абитуриентом заявления на поступление, технический секретарь приемной комиссии вносит данные

[illegible]

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

9 1 1 91

Из ИС «Поручения» и ИС «График учебного процесса» информация поступает в ИС «Редактор расписания», где происходит понедельный расчет часов и создание расписание каждому студенту и преподавателю.

Расписание отображается на образовательном портале института и мобильном приложении «Расписание» (для ОС Android), размещенное на Google Play (Рисунок 1.16 и 1.17).

Редактор расписания занятий - Черноусова Ульяна Николаевна (chernousova)

Расписание по группам

Форма обучения: ДО (Бак.) Специальности: 6 элементов

Курс: 3 Лет обучения 4 Группы: Все (2) ДО 1 поток 3 курса ЭФ

01.10.2015

Дата и пара		Группы	
№	Пара	Государственное и муниципальное управление 1234	Прикладная информатика 1235
ВТОРНИК 29 СЕНТЯБРЯ	1		Проектирование, настройка и обслуживание ЛВС, 1235, (9 чел.), Лекция №5, Ст. преп. Дергилев О. В., 306
	2		Программная инженерия, 1235, (9 чел.), Лабораторная работа №3, Ст. преп. Кулаков К. М., 205
	3	Государственная и муниципальная служба, 1234, (7 чел.), Лекция №6, Доц. кандидат экономических наук Голева О. Г., 203	Проектирование обучающих систем, 1235, (9 чел.), Лекция №5, Ст. преп. Рязанова О. В., 305
	4	Основы математического моделирования социально-экономических процессов, 1234, (7 чел.), Лекция №4, Доц. кандидат технических наук Жданова Е. А., 213	
	5	Основы управления персоналом, 1234, (7 чел.), Лекция №7, Проф. кандидат экономических наук Машуков В. И., 310	
	6		
	7		
	8		
СРЕДА 30 СЕНТЯБРЯ	1		Базы данных, 1235, (9 чел.), Лекция №4, Ст. преп. Рязанова О. В., 307
	2		Теория оптимального управления, 1235, (9 чел.), Семинар №3, Доц. кандидат физико - математических наук Шевченко А. С., 314
	3		
	4	Индикаторы социально-экономического развития муниципальных образований, 1234, (7 чел.), Семинар №3, Доц. кандидат экономических наук Голева О. Г., 108	
	5	Учебное предприятие, 1234, (7 чел.), Лабораторная работа №5, Доц. кандидат экономических наук Мануковская Л. А., 204	
	6		
	7		
	8		

Рисунок 1.16 – Информационная система «Редактор расписания занятий»

← Занятия

☒ Группа ☐ Преподаватель ☐ Аудитория

1248C9 29.04.2015

ПОКАЗАТЬ

День		Неделя
№	Среда 29.04.15	Время
1	Биология (Семинар №21), 213 1248C9 Старший преподаватель, Миронова Елена Владимировна	08:00 / 09:30
2	Обществознание (Семинар №15), 309 1248C9, 1245C9 кандидат исторических наук Доцент, Папушева Оксана Николаевна	09:40 / 11:10
3	Основы безопасности жизнедеятельности (Семинар №28), 309 1248C9, 1245C9, 1242C9 кандидат педагогических наук Доцент, Загайнов Сергей Алексеевич	11:20 / 12:50
4	Литература (Семинар №42), 214 1248C9 кандидат педагогических наук Доцент, Лямина Ольга Николаевна	13:20 / 14:50

Рисунок 1.17 – Мобильное приложение «Расписание» для ОС Android.

На основании тематических планов дисциплин на портал института для дальнейшей реализации балльно-рейтинговых технологий (БРТ) методист кафедры вносит электронно-тематические планы дисциплин с разбивкой на дидактические единицы (Рисунок 1.18 и 1.19).

По мере изучения ДЕ преподаватели вносят результаты текущей успеваемости в ведомости по БРТ, а методист кафедры вносит эти результаты в ИС «Студенты» (Рисунок 1.20). Рейтинги студентов по результатам БРТ отображаются на сайте института, студенты могут их посмотреть в Личном кабинете на сайте (Рисунок 1.21).

Начало	Пред.	1	След.	Конец	Обновить	Печать	PDF	Экспорт	Всего страниц: 1
--------	-------	---	-------	-------	----------	--------	-----	---------	------------------

Тематический план						
Дисциплина: Имитационное моделирование экономических процессов						
Прикладная информатика (дневная (бакалавриат)-4) год поступления: 2013						
Вариативная часть						
Темы	Всего	Лекц.	Семин.	Лабор. раб.	Сам. раб.	
ДЕ 1				Макс. балл: 10		
1. Математические модели планирования в социально-экономических системах	10	2			8	
2. Имитационное моделирование и интерактивные проблемно-ориентированные системы.	12	2	2	2	6	
ДЕ 2				Макс. балл: 30		
1. Модели выпуклого программирования	11	2	2		7	
2. Модели динамического программирования	10	2	2		6	
ДЕ 3				Макс. балл: 30		
1. Модели теории систем массового обслуживания	12	4	2	4	2	
2. Статистическое моделирование (метод Монте-Карло)	14	6		2	6	
ДЕ 4				Макс. балл: 30		
1. Обработка результатов моделирования: оценка вероятности; гистограмма; оценка мат. ожидания; оценка дисперсии; оценка корреляции; оценка характеристик случайного процесса; количество реализаций, обеспечивающих заданную точность	22	8	4	4	6	
Курсовая работа				Макс. балл: 100		
1. Курсовая работа	0				0	
Экзамен				Макс. балл: 40		
1. Экзамен 1	0				0	
Всего часов:	91	26	12	12	41	

Рисунок 1.18 – Электронный тематический план

Контроль за проведением текущих аудиторных занятий и промежуточных аттестаций происходит в ИС «Редакторе расписания» автоматически.

Таким образом, ни один час аудиторных обязательных занятий и консультаций (ставятся в расписание каждому ППС еженедельно методистом кафедры и отображаются на портале) (Рисунок 1.22) не пропадает, осуществляется тройной жесткий контроль: бюро расписания – кафедра – руководство института.

Редактирование электронного тематического плана

Имитационное моделирование экономических процессов
Прикладная информатика (дневная (бакалавриат) - 4) 2013

Создание/редактирование тем

Добавить/редактировать тему

* Добавление /редактирование темы для последующего закрепления за дисциплиной

Создание/редактирование ДЕ

Создать/редактировать ДЕ

* Создание /редактирование ДЕ, прикрепление к темам

Закрепление созданных/отредактированных тем за занятиями

.....

* Для закрепления темы, нужно выбрать ТЕМУ из списка и отметить то занятие, за которым она должна быть закреплена
Одна тема может быть закреплена за несколькими занятиями
За одним занятием может быть закреплено несколько тем

Вид занятия	Закрепленные темы
☒ Лекция №1	Математические модели планирования в социально-экономических системах [X]
☒ Лекция №2	Имитационное моделирование и интерактивные проблемно-ориентированные системы. [X]
☒ Лекция №3	Модели выпуклого программирования [X]
☒ Лекция №4	Модели динамического программирования [X]
☒ Лекция №5	Модели теории систем массового обслуживания [X]
☒ Лекция №6	Модели теории систем массового обслуживания [X]

Рисунок 1.19 – Редактирование электронного тематического плана

Студенты (РИ АлтГУ) Версия 3.1.1.102

Фильтр: Обучается

Поиск: писа

Группа: 1225 Долгов: 7

Анкетные данные Учебный процесс Документооборот Безопасность Медик Портал РИ АлтГУ

Успеваемость	Статистика	Архив	Аттестация	Индивидуальный план	
Баллы за ДЕ	Факультативы	Курсовые	Диплом	Практики	
Дисциплина	Семестр	ДЕ	Балл	Балл по группе	Пренциальный балл
Дифференциальные уравнения	3	ДЕ 3	20	73	0
		Экзамен	35		
		ДЕ 1	10		
		ДЕ 2	30		
Имитационное моделирование...	6	ДЕ 3	30	90	0
		ДЕ 4	20		
		Экзамен	0		
	6	Курсовая работа	80	80	0
		ДЕ 1	50		
Информатика и программиров...	1	ДЕ 2	20	70	0
		Экзамен	0		
	2	ДЕ 1	0	10	0
		ДЕ 2	10		
Информационные системы и т...	3	ДЕ 3	7		
		ДЕ 4	10	60	0
		ДЕ 5	6		
		ДЕ 6	10		

Управление

Оценить...

Удалить балл за ДЕ...

Удалить балл за группу...

Выбрано: 1045 Save All

Рисунок 1.20 – Информационная система «Студенты»

Курс 1, сессия:1		
1	Шац Нелли Андреевна	81
2	Боечко Александр Сергеевич	76
3	Комолов Андрей Андреевич	62
4	Болдаков Олег Витальевич	57
5	Граф Станислав Андреевич	45
6	Краснобрыжев Владислав Николаевич	36
7	Кулапин Роман Игоревич	34
8	Уразов Дмитрий Александрович	25
9	Еремин Николай Алексеевич	23
Средний балл:		49
Курс 2, сессия:2		
1	Шац Нелли Андреевна	75
2	Боечко Александр Сергеевич	65
3	Уразов Дмитрий Александрович	61
4	Краснобрыжев Владислав Николаевич	60
5	Комолов Андрей Андреевич	52
6	Болдаков Олег Витальевич	43
7	Граф Станислав Андреевич	37
8	Кулапин Роман Игоревич	22
9	Еремин Николай Алексеевич	8
Средний балл:		47



**Рубцовский институт
(филиал)
АлтГУ**

[← Старая версия](#) [👁 Версия для слабовидящих](#)

[Личный кабинет](#)
Войти

Сведения об образовательной организации ▾
Образование ▾
Наука ▾
Библиотека ▾
Расписание ▾

👤 Поступающему
🎓 Студенту
🔔 Школьнику
🏢 Сотруднику

Главная / Индивидуальное расписание занятий

День Неделя Месяц

« Март 2018 »

Пн	Вт	Ср	Чт	Пт	Сб	Вс
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Сегодня

Преподаватель Аудитория Группа Кафедры

1245

Пара	1245 Вторник 06.03.2018	Время
1		08:00-09:30
2	Сертификация 1С:Профессионал (Лабораторная работа №10), 215 1255 У; 1245; Преподаватель, Соловьева Татьяна Сергеевна;	09:40-11:10
3	Геоинформационные системы в экономике (Лекция №7), 203 1255 У; 1245; Старший преподаватель, Досымова Маргарита Владимировна;	11:20-12:50
4	Предметно-ориентированные экономические информационные системы (Лабораторная работа №12), 215 1255 У; 1245; Преподаватель, Соловьева Татьяна Сергеевна;	13:20-14:50
5		15:00-16:30
6		16:40-18:10
7		18:20-19:40
8		19:50-21:10

Список группы:

Грибенников Александр Васильевич
 Гумеров Константин Александрович
 Костенко Виталий Витальевич
 Течин Денис Дмитриевич

24

Информационная система «Электронная научная библиотека института» (Рисунок 1.23).

Продолжается совершенствование созданной в институте в 2013 г. в соответствии со стандартом RUSMARC электронной библиотеки, которая имеет базу учебных и методических материалов, содержащую более 90000 наименований.

В настоящее время вместе с АлтГУ библиотека института подключена к двум электронным библиотечным системам: «Лань» и «Университетская библиотека on-line».

Самостоятельно научная библиотека института вошла в Корпоративную библиотечную сеть Алтайского края и нами подписаны договоры с электронными библиотечными системами:

1. ЭБС «Центральная библиотека» г. Рубцовск.
2. ЭБС ФСС «Финансовый Директор».
3. ЭБСС «Система Главбух VIP».
4. ЭБС «Юрайт».
5. ЭБС ZNANIUM (СПО).
6. ЭБС eLIBRARY.RU.
7. ЭБС East View.

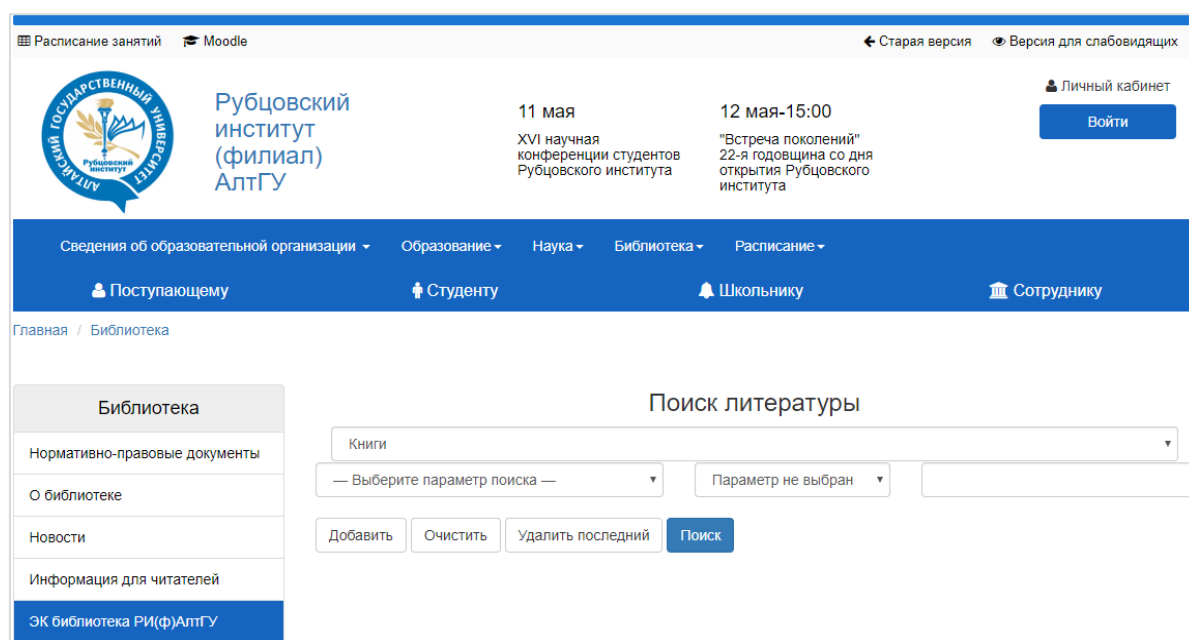


Рисунок 1.23 – Электронная научная библиотека института

Благодаря ЭБИС института нами создана методическая и информационная поддержка для перехода обучения студентов на индивидуальную траекторию, теперь в начале учебного года каждый преподаватель по своей дисциплине в «Личном кабинете» студента формирует список индивидуальной учебной литературы, студент заказывает ее через электронный каталог и получает на абонементе.

Созданная система управления всеми образовательными процессами института, которая является очень гибкой и динамичной, позволила встроить в нее систему электронного обучения на платформе Moodle (Рисунок 1.24).

Новые информационные технологии, средства вычислительной техники и телекоммуникации института предоставляют развитые возможности для организации данного процесса обучения, поэтому в 2014 г. нами было принято решение о внедрении дистанционных форм обучения на основе современных средств телекоммуникационной связи.

Данная цель является долгосрочной, поэтому в настоящее время готовится основная база для ее выполнения.

The screenshot shows a Moodle course interface. At the top, the header includes 'Рубцовский институт', 'Русский (ru)', and a user profile 'Ирина Сергеевна Торопова'. The course title is 'Информационные технологии и платформы разработки информационных систем'. Below the title is a breadcrumb trail: 'В начало' > 'Курсы' > 'Кафедра математики и прикладной информатики' > 'Информационные технологии и платформы разработки ИС'. A 'Режим редактирования' button is in the top right. The left sidebar contains a 'НАВИГАЦИЯ' section with links like 'Моя домашняя страница', 'Страницы сайта', and 'Текущий курс'. Below it is a 'НАСТРОЙКИ' section with options like 'Управление курсом', 'Режим редактирования', and 'Пользователи'. The main content area features a decorative graphic with the text 'На курс «Информационные технологии и платформы разработки информационных систем»'. Below this is a 'Новостной форум' section with the heading 'ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ' and a list of topics including 'Лекция 1: Введение в информационные технологии', 'Эволюция ИТ, этапы их развития', 'Развитие современных ИТ', 'Классификация ИТ', 'Лекция 2: Аспекты реализации информационных технологий', 'Пользовательский интерфейс и его виды', and 'APM'. On the right side, there are three sections: 'ПОИСК ПО ФОРУМАМ' with a search bar, 'ПОСЛЕДНИЕ НОВОСТИ' with a 'Добавить новую тему...' button and a list of recent posts, and 'ПРЕДСТОЯЩИЕ СОБЫТИЯ' with a list of upcoming events like 'Лекция 8.1: Организация программного обеспечения АИС в локальных сетях' and 'Доклады (Окончание срока подачи работ)'. At the bottom right, there are links to 'Перейти к календарю...' and 'Новое событие...'.

Рисунок 1.24 – Пример учебного курса в обучающей среде Moodle

Сейчас в институте создан многоуровневый информационный комплекс для обеспечения учебного процесса. Данный комплекс на основе информации, хранящейся в ИС «Кадры», «Абитуриенты», «Студенты», «Электронная библиотека», «1С: Предприятие 8.3.», позволяет автоматизировать все этапы управления учебным процессом, начиная от проектирования учебных планов и заканчивая формированием поручений ППС.

В институте завершено создание единого образовательного пространства, которое функционирует в рамках образовательного портала, все существующие информационные системы (ИС), обеспечивающие учебный процесс, связаны и функционируют следующим образом: на основании электронных рабочих учебных планов (ИС «Учебные планы»), графиков учебного процесса (ИС «Графики учебного процесса») специальностей и направлений осуществляется расчет часов на текущий учебный год (ИС «Расчет часов»); на основании расчета часов и данных из ИС «Кадры» в ИС «Поручения» происходит закрепление часов за преподавателями кафедр; на основании закрепления преподавателей и расчета часов формируются планы-графики специальностей, которые выдаются на руки каждому студенту всех форм обучения в виде учебных поручений; на основании планов-графиков и графиков учебного процесса в ИС «Расписание» разрабатывается еженедельное расписание занятий всего института. В расписании указываются номер учебной группы, учебная дисциплина, вид занятия, дата, время и место его проведения, фамилии преподавателей, проводящих занятие. Расписание занятий выставляется в электронном виде на образовательном портале института, на сенсорных экранах в холлах учебных корпусов.

ИС «Студенты» связана через модуль сопряжения с ИС «1С: Бухгалтерия», таким образом, контролируется оплата студентов за обучение и на основании этой информации формируются Справки-вызовы и Справки-подтверждения для студентов заочного отделения.

Научная электронная библиотека института оснащена считывающими сканер-устройствами, связывающими электронный библиотечный каталог и читательский билет студента, что позволяет контролировать и сохранять фонд научной библиотеки института.

В парке средств вычислительной техники РИ (филиала) АлтГУ имеются 193 единиц ноутбуков и персональных компьютеров. Из них:

- 15 персональных компьютеров с процессорами Intel Core i3;
- 15 персональных компьютеров с процессорами Intel Pentium G2010;
- 34 персональных компьютеров с процессорами Intel Pentium E5700;
- 32 персональных компьютера с процессорами Intel Pentium G630;
- 18 персональных компьютеров с процессорами Intel Celeron D 315;
- 18 персональных компьютеров с процессорами Intel Celeron 420;
- 16 ноутбуков с процессорами Intel Celeron T3100;
- 17 ноутбуков с процессорами Intel Pentium B970;
- 14 ноутбуков с процессорами Intel Duo Core T6570;
- 14 ноутбуков с процессорами Intel Celeron 550.

Совместно с данным оборудованием, в учебном и производственном процессах, используется 13 мультимедийных проекторов, 43 единицы лазерных принтеров (3 цветных), 22 единицы планшетных сканеров, 14 единиц копировально-множительной техники.

Количество персональных компьютеров, используемых в учебном процессе, составляет 193 единицы Аудиторный фонд института, оснащенный вычислительной техникой, включает в себя: 11 компьютерных классов. Из них один класс на 15 машин используются в режиме свободного доступа студентов.

Еще одна аудитория на 8 компьютеров используется для проведения лабораторных занятий по дисциплинам специальностей кафедры

«Математики и прикладной информатики». Мобильные классы включают в себя 4 компьютерных кабинета с ноутбуками, и активно используются в учебно-образовательной деятельности, как для тематических занятий, так и для организации доступа к ресурсам корпоративной сети и Internet на территории РИ (филиала) АлтГУ.

Все административные подразделения, кафедры, компьютерные классы и лаборатории объединены в единую локальную вычислительную сеть, построенную на основе традиционных структурированных кабельных системах (СКС), суммарной протяженностью около 12 км, со скоростью передачи данных в сетях подразделений 1Гбит/с.

Кроме этого, имеется 2 сегмента магистральных волоконно-оптических каналов связи, объединяющих между собой оба корпуса РИ (филиала) АлтГУ в пределах кампуса. Общая протяженность данной линии 1,58 км с пропускной способностью кабельной системы 1Гбит/с и возможностью расширения до 10 Гбит/с, за счет модернизации активного сетевого оборудования. Один оптический сегмент используется как опорный канал для доступа пользователей к ресурсам локальной сети института и внешним сетям, а другой для организации внутренней цифровой IP-телефонии на основе голосовых шлюзов VoIP.

Необходимо отметить, что доступ к ресурсам ЛВС на территории Рубцовского института (филиала) АлтГУ может осуществляться как по традиционным проводным технологиям, так и через беспроводные системы Wireless WiFi (до 300 Мбит/с).

В настоящее время, в филиале существует следующая конфигурация серверов института:

1. Файловый сервер RFAGU. Выполняет функции главного контроллера домена RBDOM, обеспечивающего централизацию управления ЛВС института, файлового сервера личных и общедоступных ресурсов, СПС и Консультант +. Технические характеристики: 2xP4Xeon 2,8GHz, 4x2Gb, IDE-SATA 500Gb, 1Tb, RAID5 SATA 3x1Tb, 2xGigabit Ethernet,

установленное ПО – Gentoo Linux Server Edition.

2. Сервер WWW. Выполняет функции базового WEB-сервера со следующим набором сервисов: HTTP, FTP, DNS, MAIL, PROXY, APACHE, MySQL. Технические характеристики: 2xP4Xeon, 3,0GHz, 4x2Gb, RAID1 SATA 2x500Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux Server Edition.

3. Сервер GTW. Выполняет функции шлюза, обеспечивающего интеграцию сети института в глобальные сети Internet. Здесь установлен корпоративный межсетевой экран. Технические характеристики: Pentium-IV 2,4GHz, IDE-SATA 80Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux Server Edition.

4. Сервер EDUSERVER. Выполняет функции сервера БД для используемых в учебном процессе банковских информационных систем «RS-Bank 5.1». На нем установлен web-сервер Apache с модулем PHP 7.2.2, обеспечивающий онлайн доступ к расписанию занятий, система управления курсами Moodle, СКУД обеспечивающая автоматическое управления входом и выходом людей в здание организации, модуль FastReport, обеспечивает формирование отчетов печати для приложений. Технические характеристики: Intel-I7 (4 core) 3,00GHz, 8Gb, IDE-SATA 500Gb, Gigabit Ethernet. Установленное ПО – MS Windows 2012 AS.

5. Сервер ORACLE. Выполняет функции сервера БД для хранения информации для портала и информации необходимой для учебного процесса. Технические характеристики: 2xP4Xeon, 2,8GHz, 24Gb, RAID5 SATA 4x200Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux Server Edition.

1.2 Анализ функционирования объекта исследования

Для функционального и объектного описания корпоративной информационной сети института был выбран язык моделирования UML [5, с.

23-25].

UML (англ. Unified Modeling Language – унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью [3, с 56-78].

UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

Для построения подобных схем в проекте использовалось CASE средство UML DESIGNER. Данное программное обеспечение является бесплатным продуктом компании Obeo. Выбранное CASE средство поддерживает последний стандарт языка моделирования 2.5 выпущенный в 2015 году. Для моделирования предметных областей предоставляются следующие виды диаграмм:

- диаграмма пакетов;
- диаграмма классов;
- диаграмма компонентов;
- диаграмма композитной/составной структуры;
- диаграмма развертывания;
- диаграмма использования;
- диаграмма деятельности;
- диаграмма состояний;
- диаграмма последовательности;
- диаграмма профиля.

Для моделирования исследуемого объекта основными видами

диаграмм стали:

- диаграмма компонентов;
- диаграмма развертывания;
- диаграмма вариантов использования.

Диаграмма компонентов (англ. Component diagram) – элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами.

В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т.п.

При помощи данного вида диаграмм легко описать структуру всей информационной системы в целом, поэтому для исследуемой корпоративной ИС Рубцовского института был смоделирован данный вид диаграмм (Приложение А.1).

На диаграмме компонентов легко видеть, что ИС института имеет очень сложную структуру взаимосвязанности компонентов. Но можно выделить несколько основополагающих факторов функционирования данной системы:

- основополагающими компонентами являются ИС «Приемная комиссия», «Студенты», «Кадры». Они являются основными источниками данных для всего рабочего процесса системы и организации в целом;
- вторыми по значимости идут ИС «Учебные планы», «График учебного процесса», «Поручения», «Расписание», «Компетенции» при помощи которых формируется рабочий процесс всей организации;
- информационные системы обеспечивающие вспомогательные функции являются «1С Розница», «1С Бухгалтерия Предприятия», «1С Зарплата и Кадры государственного учреждения», «Библиотека», «СКУД», «LDAP»;
- открытыми точками доступа к корпоративной информации являются ИС «Портал», мобильное приложение «Расписание РИ (филиал)

АлтГУ», ЭОС «Moodle».

Диаграмма развёртывания, Deployment diagram в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI). Для исследуемой сети диаграмма развертывания выглядит следующим образом (Приложение А.2).

Диаграмма прецедентов (диаграмма вариантов использования) в UML – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент – возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Для информационной системы института узлом, предоставляющим общедоступную информацию, является Образовательный портал. Именно для него и следует строить диаграмму прецедентов (Приложение А.3).

В ходе анализа и моделирования объекта исследования были выявлены проблемы:

- сильная связанность компонентов ИС;
- взаимосвязь между компонентами реализуется не по регламентируемому принципу;
- образовательный портал выступает в качестве промежуточного

звена в обмене информацией между сторонними приложениями и КИС;

- устаревшее оборудование в корпоративной сети института;
- разобщенность применимых технологий для разработки.

1.3 Определение цели и задач проектирования API

API – программный интерфейс системы для взаимодействия с системами ТСП. Использование подобного интерфейса в корпоративной информационной системе позволяет достичь следующих положительных сторон:

- возрастает скорость обработки данных на серверной стороне, потому что в ответ возвращаются только данные;
- нагрузку на API можно регулировать, к примеру, выставять лимиты на определенное приложение, которое ведет «агрессивный» запрос данных;
- изменение алгоритма обработки данных на сервере не затрагивает клиентскую сторону приложений;
- внедрение нового приложения по временным и финансовым затратам удешевляется;
- идемпотентность, большое количество идентичных запросов будут возвращать объект одинакового состояния, поэтому сохраняется главный критерий информации – ее точность и идентичность для всех пользователей.

Перечисленные положительные факторы от внедрения API в корпоративную ИС института дают возможность поставить главную цель данного исследования:

- разработать программный интерфейс (API) для корпоративной информационной сети Рубцовского института (филиал) АлтГУ.

Главными решаемыми задачами API являются:

- разграничение процесса обработки данных и формирование GUI

ответа пользователю;

- усовершенствование существующей архитектуры ИС;
- регулирование нагрузки работающих приложений (на первом этапе): внутренних – официальный портал института и внешних – мобильное приложение;
- улучшение таких показателей ИС как масштабируемость и расширяемость;
- уменьшение затрат внедрения новых приложений;
- уменьшить платформа зависимость всех старых и новых приложений.

1.4 Обзор и анализ существующих разработок, выбор технологии проектирования

1.4.1 Обзор и анализ существующих разработок

В настоящее время официальный портал Рубцовского института имеет минимальную реализацию API, для работы мобильного приложения, которое обеспечивает следующие возможности для пользователя:

- просмотр расписания (день, неделя, месяц);
- просмотр консультаций;
- просмотр предметных комиссий.

Соответственно имеющийся программный интерфейс реализует модули для обмена информацией с мобильным приложением.

Существующая разработка в настоящее время перестала удовлетворять следующим критериям:

- недостаточный объем получаемой информации;
- не структурированность возвращаемого объекта данных;
- не структурированность запросов;

- время ответа на запрос;
- версионированность API (полностью отсутствует);
- отсутствие возможности выбора формата возвращаемых данных;
- избыточность возвращаемых данных;
- отсутствие контроля доступа к полученным данным;
- отсутствие возможности авторизации пользователя, для получения персональной информации;
- не соответствие принятым стандартам архитектуры проектирования API;
- отсутствие шифрования всего соединения (HTTPS).

Новая разработка призвана устранить все перечисленные недостатки.

Но стоит учитывать, то что старые наработки должны остаться полностью в работоспособном состоянии, для того что действующее мобильное расписание не потеряло своего функционала.

1.4.2 Выбор технологии проектирования информационной системы

Технология проектирования ИС – совокупность методов и средств проектирования ИС, а также организации и управления, внедрения и модернизации проекта, ИС [7, с. 78-79].

Проектирование ИС охватывает три основные области:

- проектирование объектов данных, которые будут реализованы в базе данных;
- проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
- учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т.п.

При проектировании объектов данных при реализации API на базе

работающей ИС стоит учитывать, что они не должны нарушать работоспособность действующих систем. Поэтому при таком подходе создаются лишь вспомогательные или расширяющие объекты хранилища данных.

Примерами таким вспомогательных объектов могут быть таблицы для хранения состояния клиента, истории обращения за ресурсами, данные о выданных ключах доступа к персонализированным данным пользователей и др.

Для выполнения запросов к данным предполагается традиционный подход при создании API с реализацией REST архитектуры. REST (сокр. от англ. Representational State Transfer – «передача состояния представления») – архитектурный стиль взаимодействия компонентов распределённого приложения в сети. В сети Интернет вызов удалённой процедуры может представлять собой обычный HTTP-запрос (обычно «GET» или «POST»; такой запрос называют «REST-запрос»), а необходимые данные для формирования ответа передаются в качестве параметров запроса [1, с. 7-9].

Свойства архитектуры, которые зависят от ограничений, наложенных на REST-системы:

- производительность – взаимодействие компонентов системы может являться доминирующим фактором производительности и эффективности сети с точки зрения пользователя;
- масштабируемость для обеспечения большого числа компонентов и взаимодействий компонентов.

Рой Филдинг – один из главных авторов спецификации протокола HTTP, описывает влияние архитектуры REST на масштабируемость следующим образом:

- простота унифицированного интерфейса;
- открытость компонентов к возможным изменениям для удовлетворения изменяющихся потребностей (даже при работающем приложении);

- прозрачность связей между компонентами системы для сервисных служб;
- переносимость компонентов системы путем перемещения программного кода вместе с данными;
- надежность, выражающаяся в устойчивости к отказам на уровне системы при наличии отказов отдельных компонентов, соединений, или данных.

Для упрощения работы с составлением запросов для обращения за данными, следует создать несколько экранных форм в виде WEB приложения.

Данное приложение позволит реализовать базовую справку для работы с программным интерфейсом для сторонних разработчиков, а также конструктор REST запросов для клиентского приложения.

Наиболее частым реализуемой архитектурой для API является микросервисная архитектура. Микросервисы – современное представление сервис-ориентированной архитектуры (SOA), используемое для создания распределенных программных систем. Как и в SOA, модули в архитектуре микросервисов взаимодействуют по сети друг с другом для выполнения цели.

Ещё одно сходство в том, что микросервисы используют протоколо-независимую технологию. Данная архитектура является первой реализацией SOA, появившейся после внедрения DevOps, и она постепенно становится стандартом для непрерывно развивающихся систем [2, с. 20-25].

При реализации данной архитектуры в проектируемую ИС закладываются такие факторы как:

- модульность: модули можно легко заменить в любое время;
- функциональность: модули организованы вокруг функций, например, пользовательский интерфейс, логистика, выставление счета и т.д.;
- платформа независимость: модули могут быть реализованы с использованием различных языков программирования, баз данных,

аппаратных средств и программного обеспечения, в зависимости от того, что подходит лучше всего;

- отсутствие иерархии: архитектура симметричная, а не иерархическая (производитель – потребитель).

При реализации данной архитектуры принято выделять под каждый микросервис (модуль ИС) свой отдельный сервер (группу серверов) и объединять их высокоскоростными каналами связи, с дублированием если требуется высокая отказоустойчивость. Но при разработке небольших систем или прототипов применяют подход кластеризации. Так к примеру, могут быть выделены основные модули контроля доступа к ресурсам, сервис ресурсов (База данных или другое альтернативное хранилище данных), сервис обработки запросов к ресурсам, сервис распределения нагрузки.

Для построения данной архитектуры очень хорошо подходит модель жизненного цикла «Спираль». Она имеет ряд преимуществ перед инкрементной и каскадной моделями:

- улучшенный анализ рисков;
- хорошая документация процесса разработки;
- гибкость – возможность внесения изменений и добавления новой функциональности даже на относительно поздних этапах;
- раннее создание рабочих прототипов.

Для микросервисной архитектуры каждый виток будет являться новым разрабатываемым сервисом.

1.5 Выбор и обоснование проектных решений

1.5.1 Выбор фреймворков, технологий разработки и технологических платформ

В настоящее время существует большое количество разнообразных фреймворков для веб-разработок, например: Yii, Symphony, CodeIgniter. Но

гораздо выгоднее разработать систему внутри предприятия, основанную на уже существующих и применяемых в институте технологиях. Это упростит как создание проекта, так его модернизацию и поддержку в дальнейшем.

В качестве средств для разработки API был выбран Zend Framework, который использовался для разработки действующего портала, что является главным фактором выбора данного фреймворка.

Zend Framework – это бесплатный PHP фреймворк с открытым исходным кодом.

Его разработка направляется и спонсируется компанией Zend Technologies, также разработавшей и сам язык PHP. Первая версия (Zend Framework 1) была выпущена в 2007; Zend Framework 2, вторая версия этого ПО, была выпущена в сентябре 2012. Zend Framework 3 (или коротко ZF3) была выпущена в июне 2016.

Zend Framework предоставляет следующие возможности:

- разработать веб сайт гораздо быстрее, чем бы это делать на чистом PHP. ZF предоставляет много компонентов, которые могут использоваться как база для создания сайтов;
- более простое взаимодействие с членами команды. Шаблон «Модель-Вид-Контроллер», используемый в ZF, позволяет разделять бизнес-логику и слой презентации, делая структуру кода более однообразной и управляемой;
- масштабирование веб сайта облегчается, используя концепцию модулей. ZF использует термин модуль, позволяющий разделять слабо связанные части сайта, таким образом повторно используя классы моделей, вида и контроллеров в других работах;
- доступ к базе данных в объектно-ориентированном стиле. Вместо того, чтобы напрямую взаимодействовать с БД посредством SQL-запросов, используется Doctrine Object-Relational Mapping (ORM), чтобы управлять структурой и отношениями данных. С Doctrine работает с таблицей БД как с классом на языке PHP, также называемый классом «сущности» (entity), а

строка из таблицы БД представляется экземпляром такого класса. Doctrine позволяет абстрагироваться от типа БД и одинаково работать, например с MySQL и PostgreSQL [18];

- создание безопасных веб сайтов с использованием таких компонентов, как фильтры (filters) и валидаторы (validators) форм, HTML output escapers и крипто алгоритмы, human check (Captcha) и Cross-Site Request Forgery (CSRF) элементы форм.

Для реализации разрабатываемого программного интерфейса была выбрана версия Zend 1.12, так как основной портал института функционирует именно на данной версии. Так же использование более новой версии не позволяет сделать программная и аппаратная составляющая WEB-сервера [6, с. 30-78].

Данная проблема сделала не возможным использовать стандартные библиотеки для реализации протокола авторизации OAuth2.0 для API.

Для решения данной проблемы было решено использовать .Net Core 2.0 с Identity Server 4.

Платформа ASP.NET Core представляет технологию от компании Microsoft, предназначенную для создания различного рода веб-приложений: от небольших веб-сайтов до крупных веб-порталов и веб-сервисов.

С одной стороны, ASP.NET Core является продолжением развития платформы ASP.NET. Но с другой стороны, это не просто очередной релиз.

Выход ASP.NET Core фактически означает революцию всей платформы, ее качественное изменение.

ASP.NET Core может работать поверх кросс-платформенной среды .NET Core, которая может быть развернута на основных популярных операционных системах: Windows, Mac OS X, Linux.

Таким образом, с помощью ASP.NET Core можно создавать кросс-платформенные приложения. Windows в качестве среды для разработки и развертывания приложения до сих пор превалирует, но теперь уже мы не ограничены только этой операционной системой.

То есть можно запускать веб-приложения не только на ОС Windows, но и на Linux и Mac OS. А для развертывания веб-приложения можно использовать традиционный IIS, либо кросс-платформенный веб-сервер Kestrel.

Хотя ASP.NET Core преимущественно нацелено на использование .NET Core, но фреймворк также может работать и с полной версией фреймворка .NET.

Благодаря модульности фреймворка все необходимые компоненты веб-приложения могут загружаться как отдельные модули через пакетный менеджер Nuget [14, 63-152].

В качестве хранилища данных будет использоваться существующая СУБД Oracle.

Современная СУБД Oracle это мощный программный комплекс, позволяющий создавать приложения любой степени сложности.

Ядром этого комплекса является база данных, хранящая информацию, количество которой за счет предоставляемых средств масштабирования практически безгранично.

С высокой эффективностью работать с этой информацией одновременно может практически любое количество пользователей (при наличии достаточных аппаратных ресурсов), не проявляя тенденции к снижению производительности системы при резком увеличении их числа.

Механизмы масштабирования в СУБД Oracle последней версии позволяют безгранично увеличивать мощность и скорость работы сервера Oracle и своих приложений, просто добавляя новые и новые узлы кластера.

Это не требует остановки работающих приложений, не требует переписывания старых приложений, разработанных для обычной одно-машинной архитектуры.

Кроме того, выход из строя отдельных узлов кластера также не приводит к остановке приложения. Данный фактор положительно увеличивается использование микросервисной архитектуры.

Ввиду того, что острая проблема с устаревшим Web сервером не дает функциональной возможности использовать его как сервис авторизации, следует внедрить в действующую корпоративную информационную сеть новый сервер авторизации, который в последующем станет единым окном выделения прав (авторизации пользователей) на использование ресурсов корпоративной сети.

1.5.2 Выбор средств разработки и проектирования

В качестве основной среды разработки API выступило решение от компании JetBrains – PhpStorm.

PhpStorm предоставляет богатый и интеллектуальный редактор кода для PHP с подсветкой кода, расширенной конфигурацией форматирования кода, проверкой на наличие ошибок на лету и умным автодополнением.

Основные возможности данной среды:

- поддержка PHP 5.3, 5.4, 5.5, 5.6, 7.0 и 7.1, включая генераторы, сопрограммы, пространства имен, замыкания, типаж, синтаксис коротких массивов, доступ к члену класса при инстанцировании, разыменовывание массива при вызове функции, бинарные литералы, выражения в статичных вызовах и т.д. PhpStorm может использоваться как для современных, так и для традиционных проектов на PHP;
- автодополнение кода финализирует классы, методы, имена переменных, ключевые слова PHP, а также широко используемые имена полей и переменных в зависимости от их типа;
- поддержка стандартов оформления кода (PSR1/PSR2, Drupal, Symfony2, Zend);
- поддержка PHPDoc. PhpStorm предоставляет соответствующее автодополнение кода, основанное на аннотациях `@property`, `@method` и `@var`;
- детектор дублируемого кода;
- php code sniffer (phpcs), проверяющий код на лету;

- рефакторинги (Переименование, введение переменной / константы / поля, Встраивание переменной);
- поддержка редактирования шаблонов Smarty (подсвечивание синтаксических ошибок, автодополнение функций и атрибутов Smarty, автоматическая вставка парных скобок, кавычек и закрывающих тегов и др.);
- распознавание кода, запакованного в PHAR-архивы.

Данное программное решение является коммерческим решением и предоставляет несколько способов лицензирования данного продукта:

- персональная лицензия (платная, для индивидуальных разработчиков);
- коммерческая лицензия (платная, для компаний и организаций);
- академическая лицензия (бесплатная, для студентов и преподавателей);
- лицензия для образовательных учреждений (бесплатная);
- лицензия для проектов с открытым исходным кодом (бесплатная).

Для данного проекта была выбрана академическая лицензия, которая поставляется для студентов бесплатно, при этом не вводя ограничения функционала.

Второй средой разработки стала Microsoft Visual Studio.

Использование данной среды позволит реализовать сервис авторизации клиентских приложений на сервере ресурсов API корпоративной ИС института.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, приложения с графическим интерфейсом и серверных приложений, в том числе с поддержкой .NET CORE, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows

Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework, Silverlight, Android и IOS .

Выбранная среда Microsoft Visual Studio 2017 Community распространяется бесплатно.

Для работы с объектами данных (таблицы в СУБД Oracle), был выбран программный продукт от JetBrains. Программное обеспечение JetBrains DataGrip представляет собой IDE-инструмент для работы с базами данных MySQL, PostgreSQL, Oracle, SQL Server, Sybase, DB2, SQLite, HyperSQL, Apache Derby и H2. Решение JetBrains DataGrip включает мощный текстовый редактор с мультикурсорами, обеспечивает синтаксическое выделение кода, поддерживает интеграцию с системами контроля версий Git, Subversion и т.д.

Данный продукт обладает некоторыми преимуществами перед своими конкурентами:

- работа с объектами БД и генерация кода: DataGrip предоставляет инструменты для работы с объектами базы данных. Если пользователь создает или изменяет таблицу, добавляет или изменяет колонку, ему пригодится графический интерфейс JetBrains DataGrip. Подобные изменения сопровождаются генерацией соответствующего скрипта: можно сразу выполнить сделанные изменения в базе или скопировать сгенерированный DDL-запрос в редактор и работать уже непосредственно с кодом;

- автодополнение: DataGrip поддерживает автодополнение кода, что ускоряет написание запросов. Когда набирается код, IDE понимает контекст и делает работу за пользователя: не только помогает писать код, зная о ключевых словах и именах объектов БД, но и учитывает зависимости при написании JOIN, подсказывает тип параметров для выполнения функции, описывает структуру таблицы в предложениях INSERT;

- поиск по коду и переименование: IDE понимает, какие объекты базы используются в коде: если переименовать объект в запросе, то же случится и в базе. Есть поиск использования переменной или объекта (колонки, таблицы) в запросе, а также возможность перехода от

использования к месту объявления. Если пользователь применяет то же самое к объекту, который уже был создан в базе, курсор отправит его в окно структуры базы данных. А если в запросе использовано имя объекта, которого нет в базе, – например, ошиблись с названием столбца или таблицы, – IDE сообщит о проблеме и предложит возможные решения;

- работа с данными: Табличный редактор в DataGrip может фильтровать данные. Текстовый поиск по таблице тоже умеет фильтровать, также доступна навигация по данным – при наличии связи по внешним ключам можно попадать в те строчки таблиц, которые ссылаются на эти по foreign key, и наоборот;

- выполнение запросов: для выполнения части запроса надо выделить код и запустить его. В окне результата запроса доступны многие функции Table Editor, например он позволяет изменять данные и в нем работает текстовый поиск. Для сравнения двух результатов используются смежные окна.

2 Проектная часть

2.1 Разработка функционального обеспечения

Выявленные недостатки в работе действующего API приводят к необходимости построения модели «Как должно быть». Задачей описания модели как должно быть, является нахождение мер блокирования отрицательного влияния неудовлетворительных бизнес-факторов, найденных при анализе. Компонентная диаграмма КИС «Как должно быть» указана в приложении Б.1 и диаграмма развертывания КИС «Как должно быть» представлены в приложении Б.2. Предложенная модель как должна быть полностью устраняет главные недостатки действующей модели:

- сильную связанность компонентов ИС;
- взаимосвязь между компонентами реализуется не по регламентируемому принципу;
- образовательный портал выступает в качестве промежуточного звена в обмене информацией между сторонними приложениями и КИС;
- разобщенность применимых технологий для разработки.

Так же данная модель предполагает, что все приложения, которые в дальнейшем будут интегрироваться с КИС института будут иметь единую точку получения данных, что позволяет повысить безопасность всей ИС в целом, так как появляется возможность точечного контроля приложений.

2.2 Разработка информационного обеспечения

2.2.1 Используемые классификаторы и системы кодирования

При реализации проекта для обеспечения уникальности объектов в пределах класса, где применяется порядковая система кодирования объектов.

При этом все множество объектов предварительно не упорядочивается, в результате чего всю работу по обеспечению максимальной скорости поиска объекта и определения его принадлежности во время сложной логической обработки с использованием имеющихся данных берет на себя СУБД.

Таким образом, некоторые таблицы имеют в своем составе автоинкрементное поле – идентификатор. Это позволяет не контролировать процессы генерации и присвоения уникальных кодов для записей программно, а полностью предоставить этот процесс под контроль СУБД. Все даты представляются в стандартном для России формате «ДД.ММ.ГГГГ», где ДД – день месяца от 0 до 31, ММ – месяц от 0 до 12, ГГГГ – год в четырехзначном представлении.

Вся информация в базе данных представлена в кодировке UTF-8 – распространённой кодировке, реализующей представление Юникода, совместимое с 8-битным кодированием текста, позволяющая представить знаки практически всех письменных языков.

2.2.2 Характеристика нормативно-справочной и входной оперативной информации

Вся справочная информация информационного пространства может быть представлена в виде объектов, порожденных на основе классов, имеющих в своей структуре следующие свойства: идентификатор объекта, собственно справочную информацию и ряд специфических, дополнительных свойств.

Методами у таких объектов могут быть добавление новой справочной информации, изменение существующей и удаление.

Для обеспечения функционирования информационных систем и портала в Институте реализована структура справочников. Все методы объектов реализованы с помощью программных средств организации, а также с использованием существующей системы управления содержимым

сайта.

Для исследуемой предметной области можно выделить следующие основные справочники:

- студенты;
- преподаватели;
- группы;
- аудитории.

2.2.3 Характеристика результатной информации

В качестве результатной информации работы API будут выступать:

- списки групп;
- список аудиторий;
- личная информация о студенте;
- сведения об успеваемости;
- расписание занятий;
- расписание консультаций;
- список предметных комиссий;
- список пересдач;
- список звонков;
- список преподавателей (ППС и преподавателей для расписания);
- сведения для подачи заявки на получение справки.

Все перечисленные списки данных перед отправкой на клиентское приложение должны быть определённым образом преобразованы к стандартным форматам обмена данными.

На данный момент самыми популярными форматами являются JSON и XML. К преимуществам данных форматов относят:

Преимущества JSON:

- меньший размер сообщения;
- больше структурной информации в документе;
- можно легко различить число 1 и строку «1» как числа, строки (и логические значения) представлены в JSON по-разному;
- может легко различать отдельные элементы и коллекции размером один (с использованием массивов JSON);
- легкое представление нулевого значения;
- легко потребляется JavaScript.

Преимущества XML:

- пространства имен позволяют использовать стандартные структуры;
- лучшее представление для наследования;
- стандартные способы выражения структуры документа: XML-схема, DTD и т.д.;
- стандарты анализа: DOM, SAX, StAX;
- стандарты для запросов: XQuery и XPath;
- стандарты для преобразования документа: XSLT.

При каждом обращении к API за данными клиентское приложение имеет возможность передать параметр, отвечающий за формат ответа.

По умолчанию данные вернутся в JSON-формате [15].

2.2.4 Структура результатной информации

Структура результативной информации имеет относительный формат, потому что сам формат данных может выбирать клиентское приложение, и структура ответа может меняться в зависимости от версии API.

API 1.0 имеет возможность передачи результативной информации в формате JSON и XML.

Базовыми справочниками являются аудитории, группы, преподаватели.

Самым востребованным модулем API является Модуль «Расписание», характеристика его выходной информации представлена на рисунке (включенные параметры: группа 1245, дата 7 февраля 2018, тип расписания группа, временной интервал 1 день).

На рисунке 2.1 представлена структура справочника Аудитории в API 1.0.

<pre> { "Error": null, "Data": [{ "ID": "191", "NAME": "библиотека 1" }, { "ID": "193", "NAME": "библиотека 2" }, { "ID": "117", "NAME": "Стадион 1" }, { "ID": "116", "NAME": "Стадион 2" }, { "ID": "228", "NAME": "102" }, { "ID": "126", "NAME": "103Б" }, { "ID": "125", "NAME": "104Б" }, { "ID": "97", "NAME": "105Б" }, { "ID": "41", "NAME": "106" }] } </pre>	<pre> <?xml version="1.0" encoding="UTF-8" ?> <root> <Error /> <Data> <ID>191</ID> <NAME>библиотека 1</NAME> </Data> <Data> <ID>193</ID> <NAME>библиотека 2</NAME> </Data> <Data> <ID>117</ID> <NAME>Стадион 1</NAME> </Data> <Data> <ID>116</ID> <NAME>Стадион 2</NAME> </Data> <Data> <ID>228</ID> <NAME>102</NAME> </Data> <Data> <ID>126</ID> <NAME>103Б</NAME> </Data> <Data> <ID>125</ID> <NAME>104Б</NAME> </Data> <Data> <ID>97</ID> <NAME>105Б</NAME> </Data> <Data> <ID>41</ID> <NAME>106</NAME> </Data> <Data> <ID>85</ID> <NAME>106Б</NAME> </Data> </root> </pre>
---	--

Рисунок 2.1 – Структура справочника Аудитории в API 1.0 (JSON и XML формат)

На рисунке 2.2 представлена структура выходной информации Модуля «Расписание» в формате JSON.

```
{
  "Error": null,
  "Data": [
    {
      "Date": "07.02.2018 00:00",
      "Lessons": [
        {
          "Number": "1",
          "Room": "31",
          "LessonName": "Лекция №6",
          "DisciplineName": "Предметно-ориентированные экономические информационные системы",
          "Teachers": [
            "e3b75a0b-884f-11e5-95d4-00259001ddd3"
          ],
          "Groups": [
            "2798",
            "2574"
          ]
        }
      ],
    },
    {
      "Number": "2",
      "Room": "46",
      "LessonName": "Лабораторная работа №3",
      "DisciplineName": "Сертификация 1С:Профессионал",
      "Teachers": [
        "e3b75a0b-884f-11e5-95d4-00259001ddd3"
      ],
      "Groups": [
        "2798",
        "2574"
      ]
    },
    {
      "Number": "3",
      "Room": "27",
      "LessonName": "Лекция №3",
      "DisciplineName": "Информационная безопасность",
      "Teachers": [
        "a69dd5e0-ade5-11dc-8c11-0016e652202c"
      ],
      "Groups": [
        "2798",
        "2574"
      ]
    }
  ],
  "TypeHour": "Workday"
}
```

Рисунок 2.2 – Структура выходной информации Модуля «Расписание» в формате JSON

На рисунке 2.3 представлена Структура выходной информации Модуля «Расписание» в формате XML.

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <Error />
  <Data>
    <Date>07.02.2018 00:00</Date>
    <Lessons>
      <Number>1</Number>
      <Room>31</Room>
      <LessonName>Лекция №6</LessonName>
      <DisciplineName>Предметно-ориентированные экономические информационные системы
        </DisciplineName>
      <Teachers>e3b75a0b-884f-11e5-95d4-00259001ddd3</Teachers>
      <Groups>2798</Groups>
      <Groups>2574</Groups>
    </Lessons>
    <Lessons>
      <Number>2</Number>
      <Room>46</Room>
      <LessonName>Лабораторная работа №3</LessonName>
      <DisciplineName>Сертификация 1С:Профессионал</DisciplineName>
      <Teachers>e3b75a0b-884f-11e5-95d4-00259001ddd3</Teachers>
      <Groups>2798</Groups>
      <Groups>2574</Groups>
    </Lessons>
    <Lessons>
      <Number>3</Number>
      <Room>27</Room>
      <LessonName>Лекция №3</LessonName>
      <DisciplineName>Информационная безопасность</DisciplineName>
      <Teachers>a69dd5e0-ade5-11dc-8c11-0016e652202c</Teachers>
      <Groups>2798</Groups>
      <Groups>2574</Groups>
    </Lessons>
    <TypeHour>Workday</TypeHour>
  </Data>
</root>
```

Рисунок 2.3 – Структура выходной информации Модуля «Расписание» в формате XML

2.2.5 Информационная модель и ее описание

Информационная модель – это модель объекта, представленная в виде информации описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта и позволяющая путём подачи на модель информации об

изменениях входных величин моделировать возможные состояния объекта [8, с. 123-125].

Разработка информационной модели является последним этапом при проектировании информационной системы. Данный процесс представляет собой построение двух типов моделей логической и физической.

Для реализации данной ИС требовалось лишь создать БД для работы сервера OAuth2, так как все данные для обработки их API уже находятся в БД института и вносить изменения в их модель не потребовалось [9, с. 73-75].

Сервер авторизации спроектирован на базе Identity Server 4, который работает с Entity Framework Core и позволяет использовать все плюсы данного фреймворка.

Entity Framework Core (EF Core) представляет собой объектно-ориентированную, легковесную и расширяемую технологию от компании Microsoft для доступа к данным. EF Core является ORM-инструментом (object-relational mapping – отображения данных на реальные объекты). То есть EF Core позволяет работать базами данных, но представляет собой более высокий уровень абстракции: EF Core позволяет абстрагироваться от самой базы данных и ее таблиц и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который предлагает Entity Framework, работа ведется уже с объектами.

Entity Framework Core поддерживает множество различных систем баз данных. Таким образом, мы можем через EF Core работать с любой СУБД, если для нее имеется нужный провайдер [17].

По умолчанию на данный момент Microsoft предоставляет ряд встроенных провайдеров: для работы с MS SQL Server, для SQLite, для PostgreSQL. Также имеются провайдеры от сторонних поставщиков, например, для MySQL. Так как институт имеет подписку на MS SQL Server, то в качестве рабочей СУБД был выбран продукт от компании Microsoft [10,

с. 125-365].

Данный подход имеет возможность использовать технологию code first, которая предполагает, что проектирование схемы БД ведется на абстрактном уровне объектов без привязки в конкретной СУБД.

Для реализации информационной модели при помощи данных инструментов потребовалось создать так называемые миграции, которые при первоначальном запуске приложения создадут необходимую структуру БД (Приложение В).

2.3 Разработка программного обеспечения

2.3.1 Описание серверных программных модулей

API (программный интерфейс приложения, интерфейс прикладного программирования) (англ. application programming interface) – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах. Функциональность определяется приложением (модуль, библиотека), при этом API позволяет абстрагироваться от того, как именно эта функциональность реализована.

API библиотеки функций и классов включает в себя описание сигнатур и семантики функций, таких как – REST (Representational State Transfer – «передача состояния представления») – архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы. В определённых случаях (основанные на данных) это приводит к повышению производительности и упрощению архитектуры. В широком смысле, компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине. REST является альтернативой RPC.

REST API подразумевает под собой простые правила:

- каждый URL является ресурсом;
- при обращении к ресурсу методом GET возвращается описание этого ресурса;
- метод POST добавляет новый ресурс;
- метод PUT изменяет ресурс;
- метод DELETE удаляет ресурс.

Эти правила предоставляют простой CRUD интерфейс для других приложений, взаимодействие с которым происходит через протокол HTTP.

REST API интерфейс очень удобен для межпрограммного взаимодействия. На рисунке 2.4 представлены серверные модули API.

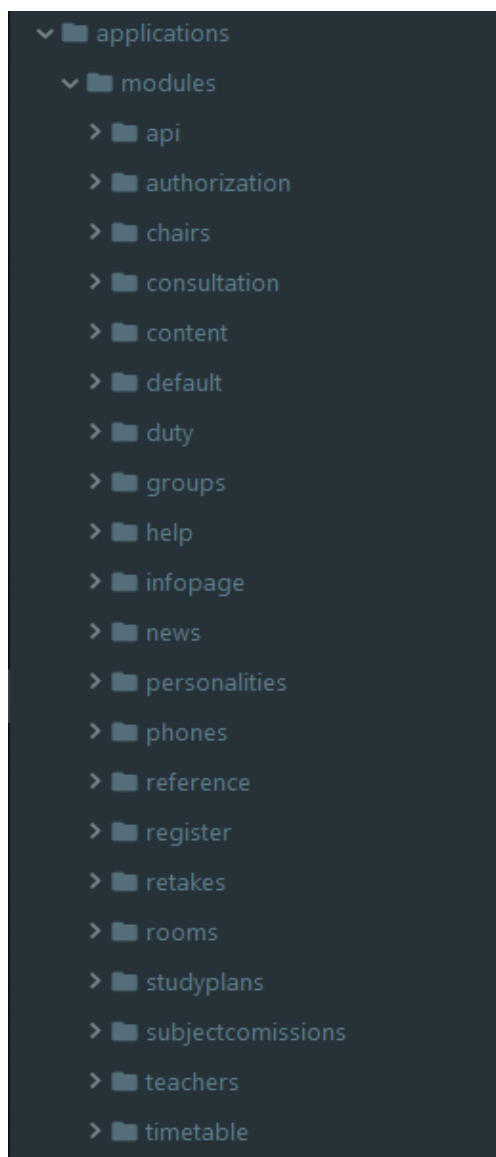


Рисунок 2.4 – Серверные модули API

Каждый серверный модуль API предполагает реализацию работы с одним ресурсом.

К примеру: модуль `rooms` полностью реализует обработку данных о аудиториях института, а Модуль `reference` реализует функцию заказа справок студентами. Преимущество модульного подхода к разработке в том, что каждый модуль не зависим от других и реализует свой микросервис и если один модуль находится в состоянии отказа, то все остальные продолжают работать в штатном режиме.

Так же при данном подходе облегчается версионированность всего API, для данной функции каждый микросервис обладает своей абстрактной фабрикой, которая в зависимости от параметров сама определит требуемую версию API и вернет ответ в нужном формате. Модуль авторизации сервера OAuth 2 представлен на рисунке 2.5.

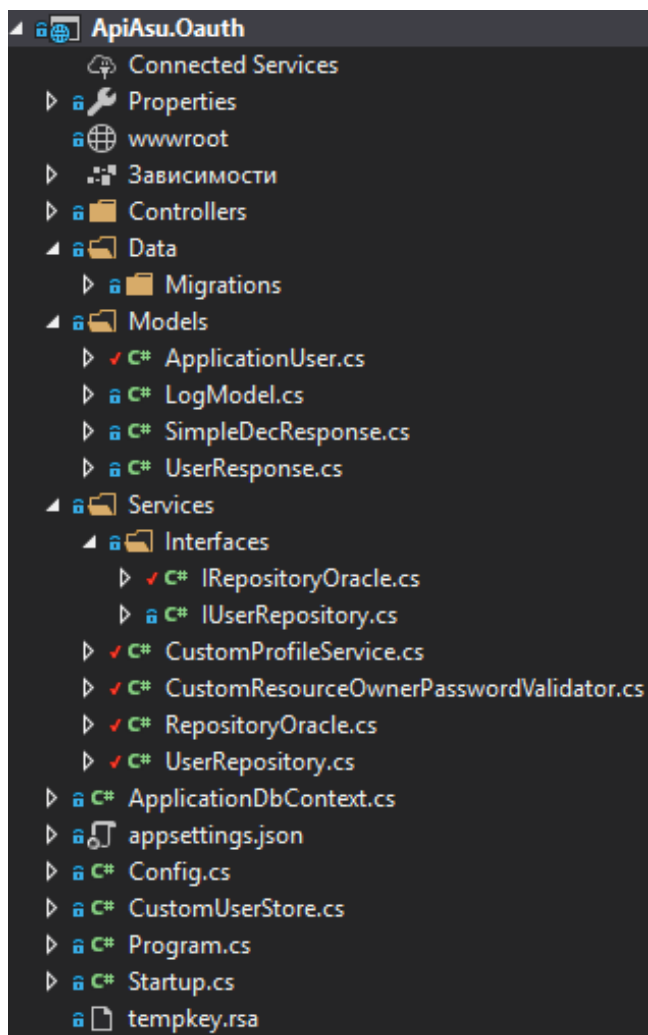


Рисунок 2.5 – Модуль авторизации сервера OAuth 2.0

Решение серверного модуля OAuth 2 реализовано на базе .Net Core 2, IdentityServer4. Модуль состоит из:

- data – подмодуль отвечающий за хранение миграций базы данных, при каждом изменении структуры таблиц имеется возможность создать новую миграцию, которая при следующем запуске перестроит БД в соответствии с новыми потребностями;
- models – подмодуль хранящий базовые классы решения;
- services – подмодуль реализующий всю логику взаимодействия решения с источником данных (БД);
- program.cs startup.cs – реализуют первоначальную настройку решения и интеграцию Identity Server 4;
- customuserstore.cs – переопределяет хранилище пользователей и систему авторизации на LDAP
- остальные модули несут в себе настроечную среду или внедрение зависимостей в данное решение.

2.3.2 Описание клиентских программных модулей

При разработке API клиентские программные модули не предусматриваются, но хорошим тоном считается создавать так называемые «обертки» для разных языков программирования.

Учитывая тот момент, что студентам института преподают курс программирования на С#, было принято решение о разработке готовой библиотеки для работы с программным интерфейсом Рубцовского института (филиала) АлтГУ для платформы .NET и ее продолжением .NET CORE2.

Данное решение имеет положительных моментов на образовательный процесс обучающихся, а именно:

- расширяет возможности лабораторного практикума студентов;
- дает возможность студентам использовать данную библиотеку в своих курсовых и дипломных проектах, для обмена данными с

корпоративной информационной системой института.

2.3 Компьютерно-сетевое обеспечение

API имеет микросервисную архитектуру. Серверная часть приложения использует существующее сетевое оборудование РИ (филиала) АлтГУ и специально добавленное для реализации API оборудование. Структура сети также не подвергается изменениям, так как её характеристики полностью удовлетворяют предъявленным требованиям. Используется следующее оборудование:

1. Сервер WWW. Выполняет функции базового WEB-сервера со следующим набором сервисов: HTTP, FTP, DNS, MAIL, PROXY, APACHE, MySQL. Технические характеристики: 2xP4Xeon, 3,0GHz, 4x2Gb, RAID1 SATA 2x500Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux Server Edition.

2. Сервер ORACLE. Выполняет функции сервера БД для хранения информации для портала и информации необходимой для учебного процесса. Технические характеристики: 2xP4Xeon, 2,8GHz, 24Gb, RAID5 SATA 4x200Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux Server Edition.

3. Сервер Oauth2.0. Сконфигурированный под сервер авторизации компьютер – далее его характеристики.

2.4 Обеспечение информационной безопасности

Информационная безопасность (ИБ) – состояние защищённости информационной среды общества, обеспечивающее её формирование, использование и развитие в интересах отдельных граждан, организаций и государства. Персональные данные – любая информация, относящаяся к определенному или определяемому на основании такой информации

физическому лицу (субъекту персональных данных), в том числе его фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация.

Информационная система персональных данных – информационная система, представляющая собой совокупность персональных данных, содержащихся в базе данных, а также информационных технологий и технических средств, позволяющих осуществлять обработку таких персональных данных с использованием средств автоматизации или без использования таких средств [11, с. 210-300].

Классификация информационной системы персональных данных

Определение категории обрабатываемых персональных данных:

- первая категория – персональные данные, касающиеся расовой, национальной принадлежности, политических взглядов, религиозных и философских убеждений, состояния здоровья, интимной жизни;
- вторая категория – персональные данные, позволяющие идентифицировать субъекта персональных данных и получить о нём дополнительную информацию, за исключением персональных данных, относящихся к категории 1;
- третья категория – персональные данные, позволяющие идентифицировать субъекта персональных данных;
- четвертая категория – обезличенные и (или) общедоступные персональные данные.

По результатам анализа исходных данных ИС присваивается один из следующих классов:

- класс 1 – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к значительным негативным последствиям для субъектов персональных данных;
- класс 2 – информационные системы, для которых нарушение

заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к негативным последствиям для субъектов персональных данных;

- класс 3 – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к незначительным негативным последствиям для субъектов персональных данных;

- класс 4 – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, не приводит к негативным последствиям для субъектов персональных данных .

Для защиты от постороннего вторжения предусматриваются определенные меры безопасности. В ИС это осуществляется программными средствами, которые выполняют следующие функции:

- идентификация субъектов и объектов;
- разграничение доступа к ресурсам и информации;
- контроль и регистрация действий.

Процедура идентификации и подтверждения подлинности предполагает проверку является ли субъект, осуществляющий доступ, или объект, к которому осуществляется доступ, тем за кого себя выдает. Здесь используются следующие методы:

- простые, сложные или одноразовые пароли;
- средства анализа индивидуальных характеристик субъекта (геометрических данных);
- ключи, жетоны, магнитные карты;
- обмен вопросами и ответами с администратором системы;
- специальные идентификаторы и контрольные суммы для программ и данных.

После процедуры идентификации пользователь получает доступ к системе, где защита от несанкционированного доступа реализуется на трех

уровнях:

- на уровне аппаратуры;
- на уровне программного обеспечения;
- на уровне данных.

Защита информации на 1 и 2 уровнях предусматривает управление доступом к различным вычислительным ресурсам (отдельным устройствам, операционной системы, служебным или личным программам пользователя).

Защита информации на уровне данных направлена на защиту информации в процессе обращения к ней, в процессе работы с файловой структурой ЭВМ, в процессе передачи информации по каналам связи. [4, с. 78-122]

В общем, комплекс программно-технических средств и организованных решений по защите информации от несанкционированного доступа характеризуется следующими действиями:

- логическое управление доступом;
- регистрацией, контролем и учетом работы;
- применением криптографических средств;
- обеспечением целостности информации.

Выделяют следующие формы контроля и управления доступом:

- предотвращение доступа к жесткому диску, каталогам и файлам;
- установка привилегий к группам файлов;
- защита от модификаций и изменений;
- защита от уничтожения;
- предотвращение копирования.

Под средствами защиты от копирования понимается такие средства, которые обеспечивают выполнение программой своих функций только при опознании некоторого уникального не копированного элемента. Таким элементом может быть ключевой сменный носитель, часть оборудования или специальное устройство, подключаемое к компьютеру [13, с. 56-203].

Защита от копирования реализуется выполнением ряда функций:

- идентификация среды, из которой запускается система;
- проверка подлинности среды, из которой произошел запуск;
- немедленная реакция на запуск из несанкционированной среды;
- обязательная регистрация санкционированного копирования;
- противодействие изучению алгоритмов работы системы [27].

Область информационной безопасности является компетенцией отдела программного и технического обеспечения. Данным отделом регулярно производятся плановые профилактические работы всех программно-аппаратных систем Института.

Стоит подробно рассмотреть схему получения разрешения приложения для доступа к файлам сервера. Схема представлена на рисунке 2.6.

Главной особенностью схемы авторизации OAuth 2.0 является то, что клиентское приложение не имеет прямого доступа к авторизационным и идентификационным данным пользователя, а сам процесс авторизации происходит на сервере авторизации Рубцовского института (филиала) АлтГУ.

Сам процесс обмена данными и идентификации пользователя в клиентском приложении, происходит при помощи токенов и ключей приложения ACCEESSTOKEN, RESOURCETOKEN и REFRESHTOKEN, а так же ключей PUBLICKEY и SECRETKEY выданных при регистрации приложения.

Процесс авторизации пользователя и получения данных имеет несколько этапов, а именно:

1. Инициализации процесса авторизации с посылы публичного ключа приложения к OAUTH-SERVER по определенному URL.
2. Пользователь загружается стандартная форма авторизации на ресурсах поставщика данных.
3. После ввода авторотационных и идентификационных данных пользователя, OAUTH-SERVER производит его авторизацию, и в случае успешного исхода авторизует пользователя и создает ACCEESSTOKEN для

клиентского приложения запросившего авторизацию. В случае не авторизации пользователя формируется ответ с ошибкой авторизации пользователя.

4. При успешной авторизации пользователя клиентскому приложению передается выданный `ACCESSTOKEN` и `URL` для получения `RESOURCETOKEN` и `REFRESHTOKEN` на следующем этапе. В случае неудачи авторизации пользователя, клиентскому приложению пересылается данные об ошибке сформированные на предыдущем этапе.

5. Для получения `RESOURCETOKEN` и `REFRESHTOKEN` клиентское приложение отправляет свой `SECRETOKEN` и полученный `ACCESSTOKEN` на данное соединение.

6. `OAUTH-SERVER` после авторизации клиентского приложения по полученным данным отправляет `REFRESHTOKEN` (это автоматическая замена пере-аутентификации пользователя, которая не требует лишней раз вводить логин и пароль) и `RESOURCETOKEN` для последующего обмена данными. В случае если произошла ошибка авторизуемого соединения, формируется объект ошибки и отправляется клиентскому приложению.

Так же имеются возможности, которые могут потребоваться в зависимости от различных ситуаций, а именно:

1. Обновление `RESOURCETOKEN`. `RESOURCETOKEN` имеет определенный период существования, при окончании которого токен устаревает и для дальнейшего обмена данными пользователю придется пройти процедуру авторизации заново. Для устранения этого недостатка на 6 этапе нам передается `REFRESHTOKEN` при помощи которого мы можем обновить токен ресурсов. Процесс обновления выглядит следующим образом: на стороне клиентского приложения срабатывает событие об окончании времени жизни `RESOURCETOKEN`. Клиентское приложение посылает запрос на обновление токена ресурса, с выданным ему ранее `REFRESHTOKEN`. Сервер авторизации идентифицирует `REFRESHTOKEN` и в случае его успеха, генерирует новый `RESOURCETOKEN` и пересылает его

обратно приложению. В случае неудачи, формируется объект ошибки и отправляется клиентскому приложению.

2. Обновление REFRESHTOKEN. REFRESHTOKEN так же имеет определенный период существования, при окончании которого токен устаревает и при помощи него невозможно получить новый RESOURCETOKEN. Для получения нового REFRESHTOKEN OAUTH-SERVER пересылается старый REFRESHTOKEN и дополнительные параметры. В случае успеха клиентскому приложению выдается новый REFRESHTOKEN, старый при этом перестает быть действительным. В случае неудачи, формируется объект ошибки и отправляется клиентскому приложению.

3. Обмен данными. Клиентское приложение делает запрос по определенному URL передавая дополнительные параметры и выданный ранее RESOURCETOKEN. RESOURCE_SERVER проверяет действительность RESOURCETOKEN и в случае успеха формирует объект данных или объект ошибки запроса данных. В случае недействительности RESOURCETOKEN формируется объект с сообщением об ошибке авторизации клиентского приложения [16].

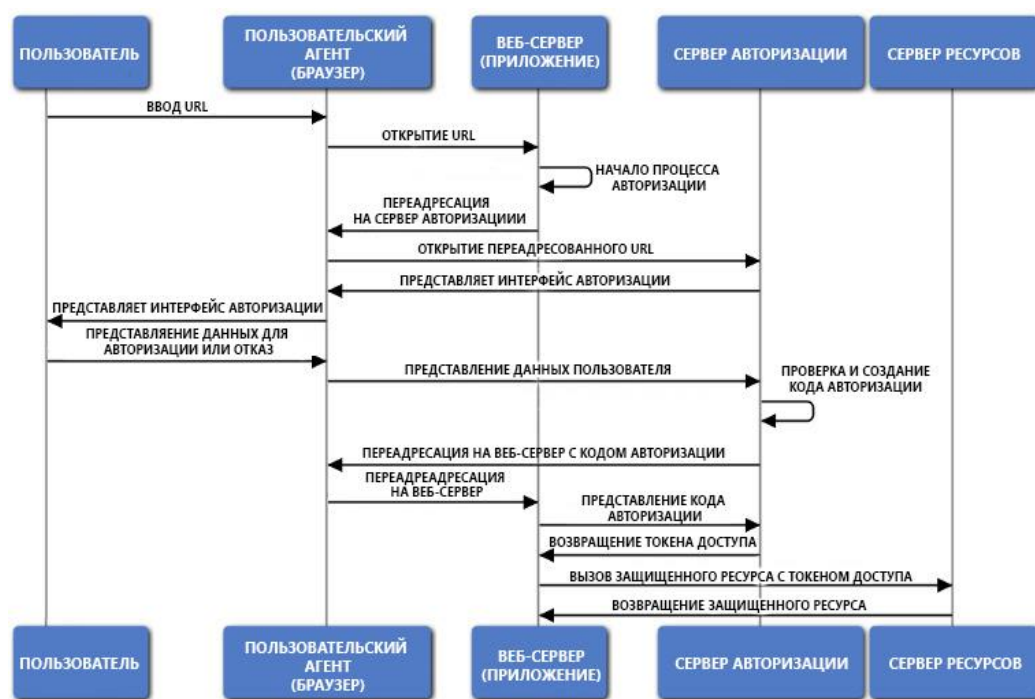


Рисунок 2.6 – Процесс авторизации по протоколу OAuth 2.0

Рубцовском институте (филиале) АлтГУ, доступ к информационным системам управления вузом осуществляется на основании приказа директора «Об оптимизации доступа к БД, в целях защиты информации», в котором указаны лица, имеющие полный доступ на внесение и изменение информации и лица, имеющие доступ для чтения информации, в случае ее использования для принятия решения.

Полный доступ на изменение информации имеют только сотрудники отдела по работе со студентами, у методистов кафедр, заведующих кафедрами, директора и его заместителей имеется доступ на чтение информации.

Защита информации БД на программном уровне информационно-образовательного портала осуществляется стандартными методами, такими как: проверка введенной пользователем информации и проверка корректности принимаемых от клиента данных.

3 Оценка эффективности внедрения информационных систем

3.1 Общие положения

При разработки любой новой информационной системы для организации следует учитывать эффективность от ее внедрения.

Оценка эффективности информационных систем для каждого отдельного предприятия не разрабатывается, если нет в наличии оценок эффективности, которые представляют собой комплексную информационную систему. В то же время отдельные составляющие информационных систем тоже не рассматриваются без такой оценки.

Эффективность ИС – это свойство системы выполнять поставленную цель в заданных условиях использования и с определенным качеством. Эта характеристика отражает:

- действенность системы, т.е. степень соответствия ИС своему назначению (прагматическая эффективность);
- техническое совершенство ИС (техническая эффективность);
- простоту, технологичность разработки и создание системы (технологическая эффективность);
- удобство использования и обслуживания системы (эксплуатационная эффективность);
- улучшение и облегчение условий труда, изменение его содержания, развитие творческих функций, способностей и потребностей людей, преодоление существенных различий в труде и др. (социальная эффективность);
- экономическую целесообразность внедрения ИС, т.е. целесообразность произведенных на создание и функционирование системы затрат (экономическая эффективность).

Понятие эффективности связано с получением некоторого полезного результата – эффекта использования.

В соответствии с ГОСТ Р ИСО 9000-2001, эффективность функционирования ИС определяется соотношением результата (эффекта) и затраченными ресурсами. Приведенной оценкой затрат ресурсов выступает их стоимость. Затраты на функционирование ИС состоят из:

- стоимости приобретения программной платформы;
- стоимости доработки;
- стоимости внедрения;
- стоимости системного и вспомогательного программного обеспечения;
- стоимости аппаратного и сетевого обеспечения ИС;
- количества циклов (лет) эксплуатации;
- стоимости эксплуатации.

Основные задачи, стоящие при создании ИС – минимизация стоимости и обеспечение требуемого качества ИС.

Качество – это совокупность свойств системы, обуславливающих возможность ее использования для удовлетворения определенных потребностей пользователей в соответствии с ее назначением.

Основными показателями качества ИС являются:

- надежность;
- достоверность;
- безопасность.

Надежность – свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных условиях применения.

Надежность ИС является средством обеспечения актуальной и достоверной информации на выходе системы.

Достоверность функционирования – свойство системы,

обуславливающее безошибочность производимых ею преобразований информации. Достоверность функционирования ИС полностью определяется и измеряется достоверностью ее результатной информации.

Безопасность – свойство, заключающееся в способности системы обеспечить конфиденциальность и целостность информации, т.е. защиту информации от несанкционированного доступа.

В любой сфере человеческой деятельности оценка эффективности внедрения любой новой техники, технологий и ИС осуществляется с помощью множества показателей. К ним относятся показатели прагматической, технической, эксплуатационной, социальной и экономической эффективности.

3.2 Показатели эффективности

Оценка эффективности внедрения любой новой технологии, информационных систем осуществляется с помощью множества показателей.

Среди показателей прагматической эффективности выделяют:

- показатели достоверности преобразования информации;
- показатели безопасности информационной системы;
- показатели точности вычислений и преобразования информации;
- показатели полноты формирования системой результатной информации;
- показатели оперативности.

С помощью показателей технической эффективности можно оценить техническое совершенство системы, оценить научно-технический уровень организации и функционирования системы [12, с. 100-189].

В качестве показателей эксплуатационной эффективности могут выступать показатели надежности, функциональность системы, количество обслуживаемых абонентов, производительность, пропускная способность, тактовая частота, временные задержки, емкость памяти, эксплуатационные

характеристики и т.п. Показателями социальной эффективности являются образ и качество жизни, отражающееся в конечном итоге в продолжительности жизни человека и всего населения страны.

Обобщающими показателями эффективности ИС являются показатели экономической эффективности. Обычно в качестве показателей экономической эффективности используются:

- годовой экономический эффект;
- коэффициент экономической эффективности капитальных вложений;
- срок окупаемости капитальных вложений;
- трудоемкость обработки информации;
- эксплуатационная стоимость затрат;
- расчет текущих затрат пользователя;
- экономия текущих затрат при автоматизации;
- годовая экономия затрат на материалы [28].

Экономический эффект – это результат внедрения какого-либо мероприятия, выраженный в стоимостной форме, в виде экономии от его осуществления. Основными источниками экономии являются:

- улучшение показателей их основной деятельности, происходящее в результате использования программного изделия;
- повышение технического уровня, качества и объёмов вычислительных работ;
- увеличение объёмов и сокращение сроков переработки информации;
- повышение коэффициента использования вычислительных ресурсов, средств подготовки и передачи информации;
- уменьшение численности персонала, занятого обработкой исходных данных, переработкой и получением необходимой информации;
- снижение затрат на эксплуатационные материалы.

Срок окупаемости – показатель эффективности использования

капиталовложений, представляет собой период времени, в течение которого произведенные затраты на программные изделия окупаются полученным эффектом.

Частные показатели экономической эффективности необходимы для оценки частного экономического эффекта, получаемого по отдельным источникам экономии.

Например: сокращение доли неквалифицированного и ручного труда на объекте экономики (ОЭ), сокращение материальных и энергетических затрат (повышение наукоемкости продукции), повышение производительности труда работников ОЭ, снижение уровня «бумажного» документооборота, повышение скорости производственных и экономических процессов на ОЭ и др.

3.3 Расчет экономической эффективности

Прежде чем осуществлять внедрение ИС, необходимо дать экономическое обоснование целесообразности внедрения. Экономическая целесообразность означает, что создание ИС должно предусматривать выбор таких проектных решений, которые при условии достижения поставленных целей и задач, обеспечивают минимизацию затрат финансовых, материальных и трудовых ресурсов.

Существует несколько показателей экономической эффективности, по величине которых можно судить об экономической целесообразности внедрения разработанной системы: затраты на проектирование и реализацию программы, затраты на использование программного продукта, показатель экономической эффективности от внедрения программы. Затраты на проектирование и реализацию подсистемы $Z_{общ}$ «Аналитика вуза» состоят из затрат на покупные материалы, зарплаты программиста и проектировщика. Рассмотрим их подробнее. Затраты на покупные материалы определяются по следующей формуле:

$$З_{mi} = Ц_{едi} \cdot N_{mi} \quad (3.1)$$

где $З_{mi}$ – затраты на i -й материал, $Ц_{едi}$ – цена за единицу i -го материала, N_{mi} – количество i -го материала.

Общие затраты на материалы определяются по следующей формуле:

$$З_m = \sum З_{mi} \quad (3.2)$$

где $З_m$ – общие затраты на материалы.

Смета затрат на материалы представлена в таблице 3.1.

Таблица 3.1 – Материалы, затраченные на проектирование модуля сопряжения

Наименование материала	Количество (шт.)	Цена за единицу, (руб.)	Затраты, (руб.)
Бумага формата А4	200	0.40	80
Ручка шариковая	5	15	75
Карандаш	5	5	25
Картридж для принтера	5	100	500
ИТОГО			680

Таким образом, общие затраты на материалы составляют 680 руб.

Затраты на заработную плату определяются по следующей формуле:

$$ЗП_{основ} = C_{час.прог}^m \cdot ФРВ_{прог} + C_{час.проект}^m \cdot ФРВ_{проект} \quad (3.3)$$

где $ЗП_{основ}$ – затраты на выплату заработной платы, $C_{час.прог}^m$ – часовая тарифная ставка программиста на предприятии, $C_{час.проект}^m$ – часовая тарифная ставка проектировщика на предприятии, $ФРВ_{прог}$ – фонд рабочего, времени программиста, приходящийся на создание программного

продукта, $\Phi RB_{проект}$ – фонд рабочего времени проектировщика, приходящийся на создание программного продукта.

Согласно исходным данным таблицы 3.2 основная заработная плата проектировщика и программиста составляет:

$$ЗП_{основ} = 89 * 168 + 72 * 112 = 14952 + 8064 = 23016 \text{ руб.} \quad (3.4)$$

Таблица 3.2 – Исходные данные для расчета заработной платы программиста и проектировщика

Показатель	Обозначение	Значение
Тарифная ставка проектировщика, (руб.)	$C_{час.проект}^m$	72
Тарифная ставка программиста, (руб.)	$C_{час.прог}^m$	89
Фонд рабочего времени программиста на реализацию программы, (часы)	$\Phi RB_{прог}$	168
Фонд рабочего времени проектировщика на проектирование программы, (часы)	$\Phi RB_{проект}$	112

Смета затрат на проектирование модуля и ее реализацию представлена в таблице 3.3.

Таблица 3.3 – Смета затрат на проектирование модуля сопряжения

Статьи затрат	Сумма (руб.)
Покупные материалы	680
Основная заработная плата	23016
ИТОГО	$З_{общ} = 23\,696$

Суммарные затраты на разработку API составили 23 696 рублей. Что бы оценить экономическую эффективность от внедрения программного интерфейса следует рассмотреть главную его задачу: стандартизация и унификация обмена данными для всех приложений, работающих и внедряемых в корпоративную ИС. Эффективность данного проекта

заключается в том, что для всех вновь создаваемых приложений часть серверного функционала будет уже реализована, и статьи затрат на его разработку будут отсутствовать или будут минимизированы, доработкой API.

Следовательно, каждая новая ИС для Рубцовского института будет стоить на 23 696 рублей дешевле.

3.4 Расчет управленческой эффективности

Разработанный программный интерфейс несет в себе не только положительный экономический эффект, но он так же увеличивает эффективность внедрения новых информационных систем. Самым главным положительным фактором в качестве управленческих решений API несет в себе скорость разработки новых прикладных решений для института, которые будут призваны усовершенствовать существующую корпоративную информационную систему.

ЗАКЛЮЧЕНИЕ

Целью выпускной квалификационной работы являлась разработка программного интерфейса интеграции программных приложений с корпоративной ИС Рубцовского института.

Для достижения поставленной цели были решены следующие задачи:

- проведен подробный анализ предметной области;
- построена объектная модель существующей организации доступа к корпоративной информационной системе института;
- выявлены недостатки существующей системы и на их основании предложена объектная модель «Как должно быть»;
- разработан программный интерфейс (API);
- оценена эффективность от внедрения API.

Таким образом, поставленная цель была достигнута в данной работе.

Был разработан программный интерфейс (API), который полностью удовлетворяет потребностям корпоративной информационной системы Рубцовского института и всем сторонним приложениям, интегрирующим ресурсы КИС.

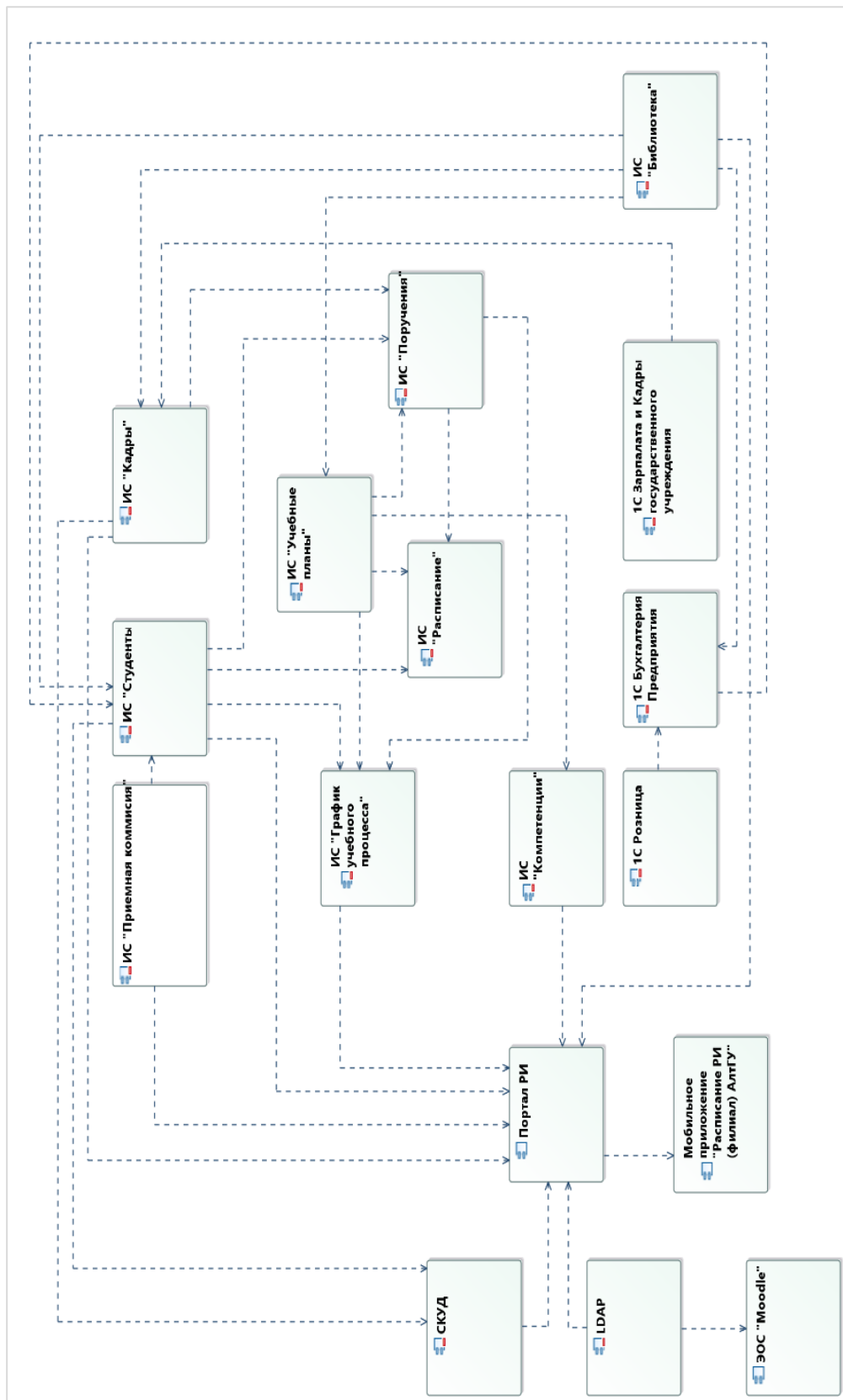
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Jonas Boner. Reactive Microservices Architecture / Jonas Boner. – O'Reilly Media, 2016. – 54 с.
2. Brian Mulloy. Announcement: Free eBook on Web API Design / Brian Mulloy. – O'Reilly Media, 2013. – 38 с.
3. Антонов, А.В. Системный анализ / А.В. Антонов. – М.: Высш. Шк., 2014. – 454 с.
4. Артемов, А. Информационная безопасность / А. Артемов. – М.: Академия безопасности и выживания, 2014. – 340 с.
5. Башмаков, А.И. Теория систем и системный анализ / А.И. Башмаков. – М.:Изд. дом МЭИ, 2013. – 52 с.
6. Васвани, В. Zend Framework: разработка веб-приложений на PHP / В. Васвани. – СПб.:Питер, 2014. – 432 с.
7. Гвоздева, Т.В. Проектирование информационных систем / Т.В. Гвоздева, Б.А. Баллод. – Ростов-н/Д:Феникс, 2013. – 508 с.
8. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – Москва:Форум, 2016. – 400 с.
9. Диго, С.М. Базы данных. Проектирование и создание / С.М. Диго. – М:ЕАОИ, 2013. – 172 с.
10. Кириллов, В.В. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб:БХВ – Петербург, 2015. – 464 с.
11. Мельников, В.П. Информационная безопасность и защита информации / В.П. Мельников, А.М. Петраков, С.А. Клейменов. – М.: АCADEMA, 2017. – 336 с.
12. Мишенин, А.И. Теория экономических информационных систем / А.И. Мишенин. – М.:Финансы и статистика, 2015. – 240 с.
13. Родичев, Ю.А. Информационная безопасность: Нормативно-правовые аспекты / Ю.А. Родичев. – СПб:Питер, 2013. – 272 с.
14. Шарп, Д. Подробное руководство по Microsoft Visual C# / Д. Шарп. – СПб.: Питер, 2017. – 848 с.

15. Architectural Styles and the Design of Network-based Software Architectures [Электронный ресурс]. – Режим доступа: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>. – Загл. с экрана.
16. Введение в OAuth 2 [Электронный ресурс]. – Режим доступа: <https://www.digitalocean.com/community/tutorials/oauth-2-ru>. – Загл. с экрана.
17. Использование Entity Framework Core code-first с СУБД SQLite [Электронный ресурс]. – Режим доступа: <https://habr.com/post/324272/>. – Загл. с экрана.
18. Сообщество PHP-Программистов, документация PHP, форум php-программистов [Электронный ресурс]. – Режим доступа: <http://www.php.ru/>. – Загл. с экрана.

ПРИЛОЖЕНИЕ А

Диаграммы корпоративной информационной системы Рубцовского
института (филиала) АлтГУ «Как есть»



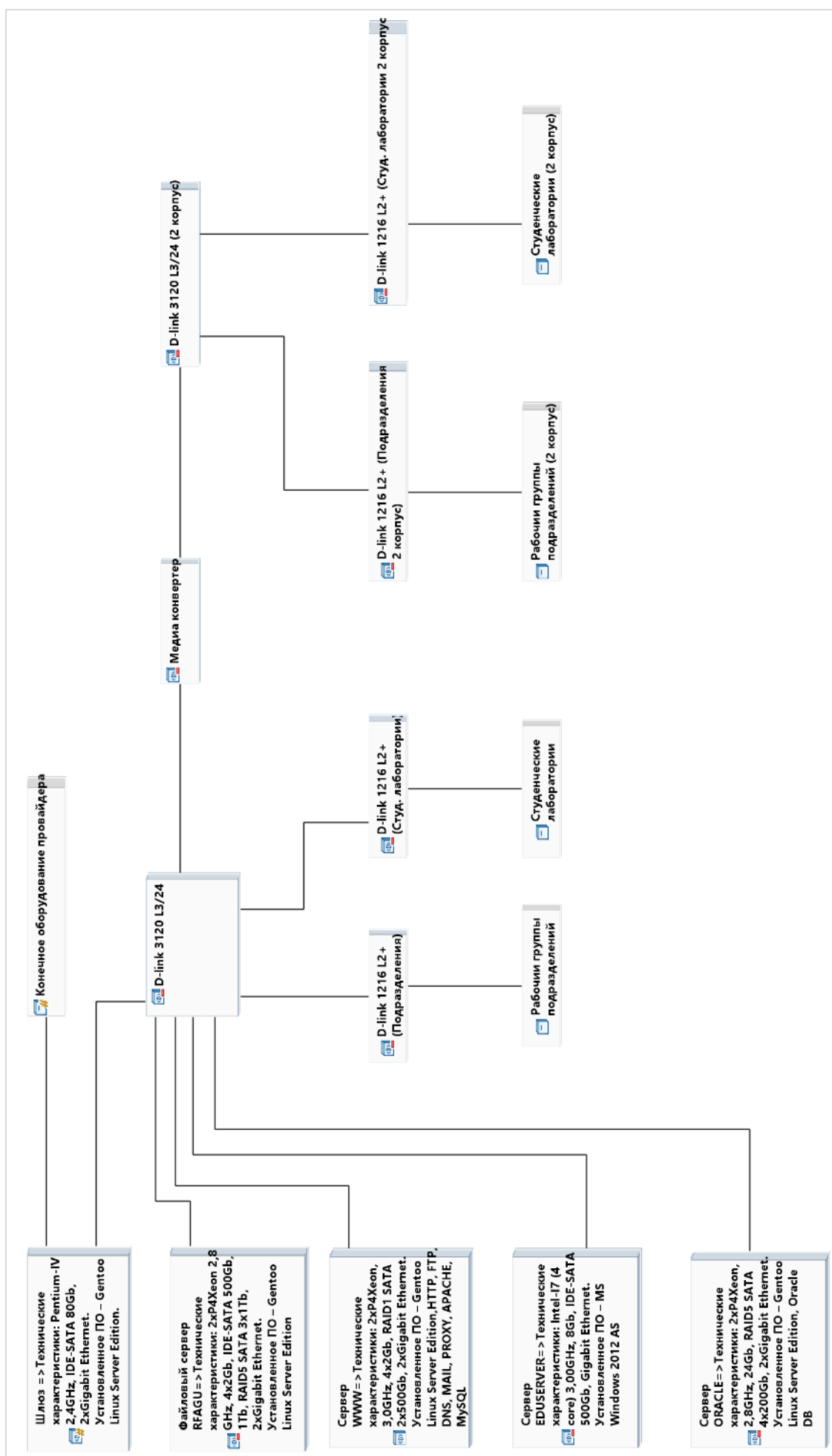


Рисунок А.2 – Диаграмма развертывания

ПРИЛОЖЕНИЕ Б

Диаграммы корпоративной информационной системы Рубцовского
института (филиала) АлтГУ «Как есть»

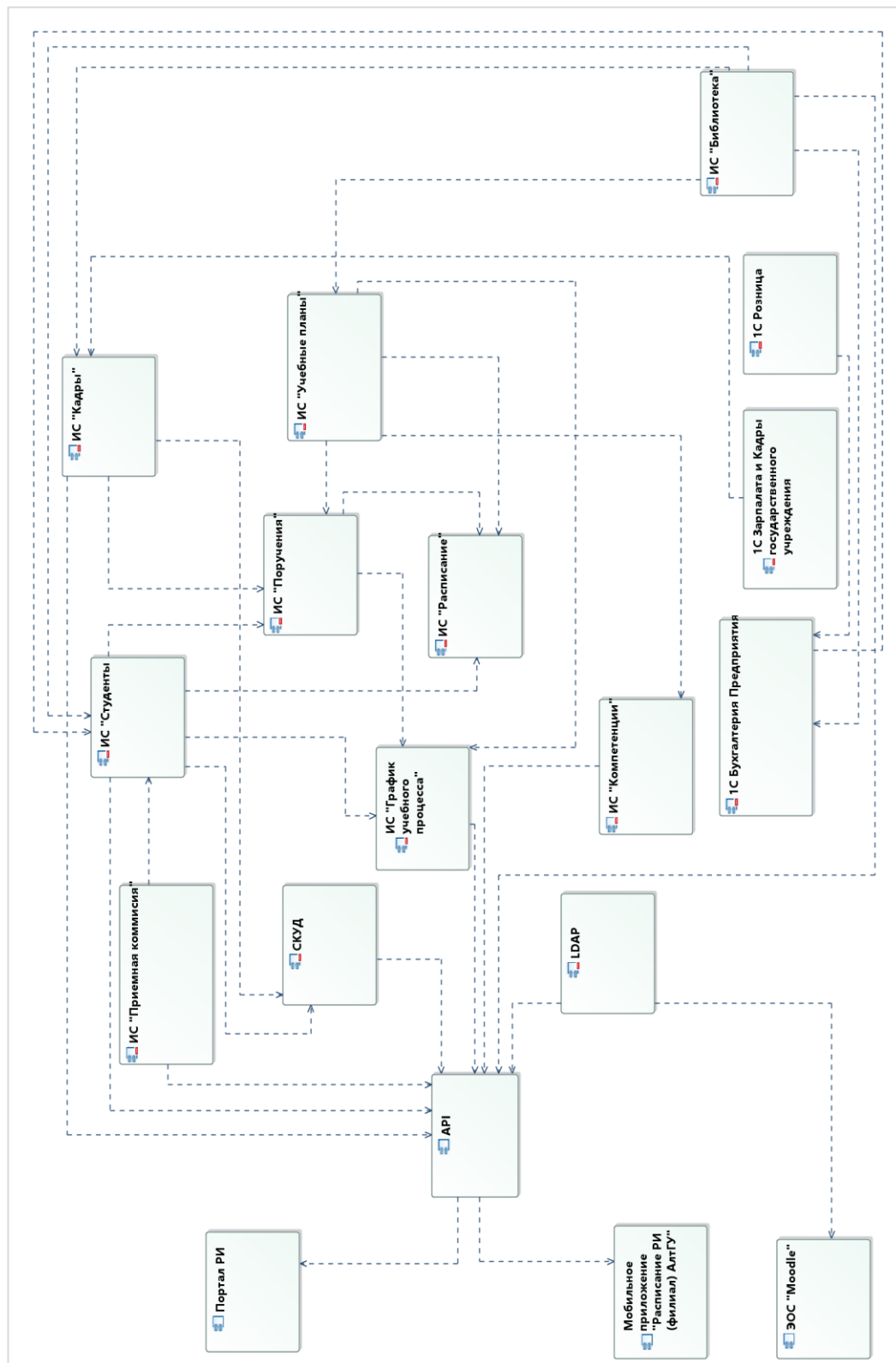
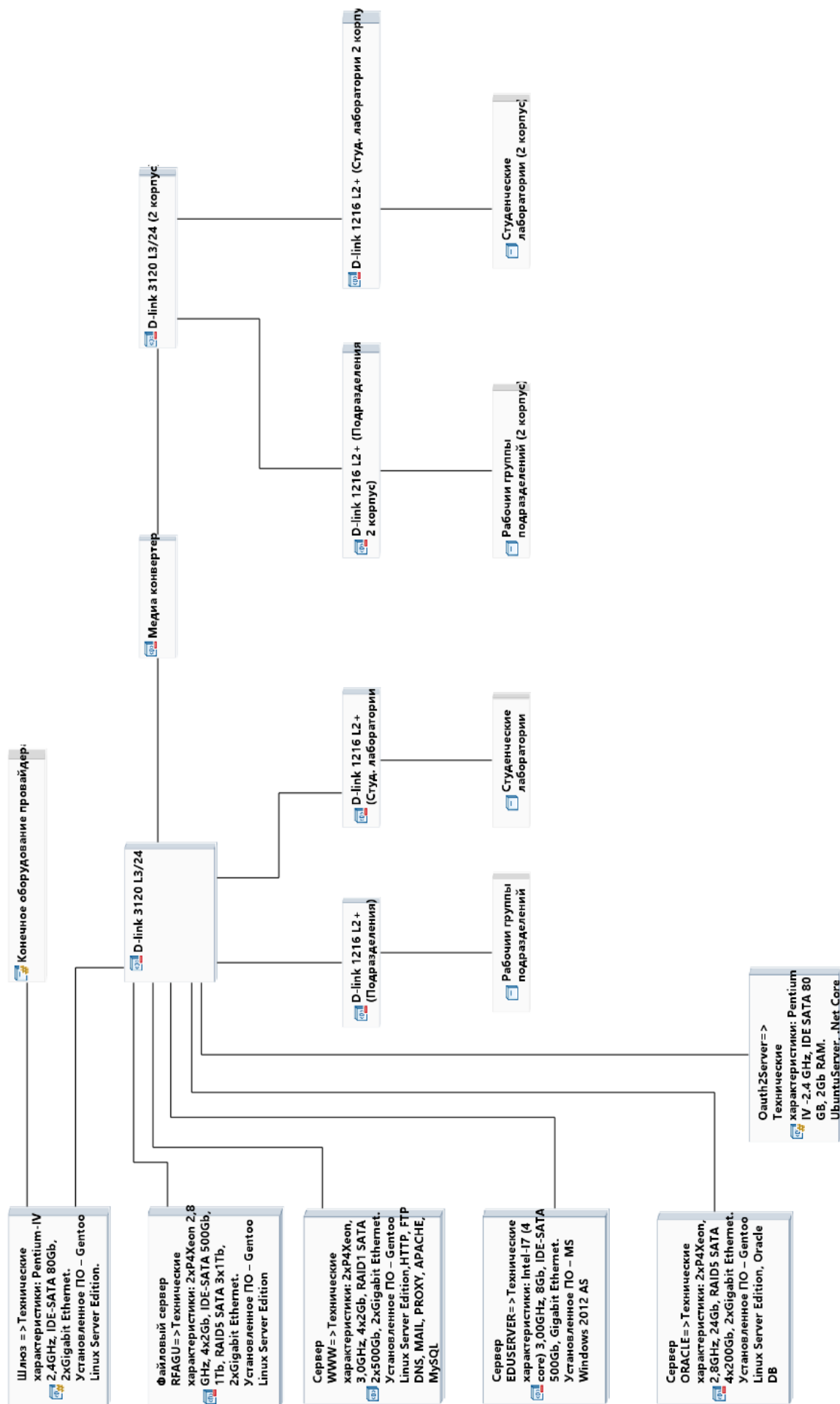


Рисунок Б.1 – Компонентная диаграмма



ПРИЛОЖЕНИЕ В

Физическая модель базы данных сервера OAuth 2.0

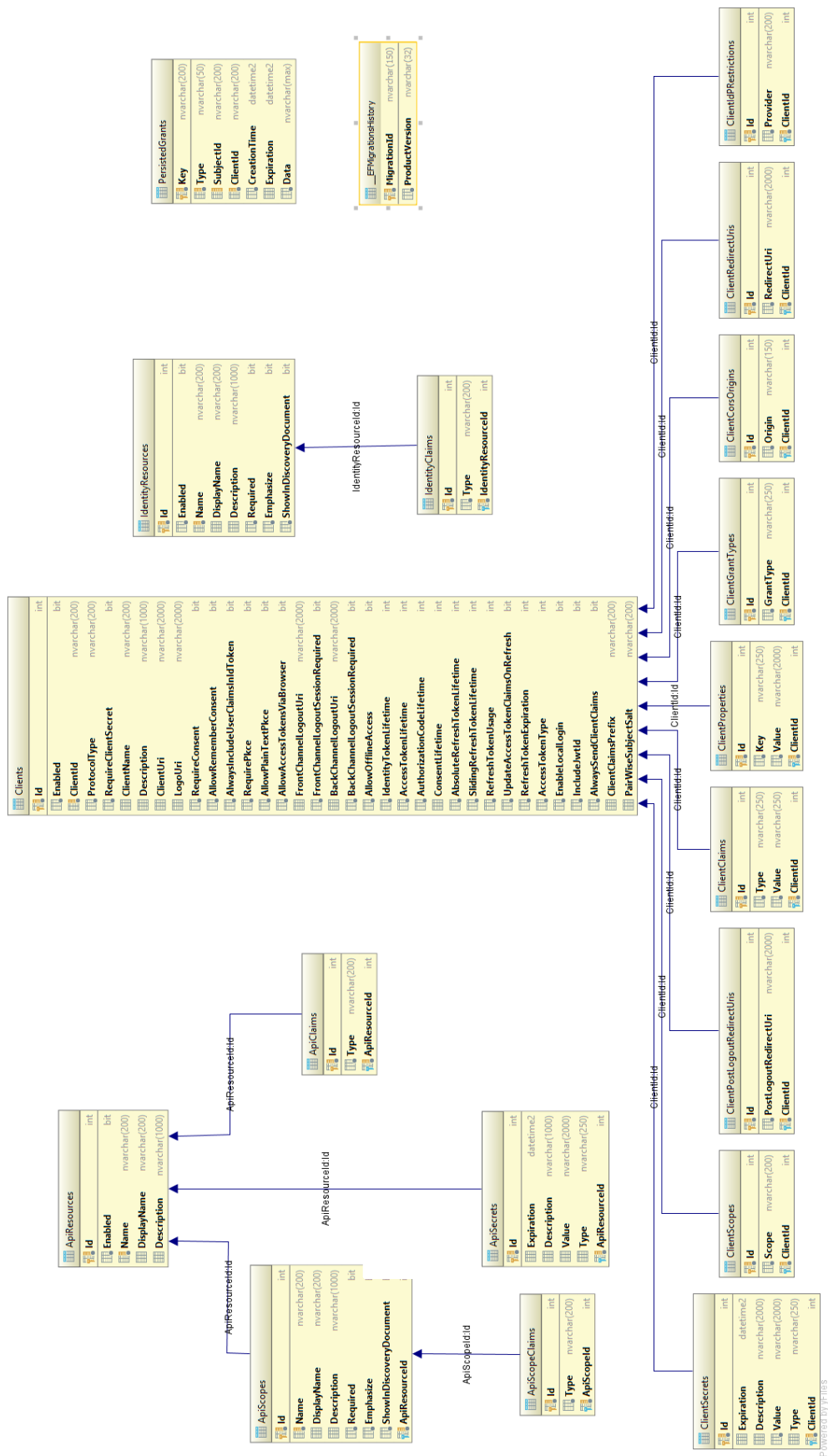


Рисунок В.1 – Физическая модель базы данных