

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)  
Кафедра автоматизированных систем управления (АСУ)

Утверждаю  
заведующий каф. АСУ  
А.М. Корилов

\_\_\_\_\_  
*Подпись*

«\_\_» \_\_\_\_\_ 2017

**Облачная информационная система обучения студентов**

Отчёт по групповому проектному обучению

Группа АСУ-1101

Научный руководитель:  
преподаватель каф. АСУ,  
профессор, д.т.н.  
М.Ю. Катаев

\_\_\_\_\_  
*Подпись*

«\_\_» \_\_\_\_\_ 2017

Участники проектной  
группы:

студент гр. 434-1  
Ю.А. Богомолов

\_\_\_\_\_  
*Подпись*

«\_\_» \_\_\_\_\_ 2017

студент гр. 434-1  
А.А. Бодрухин

\_\_\_\_\_  
*Подпись*

«\_\_» \_\_\_\_\_ 2017

## Оглавление

1	Введение .....	3
2	Описание концепта.....	5
2.1	Пользователь .....	5
2.2	Коллективы .....	6
2.3	Учебный процесс.....	7
2.4	Разработка учебного материала.....	8
2.5	Приложения .....	9
3	Описание проделанной работы.....	12
3.1	Использованные инструменты и технологии .....	12
3.2	Моделирование структуры базы данных.....	14
3.3	Приложение «Пользователи» .....	16
3.4	Приложение «Разработчики».....	18
4	Заключение.....	20
	Список использованных источников .....	21

## 1 Введение

Несмотря на все достижения научно-технического прогресса, в современном образовании, мало направлений являются автоматизированными. Проверка домашних заданий в школе, лабораторных работ в университете – всё это целиком ложится на плечи учителей и преподавателей. Одни и те же задания по математике, физике, химии – решаются многими поколениями школьников и студентов, что приводит к проблеме списывания. Большая часть упражнений до сих пор выполняется в тетрадях, хотя практически у каждого уже есть компьютер, а многие набирают тексты в разы быстрее, чем пишут текст ручкой. Эти и другие моменты являются основой для разработки и внедрения в практику автоматизированных систем в образовании.

Решением проблемы может стать система, которая сможет автоматизировать процессы проверки решений и генерации уникальных задач; позволит создавать учебный материал нового поколения, отличающийся новизной, интерактивностью и разнообразием; позволит учащимся выполнять задания на компьютерах и мгновенно получать отклик, что может их больше заинтересовать.

Предлагаемый концепт выражается следующими требованиями к системе:

- 1) должно присутствовать множество инструментов для разработки учебных материалов, а также поддерживаться их добавление;
- 2) преподаватели могут выдавать индивидуальные или коллективные задания;
- 3) учащиеся могут выполнять задания поодиночке или в группах;
- 4) перед тем как решения попадут к преподавателю, они пройдут автоматизированную проверку.

Предлагаемый концепт берёт своё начало в феврале 2017 года и всё это время он видоизменяется, расширяется, улучшается. Настоящая версия

концепта полностью отличается от той, которая была представлена весной 2017 года.

Целью, поставленной на этот семестр, было продолжить развивать идею, а также приступить к реализации системы.

В данном отчёте описывается концепт облачной информационной системы обучения студентов, которая позволит автоматизировать, упростить и улучшить многие аспекты обучения, и сделать его более разнообразным; приводится процесс разработки данной системы на языке Python 3 с использованием фреймворка для веб-разработки Django.

## **2 Описание концепта**

### **2.1 Пользователь**

Сердцем нашей системы является пользователь. Именно пользователи разрабатывают учебные материалы, решают задачи, создают средства для создания учебных материалов. Напрашивается разделение пользователей на следующие роли: автор, преподаватель, разработчик и студент. У каждой из ролей есть свои функции:

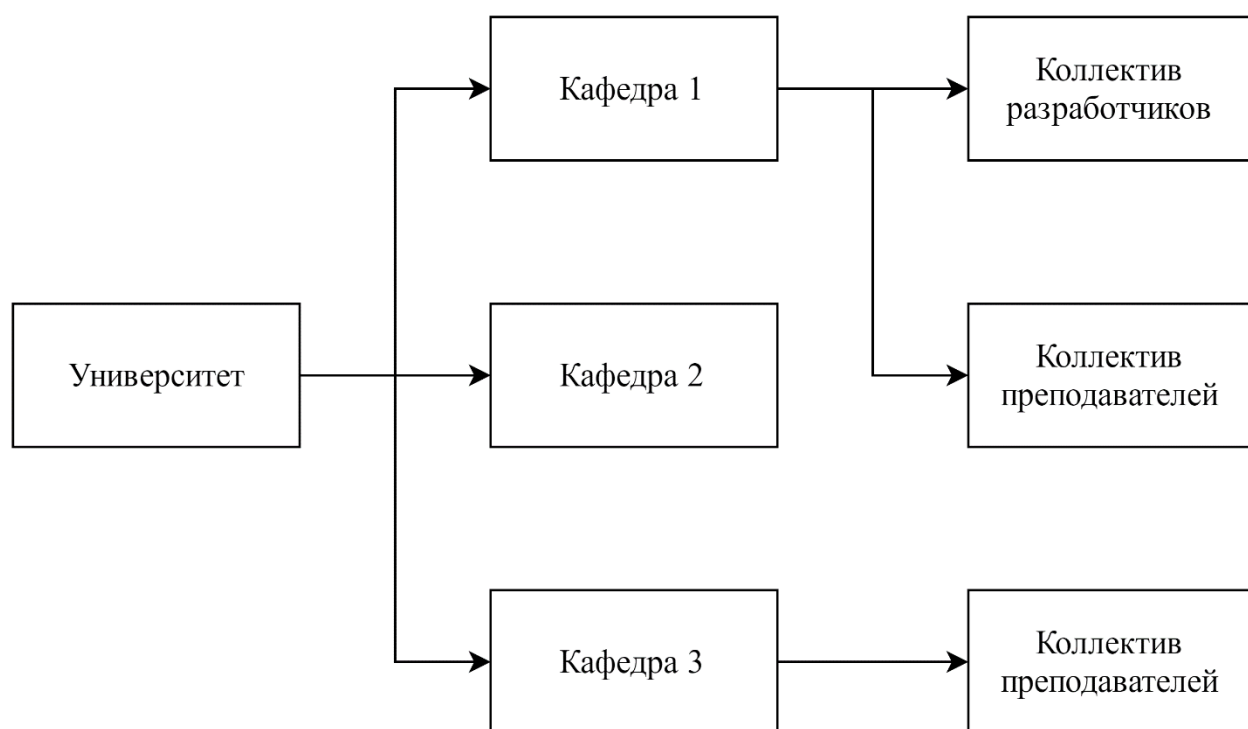
- 1) автор – пользователь, составляющий учебный материал;
- 2) преподаватель – пользователь, обучающий студентов по какому-либо учебному материалу; выдаёт задания учащимся, а также проверяет результаты их работы;
- 3) разработчик – пользователь, создающий программное обеспечение, дающее простор автору в составлении материала;
- 4) студент – пользователь, выполняющий задания, выданные преподавателем.

При этом один и тот же пользователь может иметь несколько ролей, например, аспирант, находящийся в процессе обучения, параллельно может быть ассистентом преподавателя, проводя лабораторные работы у студентов.

Следует отметить, что это не все роли пользователей в системе. Система позволяет производить логическое деление пользователей на коллективы, например, коллектив преподавателей кафедры АСУ ТУСУР, коллектив разработчиков ПО и коллектив тестировщиков.

## 2.2 Коллективы

Коллективы – средство логического деления пользователей на подразделения. Такие деления позволяют реализовывать учебные организации (например, университеты, школы, детские сады и пр.), делить их на подразделения (например, факультеты, кафедры, коллективы преподавателей и т.п.). Такие деления можно производить сколь угодно раз, пока не будет создана удовлетворяющая схема организации.



*Рисунок 2.1 Пример деления организации на коллективы*

Коллективы предоставляют функционал управления над дочерними коллективами: создание, редактирование, удаление.

Пользователь может находиться одновременно в разных коллективах и иметь разные роли.

### 2.3 Учебный процесс

Учебный процесс – организация учебно-воспитательной деятельности; направлен на достижение целей обучения. Включает все виды учебных занятий: уроки, лекции, проектные работы, контрольные работы, сочинения и другие.

Организация учебного процесса происходит следующим образом: формируется группа обучающихся, к ним прикрепляются коллективы преподавателей по каждому курсу, в рамках которых преподаватели выдают студентам материалы из этих курсов. Студенты же решают выданные им задания.

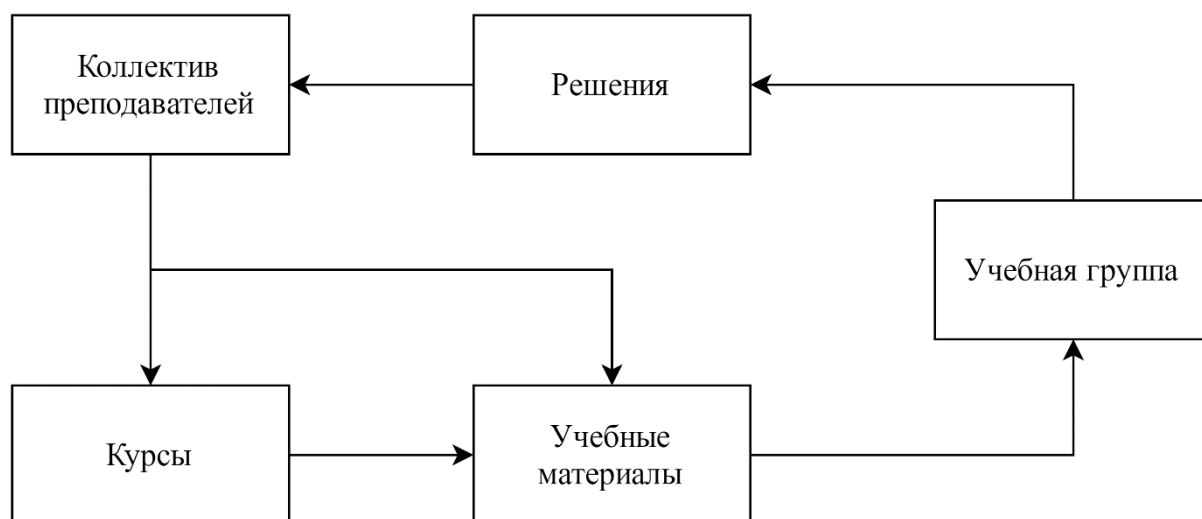


Рисунок 2.2 Схема учебного процесса

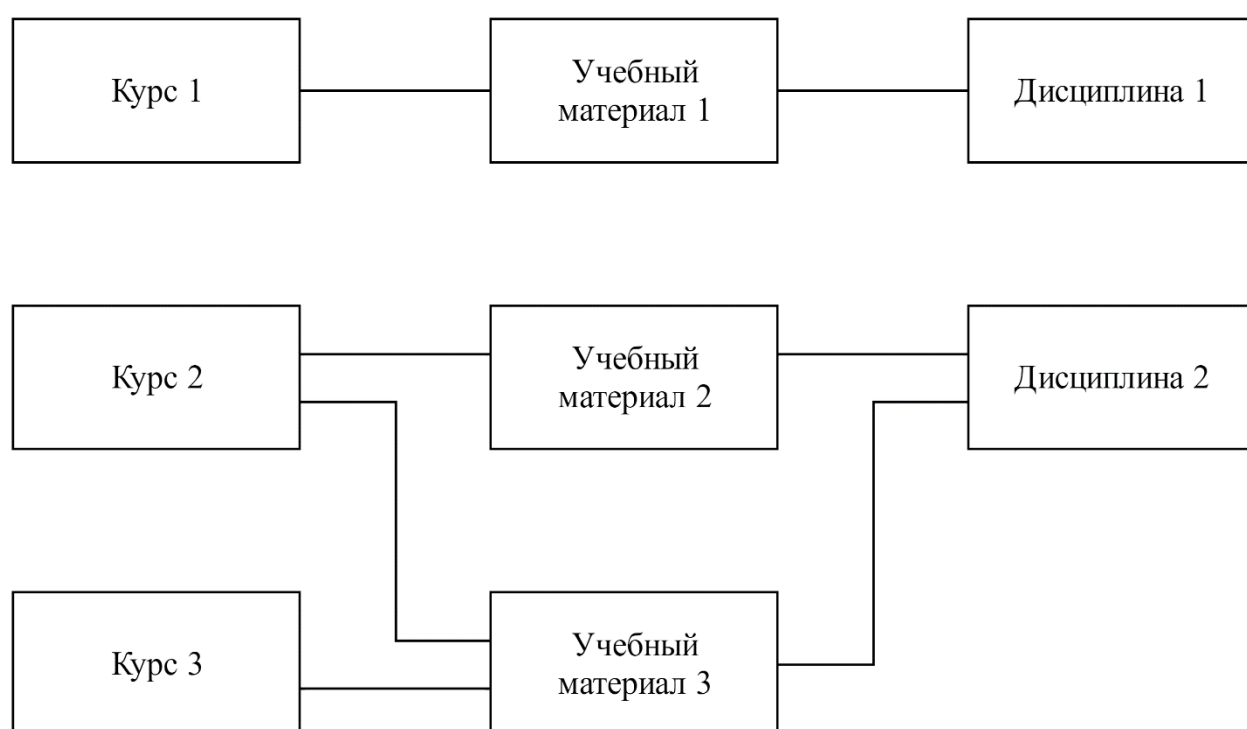
Учебный материал разрабатывается авторами и предоставляется преподавателям. Подробнее о разработке учебного материала в следующем разделе.

Непосредственный процесс обучения предоставляют приложения (раздел 2.5).

## 2.4 Разработка учебного материала

Разработка учебного материала заключается в составлении лекционного материала, прототипов задач, прототипов тестов и др. Разработка происходит с помощью специальных приложений (раздел 2.5), которые предоставляют средства для создания учебного материала.

Учебный материал относится к какой-либо дисциплине, и может быть частью какого-либо курса (курсов).



*Рисунок 2.3 Связь курсов и учебных материалов*

Каждый учебный материал имеет свои права доступа, таким образом можно предоставлять материал множеству преподавателей с сохранением авторства.



## 2.5 Приложения

Одной из важнейших частей системы являются приложения, т.к. они позволяют выполнять различные обработки данных и их отображения. Они могут и генерировать задачи по запросу преподавателя, и являться конструкторами учебного материала, и даже проверять решения студентов. Примеры использования приложений можно увидеть на рисунках 2.4-2.8.

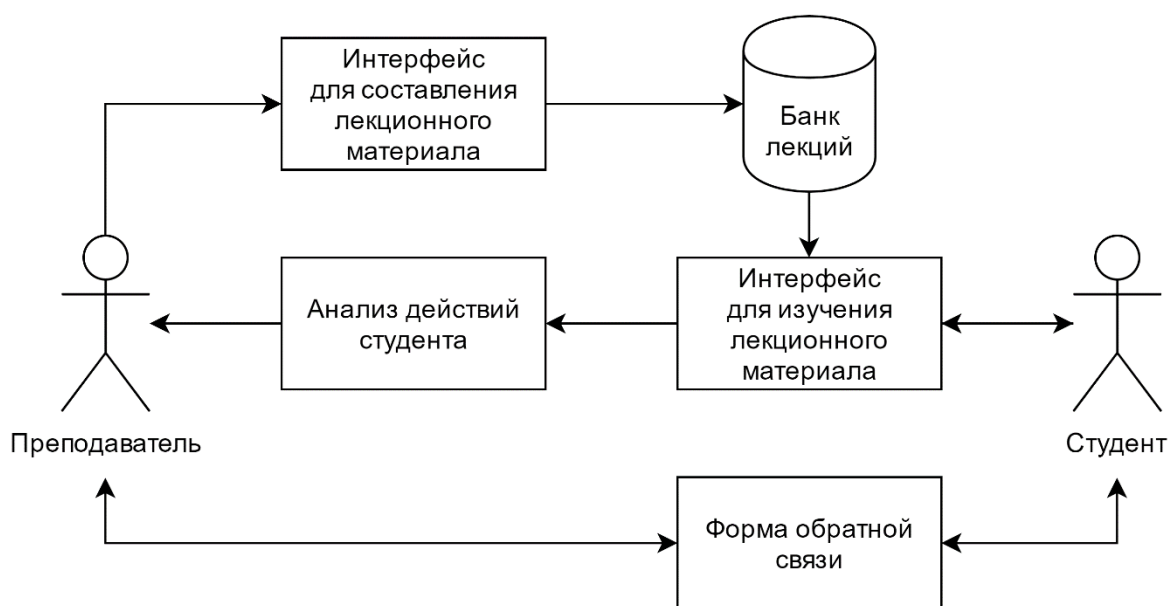
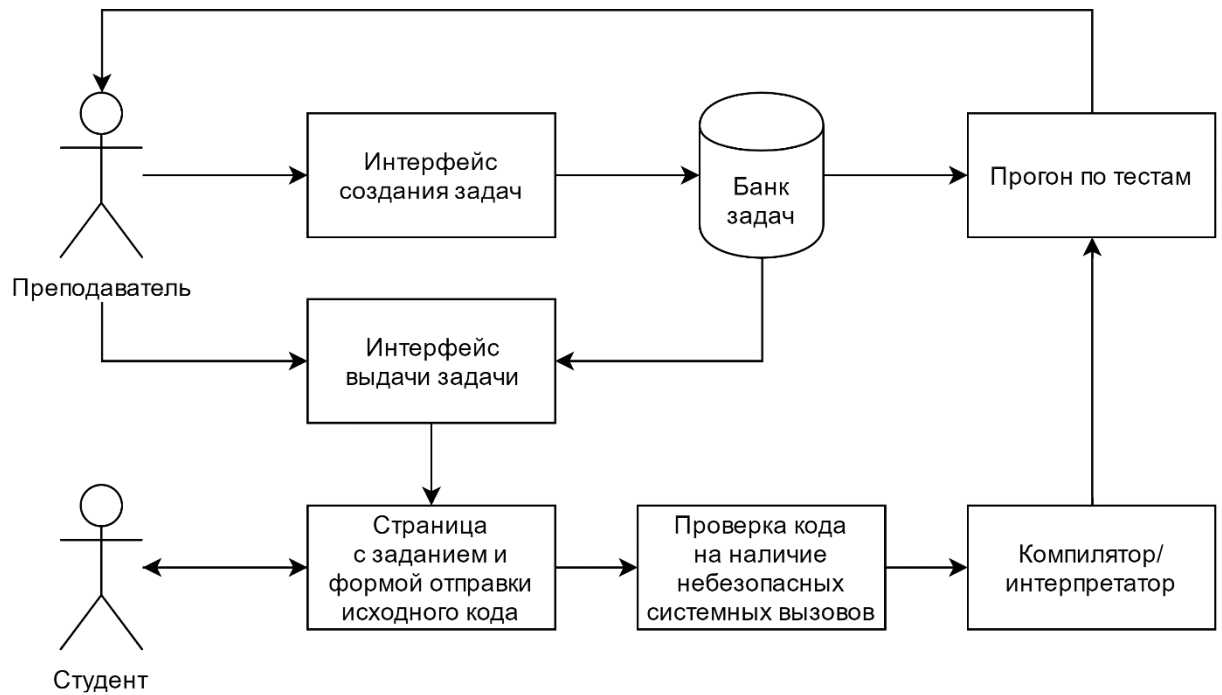


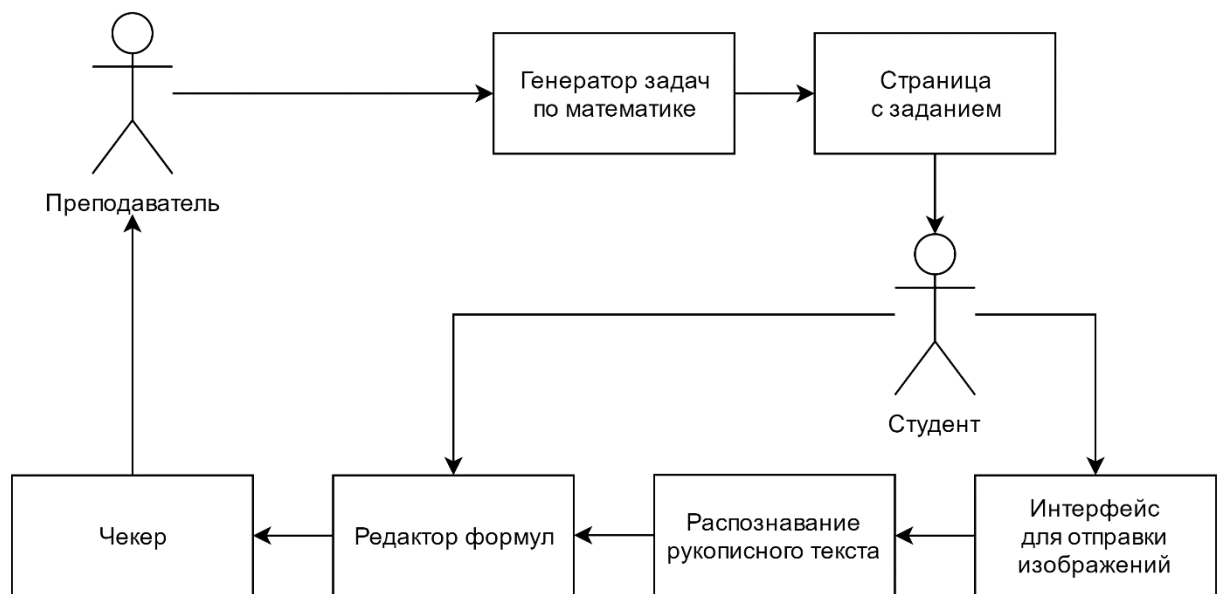
Рисунок 2.4 – Лекционный материал



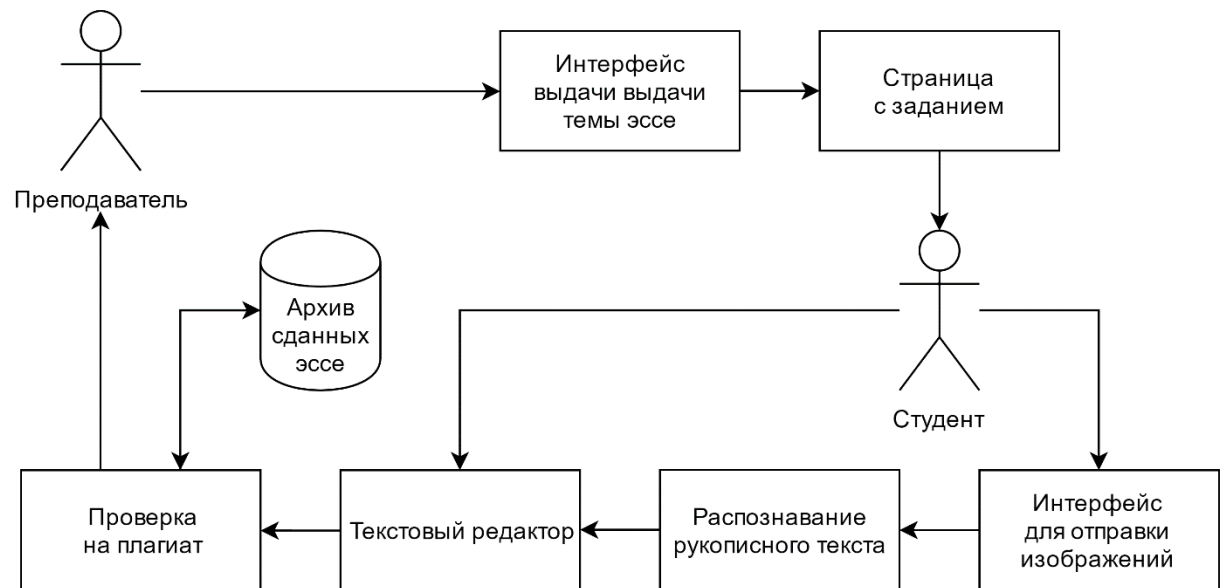
Рисунок 2.5 – Тест



*Рисунок 2.6 – Задача спортивного программирования*



*Рисунок 2.7 – Задачи по математике*



*Рисунок 2.8 – Эссе по литературе*

### 3 Описание проделанной работы

#### 3.1 Используемые инструменты и технологии

Как уже было сказано во введении, разработка системы ведётся на языке Python [1]. Это интерпретируемый высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули.

Для Python принята спецификация программного интерфейса к базам данных DB-API 2 и разработаны соответствующие этой спецификации пакеты для доступа к различным СУБД: Oracle, MySQL, PostgreSQL, Sybase, Firebird, Informix, Microsoft SQL Server и SQLite.

В нашей реализации используется PostgreSQL [2] – свободная объектно-реляционная система управления базами данных (СУБД). Сильными сторонами PostgreSQL считаются:

- высокопроизводительные и надёжные механизмы транзакций и репликации;
- наследование;
- легкая расширяемость.

Кроме того, проектирование базы данных выполнялось при помощи Pony ORM (Object-Relational Mapper) [3]. Это набор инструментов, таких как библиотека для Python и онлайн-построитель моделей баз данных. Мы

использовали только онлайн-построитель, так как инструмент, аналогичный первому, также предоставляется и Django.

Разработка веб-сайта осуществляется с помощью фреймворка Django [4][5].

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. Don't repeat yourself).

Веб-фреймворк Django используется в таких крупных и известных сайтах, как Instagram, Disqus, Mozilla, The Washington Times, Pinterest, YouTube, Google и др.

Предусмотрена локализация сайта для различных языков (на данный момент поддерживаются английский и русский языки). Однако, переводы учебных материалов должны будут предоставлять сами авторы.

### 3.2 Моделирование структуры базы данных

Модель базы данных, как уже упоминалось ранее, разрабатывалась при помощи онлайн-построителя Pony ORM.

Модель базы данных, как и концепт, претерпела значительные изменения. Модель уровня KB (Key Based) представлена на рисунке 3.1. Расшифровка легенды представлена в таблице 3.1.

Некоторые связи на рисунке 3.1 не приведены явно, так как являются динамическими (хранятся в одной таблице, но являются ссылками на строки разных таблиц). Такие связи нельзя отобразить при помощи редактора схем Pony ORM, но можно использовать в коде в проекте Django. Примеры таких связей: в таблице Access есть поля *source\_type* и *source\_id*, которые вместе образуют динамический внешний ключ (Generic Foreign Key). Они могут ссылаться на таблицы Course, Material, System, Application, Data Type. Это имеет следующий смысл: Коллектив может иметь различные права доступа к различным типам исходных данных: Курсам, Учебным Материалам, Системам, Приложениям и Типам Данных.

Таблица 3.1 Легенда к рисунку 3.1

Цветовая маркировка	Название приложения Django
<b>Красный</b>	Пользователи / Users
<b>Зелёный</b>	Разработчики / Developers
<b>Синий</b>	Учебный процесс / Education
<b>Жёлтый</b>	Учебные материалы / Materials
<b>Фиолетовый</b>	Приложения / Applications

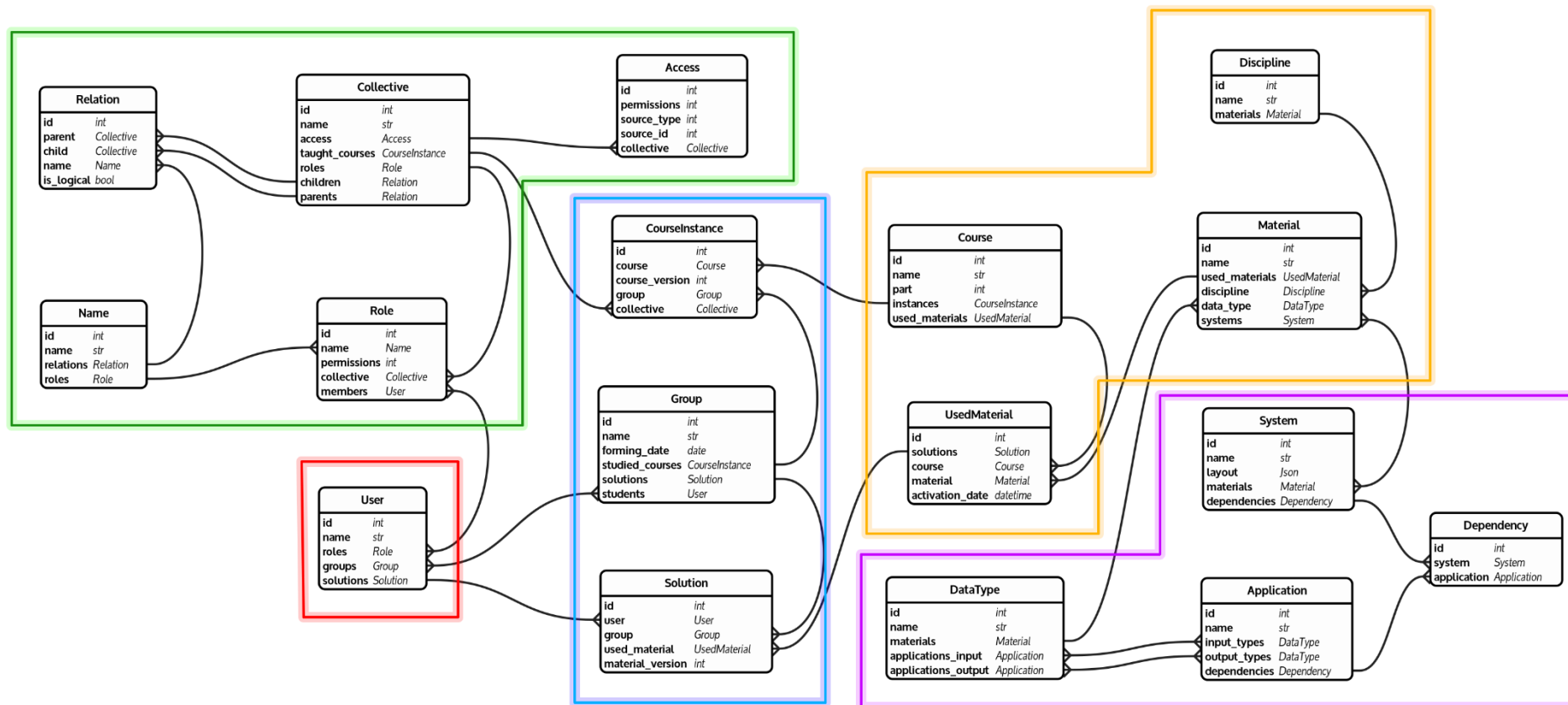


Рисунок 3.1 – Модель базы данных с разделением по приложениям Django

### 3.3 Приложение «Пользователи»

Было разработано приложение «Пользователи», которое выполняет следующие функции:

- регистрация пользователей;
- авторизация пользователей;
  - через соц. сети;
  - с использованием двухфакторной аутентификации;
- отображение страницы пользователя;
- настройки профиля пользователя;
- работа и электронной почтой и другие.

Поля сущности «Пользователь» базы данных представлены таблицей 3.2 (поля, помеченные звездочкой (\*) являются обязательными):

*Таблица 3.2 Поля сущности «Пользователь»*

Имя поля	Описание поля
pk*	ID пользователя, уникальный, первичный ключ
username*	имя пользователя, уникальный
password*	хэш пароля
first_name	имя
last_name	фамилия
email	электронная почта
gender	пол
job	место работы
birth_date	дата рождения
bio	биография и прочая информация;
country	страна
is_staff	возможность заходить в Django Admin
is_active	используется вместо удаления аккаунта
date_joined	дата регистрации
last_login	дата последнего логина

На рисунке 3.2 показана страница редактирования настроек пользователя.



## KerkDovan

Profile

E-Mail

[Change  
Password](#)

2FA

## Profile Settings

First name

Юрий

Last name

Богомолов

Gender

Male ▾

Birth Date

01/20/1997

Country

Россия ▾

Job

Tell something about your job. Max 150 symbols.

Additional information

Additional information. Max 1000 symbols.

Сохранить

Reset

*Рисунок 3.2 – Страница настроек пользователя*

### 3.4 Приложение «Разработчики»

На данный момент активно ведётся разработка приложения «Разработчики», которое должно выполнять следующие функции:

- создание коллективов;
- отображение страницы коллектива;
- редактирование информации коллектива;
- управление иерархической структурой коллективов;
- управление ролями пользователей в коллективе;
- управление разработкой различных материалов:
  - курсы;
  - учебные материалы;
  - приложения;
  - системы;
  - типы данных;
- управление обучением групп студентов.

На рисунке 3.3 показан пример того, как может выглядеть страница коллектива.

Parevia

About

KerkDovan

ASU-1101 ⚙

Members - 2

Invite

Projects

Create

AllApplicationsData typesSystemsCoursesMaterials

#3: Application/3:"Users" application

Owner

#4: Application/4:"Developers" application

Owner

#5: Application/5:"Education" application

Owner

#6: Application/6:"Materials" application

Owner

#7: Application/7:"Applications" application

Owner

#2: System/2:"Parevia" system

Owner

KerkDovan

Developer

A5trocat

Developer

Рисунок 3.3 – Страница коллектива

## 4 Заключение

Мы считаем, что нам удалось добиться значительного прогресса в разработке системы. Концепт системы значительно модернизировался; начата программная реализация проекта (и некоторые части уже завершены). На следующий семестр были поставлены такие планы:

- дальнейшее развитие концепции «облачной информационной системы обучения студентов»;
- реализация оставшихся приложений («разработчики», «учебный процесс», «разработка учебного материала» и «приложение»);
- разработать базовый набор приложений;
- разработать примеры учебных материалов.

**Список использованных источников**

1. Python [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Python>. (Дата обращения 18.12.2017)
2. PostgreSQL [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/PostgreSQL>. (Дата обращения 18.12.2017)
3. Pony ORM documentation [Электронный ресурс]. – Режим доступа: <https://docs.ponyorm.com>. (Дата обращения 24.12.2017)
4. Django Documentation [Электронный ресурс]. – Режим доступа: <https://docs.djangoproject.com/en/1.11>. (Дата обращения 24.12.2017)
5. Daniel Greenfeld. Two Scoops of Django 1.11: Best Practices for Django / Daniel Roy Greenfeld, Audrey Roy Greenfeld. – Четвёртое издание. – Two Scoops Press, 2017. – 515 страниц.