Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Специальность: «Программное обеспечение информационных технологий»

**ЛАБОРАТОРНАЯ РАБОТА №5, 6, 7, 8**

По курсу: «Объектно-ориентированные технологии программирования

и системное проектирование»

Студент-заочник 2 курса

Группы № 581072

ФИО: Богданова Кристина

Евгеньевна

Адрес: Польша, г. Реда,

ул. Млыньска 5Б-5

Тел. +48 576 335 295,

+375 25 929 99 11

Минск, 2017

# 5 ПОСТРОЕНИЕ ДИАГРАММ КОМПОНЕНТОВ (COMPONENT) И КЛАССОВ (CLASS)

## 5.1 Диаграмма компонентов

Диаграмма компонентов — статическая структурная диаграмма, 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.

На рисунке 1 изображена диаграмма компонентов.

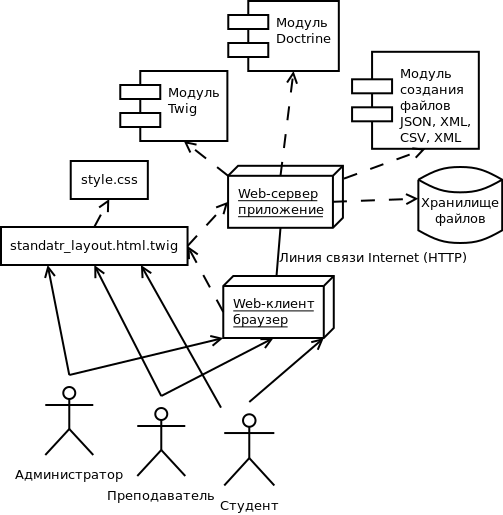


Рисунок 1 – Диаграмма компонентов

## 5.2 Диаграмма классов

Диаграмма классов— диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. Входит в UML.

Существует два вида:

* статический вид диаграммы рассматривает логические взаимосвязи классов между собой;
* аналитический вид диаграммы рассматривает общий вид и взаимосвязи классов, входящих в систему.

На рисунке 2 изображена диаграмма классов.

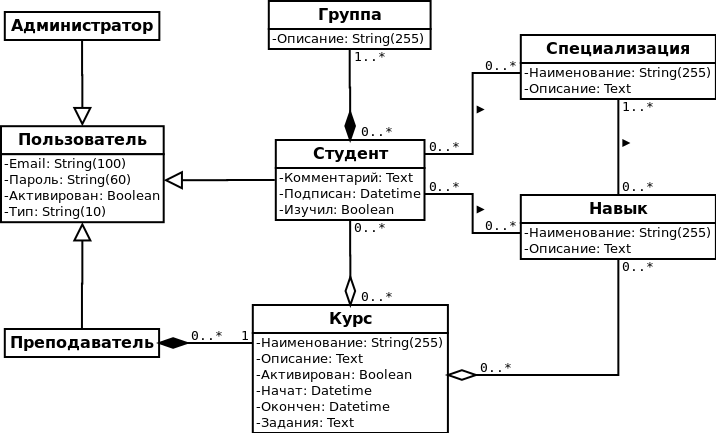


Рисунок 2 – Диаграмма классов

**6** **КОДОГЕНЕРАЦИЯ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СРЕДЕ АВТОМАТИЗИРОВАННОГО СИНТЕЗА**

На основе диаграммы классов был сформирован код, после добавления функциональности в него получены следующие работоспособные фрагменты кода:

*User.php*

**<?php**

**namespace** AppBundle\Entity;

**use** Doctrine\ORM\Mapping **as** ORM;

**use** Symfony\Component\Security\Core\User\UserInterface;

**use** Symfony\Component\Validator\Constraints **as** Assert;

**use** Symfony\Component\Serializer\Annotation\Groups;

**use** Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;

/\*\*

\* Users

\*

\* ApiResource(attributes={

\* "normalization\_context"={"groups"={"read"}}

\* }, itemOperations={

\* "get"={"method"="GET", "path"="/users/{id}"},

\* }, collectionOperations={

\* "get"={"method"="GET", "path"="/users"},

\* })

\*

\* @ORM\Table(name="users")

\* @ORM\Entity()

\* @UniqueEntity("email")

\*/

**class** **User** **implements** UserInterface

{

/\*\*

\* @const types

\*/

**const** TYPE\_ADMIN **=** 'admin';

**const** TYPE\_STUDENT **=** 'student';

**const** TYPE\_TEACHER **=** 'teacher';

/\*\*

\* @var int

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\* @Groups({"read"})

\*/

**private** $id;

/\*\*

\* @var string

\*

\* @ORM\Column(name="email", type="string", length=100, unique=true)

\* @Groups({"read"})

\*

\* @Assert\Email(

\* message = "entity.user.email",

\* checkMX = false

\* )

\*

\*/

**private** $email;

/\*\*

\* @var string

\*

\* @ORM\Column(name="password", type="string", length=60)

\*/

**private** $password;

/\*\*

\* @var bool

\*

\* @ORM\Column(name="enabled", type="boolean")

\* @Groups({"read"})

\*/

**private** $enabled;

/\*\*

\* @var string

\*

\* @ORM\Column(name="type", type="string", nullable = false, length=10)

\* @Assert\Choice(callback = "getAvailableTypes")

\* @Groups({"read"})

\*

\*/

**private** $type;

/\*\*

\* @var string

\*/

**private** $plainPassword;

/\*\*

\* @var Skill[]

\*

\* @ORM\ManyToMany(targetEntity="Skill", inversedBy="users")

\*/

**private** $skills;

/\*\*

\* @ORM\ManyToMany(targetEntity="Special", mappedBy="users")

\*/

**protected** $specials;

/\*\*

\* @return array

\*/

**public** **static** **function** **getAvailableTypes**()

{

**return** [self**::**TYPE\_ADMIN, self**::**TYPE\_STUDENT, self**::**TYPE\_TEACHER];

}

/\*\*

\* @return string

\*/

**public** **function** \_\_toString()

{

**return** $this**->**getEmail();

}

/\*\*

\* User constructor.

\*/

**public** **function** \_\_construct()

{

$this**->**enabled **=** **false**;

}

/\*\*

\* Get id

\*

\* @return integer

\*/

**public** **function** **getId**()

{

**return** $this**->**id;

}

/\*\*

\* Set email

\*

\* @param string $email

\*

\* @return User

\*/

**public** **function** **setEmail**($email)

{

$this**->**email **=** $email;

**return** $this;

}

/\*\*

\* Get email

\*

\* @return string

\*/

**public** **function** **getEmail**()

{

**return** $this**->**email;

}

/\*\*

\* Set password

\*

\* @param string $password

\*

\* @return User

\*/

**public** **function** **setPassword**($password)

{

$this**->**password **=** $password;

**return** $this;

}

/\*\*

\* Get password

\*

\* @return string

\*/

**public** **function** **getPassword**()

{

**return** $this**->**password;

}

/\*\*

\* Set enabled

\*

\* @param boolean $enabled

\*

\* @return User

\*/

**public** **function** **setEnabled**($enabled)

{

$this**->**enabled **=** $enabled;

**return** $this;

}

/\*\*

\* Get enabled

\*

\* @return boolean

\*/

**public** **function** **getEnabled**()

{

**return** $this**->**enabled;

}

/\*\*

\* Set type

\*

\* @param string $type

\*

\* @return User

\*/

**public** **function** **setType**($type)

{

$this**->**type **=** $type;

**return** $this;

}

/\*\*

\* Get type

\*

\* @return string

\*/

**public** **function** **getType**()

{

**return** $this**->**type;

}

/\*\*

\* Get skills

\*

\* @return \Doctrine\Common\Collections\Collection|array

\*/

**public** **function** **getSkills**()

{

**return** $this**->**skills;

}

/\*\*

\* @param Skill $skill

\* @return bool|mixed

\*/

**public** **function** **hasSkill**(Skill $skill)

{

**foreach** ($this**->**getSkills() **as** $item) {

**if** ($item**->**getId() **===** $skill**->**getId()) {

**return** $skill;

}

}

**return** **false**;

}

/\*\*

\* Add skill

\*

\* @param \AppBundle\Entity\Skill $skill

\* @return User

\*/

**public** **function** **addSkill**($skill)

{

**if** (**!**$this**->**hasSkill($skill)) {

$this**->**skills[] **=** $skill;

}

**return** $this;

}

/\*\*

\* {@inheritdoc}

\*/

**public** **function** **getRoles**()

{

**return** ['ROLE\_ADMIN'];

}

/\*\*

\* Returns the salt that was originally used to encode the password.

\*

\* This can return null if the password was not encoded using a salt.

\*

\* @return string|null The salt

\*/

**public** **function** **getSalt**()

{

**return** **null**;

}

/\*\*

\* Returns the username used to authenticate the user.

\*

\* @return string The username

\*/

**public** **function** **getUsername**()

{

**return** $this**->**email;

}

/\*\*

\* Removes sensitive data from the user.

\*

\* This is important if, at any given point, sensitive information like

\* the plain-text password is stored on this object.

\*/

**public** **function** **eraseCredentials**()

{

**return** **true**;

}

/\*\*

\* @return bool

\*/

**public** **function** **isTeacher**()

{

**return** $this**->**type **==** self**::**TYPE\_TEACHER;

}

/\*\*

\* @return bool

\*/

**public** **function** **isStudent**()

{

**return** $this**->**type **==** self**::**TYPE\_STUDENT;

}

/\*\*

\* @return bool

\*/

**public** **function** **isAdmin**()

{

**return** $this**->**type **==** self**::**TYPE\_ADMIN;

}

/\*\*

\* @param $plainPassword

\* @return $this

\*/

**public** **function** **setPlainPassword**($plainPassword)

{

$this**->**plainPassword **=** $plainPassword;

**return** $this;

}

/\*\*

\* @return string

\*/

**public** **function** **getPlainPassword**()

{

**return** $this**->**plainPassword;

}

}

*Student.php*

**<?php**

**namespace** AppBundle\Entity;

**use** Doctrine\ORM\Mapping **as** ORM;

/\*\*

\* Student

\*

\* @ORM\Table(name="student")

\* @ORM\Entity(repositoryClass="AppBundle\Repository\StudentRepository")

\*/

**class** **Student**

{

/\*\*

\* @var int

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

**private** $id;

/\*\*

\* @var User

\*

\* @ORM\ManyToOne(targetEntity="User")

\*/

**private** $user;

/\*\*

\* @ORM\ManyToOne(targetEntity="Course", inversedBy="students")

\* @var Course

\*/

**private** $course;

/\*\*

\* @ORM\Column(type="datetime", name="subscribed\_at")

\* @var \DateTime

\*/

**private** $subscribedAt;

/\*\*

\* @var bool

\*

\* @ORM\Column(name="studied", type="boolean", nullable=true)

\*/

**private** $studied;

/\*\*

\* @var string

\*

\* @ORM\Column(name="comment", type="text", nullable=true)

\*/

**private** $comment;

/\*\*

\* @ORM\ManyToOne(targetEntity="Group", inversedBy="students")

\* @var Course

\*/

**private** $group;

/\*\*

\* Student constructor.

\*/

**public** **function** \_\_construct(User $user)

{

$this**->**user **=** $user;

$this**->**subscribedAt **=** **new** \DateTime();

}

/\*\*

\* @return string

\*/

**public** **function** \_\_toString()

{

**return** (string) $this**->**getUser()**->**getEmail();

}

/\*\*

\* Get id

\*

\* @return int

\*/

**public** **function** **getId**()

{

**return** $this**->**id;

}

/\*\*

\* Set subscribedAt

\*

\* @param \DateTime $subscribedAt

\*

\* @return Student

\*/

**public** **function** **setSubscribedAt**($subscribedAt)

{

$this**->**subscribedAt **=** $subscribedAt;

**return** $this;

}

/\*\*

\* Get subscribedAt

\*

\* @return \DateTime

\*/

**public** **function** **getSubscribedAt**()

{

**return** $this**->**subscribedAt;

}

/\*\*

\* Set studied

\*

\* @param boolean $studied

\*

\* @return Student

\*/

**public** **function** **setStudied**($studied)

{

$this**->**studied **=** $studied;

**if** ($studied) {

$skills **=** $this**->**course**->**getSkills()**->**toArray();

**foreach** ($skills **as** $skill) {

$this**->**user**->**addSkill($skill);

}

}

**return** $this;

}

/\*\*

\* Get studied

\*

\* @return boolean

\*/

**public** **function** **getStudied**()

{

**return** $this**->**studied;

}

/\*\*

\* Set comment

\*

\* @param string $comment

\*

\* @return Student

\*/

**public** **function** **setComment**($comment)

{

$this**->**comment **=** $comment;

**return** $this;

}

/\*\*

\* Get comment

\*

\* @return string

\*/

**public** **function** **getComment**()

{

**return** $this**->**comment;

}

/\*\*

\* Set user

\*

\* @param \AppBundle\Entity\User $user

\*

\* @return Student

\*/

**public** **function** **setUser**(\AppBundle\Entity\User $user **=** **null**)

{

$this**->**user **=** $user;

**return** $this;

}

/\*\*

\* Get user

\*

\* @return \AppBundle\Entity\User

\*/

**public** **function** **getUser**()

{

**return** $this**->**user;

}

/\*\*

\* Set course

\*

\* @param \AppBundle\Entity\Course $course

\*

\* @return Student

\*/

**public** **function** **setCourse**(\AppBundle\Entity\Course $course **=** **null**)

{

$this**->**course **=** $course;

**return** $this;

}

/\*\*

\* Get course

\*

\* @return \AppBundle\Entity\Course

\*/

**public** **function** **getCourse**()

{

**return** $this**->**course;

}

}

*Teacher.php*

**<?php**

**namespace** AppBundle\Entity;

**use** Doctrine\ORM\Mapping **as** ORM;

/\*\*

\* Teacher

\*

\* @ORM\Table(name="teachers")

\* @ORM\Entity()

\*/

**class** **Teacher**

{

/\*\*

\* @var int

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

**private** $id;

/\*\*

\* @var string

\*

\* @ORM\OneToOne(targetEntity="User")

\*/

**private** $user;

/\*\*

\* @ORM\OneToMany(targetEntity="Course", mappedBy="teacher")

\* @var Course[]

\*/

**private** $courses;

/\*\*

\* Constructor

\*/

**public** **function** \_\_construct(User $user)

{

$this**->**user **=** $user;

$this**->**courses **=** **new** \Doctrine\Common\Collections\ArrayCollection();

}

/\*\*

\* @return string

\*/

**public** **function** \_\_toString()

{

**return** $this**->**getUser()**->**getEmail();

}

/\*\*

\* Get id

\*

\* @return integer

\*/

**public** **function** **getId**()

{

**return** $this**->**id;

}

/\*\*

\* Set user

\*

\* @param \AppBundle\Entity\User $user

\*

\* @return Teacher

\*/

**public** **function** **setUser**(\AppBundle\Entity\User $user **=** **null**)

{

$this**->**user **=** $user;

**return** $this;

}

/\*\*

\* Get user

\*

\* @return \AppBundle\Entity\User

\*/

**public** **function** **getUser**()

{

**return** $this**->**user;

}

/\*\*

\* Add course

\*

\* @param \AppBundle\Entity\Course $course

\*

\* @return Teacher

\*/

**public** **function** **addCourse**(\AppBundle\Entity\Course $course)

{

$this**->**courses[] **=** $course;

**return** $this;

}

/\*\*

\* Remove course

\*

\* @param \AppBundle\Entity\Course $course

\*/

**public** **function** **removeCourse**(\AppBundle\Entity\Course $course)

{

$this**->**courses**->**removeElement($course);

}

/\*\*

\* Get courses

\*

\* @return \Doctrine\Common\Collections\Collection

\*/

**public** **function** **getCourses**()

{

**return** $this**->**courses;

}

}

*Group.php*

**<?php**

**namespace** AppBundle\Entity;

**use** Doctrine\ORM\Mapping **as** ORM;

/\*\*

\* Class

\*

\* @ORM\Table(name="group")

\* @ORM\Entity

\*/

**class** **Group**

{

/\*\*

\* @var int

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

**private** $id;

/\*\*

\* @var string

\*

\* @ORM\Column(name="direction", type="string", length=255)

\*/

**private** $direction;

/\*\*

\* @ORM\OneToMany(targetEntity="Student", mappedBy="group", cascade={"persist","remove"})

\* @var students[]

\*/

**private** $students;

/\*\*

\* Get id

\*

\* @return int

\*/

**public** **function** **getId**()

{

**return** $this**->**id;

}

/\*\*

\* Set direction

\*

\* @param string $direction

\*

\* @return Class

\*/

**public** **function** **setDirection**($direction)

{

$this**->**direction **=** $direction;

**return** $this;

}

/\*\*

\* Get direction

\*

\* @return string

\*/

**public** **function** **getDirection**()

{

**return** $this**->**direction;

}

/\*\*

\* Set students

\*

\* @param integer $students

\*

\* @return Class

\*/

**public** **function** **setStudents**($students)

{

$this**->**students **=** $students;

**return** $this;

}

/\*\*

\* Get students

\*

\* @return int

\*/

**public** **function** **getStudents**()

{

**return** $this**->**students;

}

}

*Course.php*

**<?php**

**namespace** AppBundle\Entity;

**use** Doctrine\ORM\Mapping **as** ORM;

**use** Doctrine\Common\Collections\ArrayCollection;

**use** Symfony\Component\Validator\Constraints **as** Assert;

/\*\*

\* Course

\*

\* @ORM\Table(name="courses")

\* @ORM\Entity(repositoryClass="AppBundle\Repository\CourseRepository")

\*/

**class** **Course**

{

/\*\*

\* @var int

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

**private** $id;

/\*\*

\* @var Teacher

\*

\* @ORM\ManyToOne(targetEntity="Teacher", inversedBy="courses")

\*/

**private** $teacher;

/\*\*

\* @Assert\Count(

\* min = 1,

\* minMessage = "You must specify at least one email"

\* )

\* @ORM\ManyToMany(targetEntity="Skill", inversedBy="courses")

\*

\* @var Skill[]

\*/

**private** $skills;

/\*\*

\* @var Student[]

\*

\* @ORM\OneToMany(targetEntity="Student", mappedBy="course", cascade={"persist","remove"})

\*/

**private** $students;

/\*\*

\* @var string

\*

\* @ORM\Column(name="title", type="string", length=255)

\*/

**private** $title;

/\*\*

\* @var string

\*

\* @ORM\Column(name="description", type="text", nullable=true)

\*/

**private** $description;

/\*\*

\* @var string

\*

\* @ORM\Column(name="tasks", type="text", nullable=true)

\*/

**private** $tasks;

/\*\*

\* @var boolean

\*

\* @ORM\Column(name="enabled", type="boolean", nullable=true)

\*/

**private** $enabled;

/\*\*

\* @ORM\Column(type="datetime", name="start\_at", nullable=true)

\* @Assert\DateTime()

\*

\* @Assert\GreaterThan("today")

\*

\* @Assert\Expression(

\* "(this.getStartAt() > this.getNow() && !this.isStarted()) || (this.getStartAt() <= this.getNow() && this.isStarted())",

\* message="Start should be greater than today if course not started"

\* )

\*

\* @var \DateTime

\*/

**private** $startAt;

/\*\*

\* @ORM\Column(type="datetime", name="end\_at", nullable=true)

\*

\* @Assert\DateTime()

\*

\* @Assert\Expression(

\* "this.getStartAt() < this.getEndAt()",

\* message="The end at must be after the start at"

\* )

\*

\* @var \DateTime

\*/

**private** $endAt;

/\*\*

\* Course constructor.

\* @param Teacher $teacher

\*/

**public** **function** \_\_construct(Teacher $teacher)

{

$this**->**teacher **=** $teacher;

$this**->**enabled **=** **false**;

$this**->**skills **=** **new** ArrayCollection();

}

/\*\*

\* @return string

\*/

**public** **function** \_\_toString()

{

**return** $this**->**title **?:** '';

}

/\*\*

\* @return \DateTime

\*/

**public** **function** **getNow**()

{

**return** **new** \DateTime();

}

/\*\*

\* Get id

\*

\* @return integer

\*/

**public** **function** **getId**()

{

**return** $this**->**id;

}

/\*\*

\* Set title

\*

\* @param string $title

\*

\* @return Course

\*/

**public** **function** **setTitle**($title)

{

$this**->**title **=** $title;

**return** $this;

}

/\*\*

\* Get title

\*

\* @return string

\*/

**public** **function** **getTitle**()

{

**return** $this**->**title;

}

/\*\*

\* Set description

\*

\* @param string $description

\*

\* @return Course

\*/

**public** **function** **setDescription**($description)

{

$this**->**description **=** $description;

**return** $this;

}

/\*\*

\* Get description

\*

\* @return string

\*/

**public** **function** **getDescription**()

{

**return** $this**->**description;

}

/\*\*

\* Set teacher

\*

\* @param \AppBundle\Entity\Teacher $teacher

\*

\* @return Course

\*/

**public** **function** **setTeacher**(\AppBundle\Entity\Teacher $teacher **=** **null**)

{

$this**->**teacher **=** $teacher;

**return** $this;

}

/\*\*

\* Get teacher

\*

\* @return \AppBundle\Entity\Teacher

\*/

**public** **function** **getTeacher**()

{

**return** $this**->**teacher;

}

/\*\*

\* @return Student|false

\*/

**public** **function** **hasStudent**(User $user)

{

**foreach** ($this**->**getStudents() **as** $student) {

**if** ($student**->**getUser()**->**getId() **==** $user**->**getId()) {

**return** $student;

}

}

**return** **false**;

}

/\*\*

\* Add student

\*

\* @param \AppBundle\Entity\Student $student

\*

\* @return Course

\*/

**public** **function** **addStudent**(\AppBundle\Entity\Student $student)

{

$student**->**setCourse($this);

$this**->**students[] **=** $student;

**return** $this;

}

/\*\*

\* Remove student

\*

\* @param \AppBundle\Entity\Student $student

\*/

**public** **function** **removeStudent**(\AppBundle\Entity\Student $student)

{

$student**->**setCourse(**null**);

$this**->**students**->**removeElement($student);

}

/\*\*

\* Get students

\*

\* @return \Doctrine\Common\Collections\Collection|array

\*/

**public** **function** **getStudents**()

{

**return** $this**->**students **?:** [];

}

/\*\*

\* Set enabled

\*

\* @param boolean $enabled

\*

\* @return Course

\*/

**public** **function** **setEnabled**($enabled)

{

$this**->**enabled **=** $enabled;

**return** $this;

}

/\*\*

\* Get enabled

\*

\* @return boolean

\*/

**public** **function** **getEnabled**()

{

**return** $this**->**enabled;

}

/\*\*

\* Set startAt

\*

\* @param \DateTime $startAt

\*

\* @return Course

\*/

**public** **function** **setStartAt**($startAt)

{

$this**->**startAt **=** $startAt;

**return** $this;

}

/\*\*

\* Get startAt

\*

\* @return \DateTime

\*/

**public** **function** **getStartAt**()

{

**return** $this**->**startAt;

}

/\*\*

\* Set endAt

\*

\* @param \DateTime $endAt

\*

\* @return Course

\*/

**public** **function** **setEndAt**($endAt)

{

$this**->**endAt **=** $endAt;

**return** $this;

}

/\*\*

\* Get endAt

\*

\* @return \DateTime

\*/

**public** **function** **getEndAt**()

{

**return** $this**->**endAt;

}

/\*\*

\* @return boolean

\*/

**public** **function** **isStarted**()

{

$now **=** **new** \DateTime();

**return** $this**->**enabled **&&** $this**->**getStartAt() **<=** $now **&&** $this**->**getEndAt() **>=** $now;

}

/\*\*

\* @return boolean

\*/

**public** **function** **isEnded**()

{

$now **=** **new** \DateTime();

**return** $this**->**enabled **&&** $this**->**getEndAt() **<** $now;

}

/\*\*

\* Get skills

\*

\* @return \Doctrine\Common\Collections\Collection|array

\*/

**public** **function** **getSkills**()

{

**return** $this**->**skills;

}

/\*\*

\* Set skills

\*

\* @param \AppBundle\Entity\Skill $skills

\* @return Course

\*/

**public** **function** **addSkills**($skills)

{

$this**->**skills[] **=** $skills;

**return** $this;

}

/\*\*

\* Set tasks

\*

\* @param string $tasks

\*

\* @return Course

\*/

**public** **function** **setTasks**($tasks)

{

$this**->**tasks **=** $tasks;

**return** $this;

}

/\*\*

\* Get tasks

\*

\* @return string

\*/

**public** **function** **getTasks**()

{

**return** $this**->**tasks;

}

}

*Skill.php*

**<?php**

**namespace** AppBundle\Entity;

**use** Doctrine\ORM\Mapping **as** ORM;

/\*\*

\* Skill

\*

\* @ORM\Table(name="skills")

\* @ORM\Entity(repositoryClass="AppBundle\Repository\SkillRepository")

\*/

**class** **Skill**

{

/\*\*

\* @var int

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

**private** $id;

/\*\*

\* @var string

\*

\* @ORM\Column(name="title", type="string", length=255)

\*/

**private** $title;

/\*\*

\* @var string

\*

\* @ORM\Column(name="description", type="text", nullable=true)

\*/

**private** $description;

/\*\*

\* @var User[]

\*

\* @ORM\ManyToMany(targetEntity="User", mappedBy="skills")

\*/

**private** $users;

/\*\*

\* @var Course[]

\*

\* @ORM\ManyToMany(targetEntity="Course", mappedBy="skills")

\*/

**private** $courses;

/\*\*

\* @ORM\ManyToMany(targetEntity="Special", mappedBy="skills")

\*/

**protected** $specials;

/\*\*

\* Skill constructor.

\*/

**public** **function** \_\_construct()

{

}

/\*\*

\* @return string

\*/

**public** **function** \_\_toString()

{

**return** $this**->**title **?:** '';

}

/\*\*

\* Get id

\*

\* @return integer

\*/

**public** **function** **getId**()

{

**return** $this**->**id;

}

/\*\*

\* Set title

\*

\* @param string $title

\*

\* @return Skill

\*/

**public** **function** **setTitle**($title)

{

$this**->**title **=** $title;

**return** $this;

}

/\*\*

\* Get title

\*

\* @return string

\*/

**public** **function** **getTitle**()

{

**return** $this**->**title;

}

/\*\*

\* Set description

\*

\* @param string $description

\*

\* @return Skill

\*/

**public** **function** **setDescription**($description)

{

$this**->**description **=** $description;

**return** $this;

}

/\*\*

\* Get description

\*

\* @return string

\*/

**public** **function** **getDescription**()

{

**return** $this**->**description;

}

/\*\*

\* Get users

\*

\* @return \Doctrine\Common\Collections\Collection|array

\*/

**public** **function** **getUsers**()

{

**return** $this**->**users;

}

/\*\*

\* Set users

\*

\* @param \AppBundle\Entity\User $users

\* @return Skill

\*/

**public** **function** **setUsers**($users)

{

**return** $this**->**users[] **=** $users;

}

/\*\*

\* Get courses

\*

\* @return \Doctrine\Common\Collections\Collection|array

\*/

**public** **function** **getCourses**()

{

**return** $this**->**courses;

}

/\*\*

\* Set courses

\*

\* @param \AppBundle\Entity\Course $courses

\* @return Skill

\*/

**public** **function** **setCourses**($courses)

{

**return** $this**->**courses[] **=** $courses;

}

}

*Special.php*

**<?php**

**namespace** AppBundle\Entity;

**use** Doctrine\ORM\Mapping **as** ORM;

/\*\*

\* Special

\*

\* @ORM\Table(name="special")

\* @ORM\Entity(repositoryClass="AppBundle\Repository\SpecialRepository")

\*/

**class** **Special**

{

/\*\*

\* @var int

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

**private** $id;

/\*\*

\* @var string

\*

\* @ORM\Column(name="name", type="string", length=255)

\*/

**private** $name;

/\*\*

\* @var string

\*

\* @ORM\Column(name="description", type="text", nullable=true)

\*/

**private** $description;

/\*\*

\*

\* @ORM\ManyToMany(targetEntity="Skill", inversedBy="specials" )

\* @ORM\JoinTable(name="skill\_special",

\* joinColumns={@ORM\JoinColumn(name="special\_id", referencedColumnName="id")},

\* inverseJoinColumns={@ORM\JoinColumn(name="skill\_id", referencedColumnName="id")}

\* )

\*/

**private** $skills;

/\*\*

\*

\* @ORM\ManyToMany(targetEntity="User", inversedBy="specials")

\* @ORM\JoinTable(name="users\_special",

\* joinColumns={@ORM\JoinColumn(name="special\_id", referencedColumnName="id")},

\* inverseJoinColumns={@ORM\JoinColumn(name="user\_id", referencedColumnName="id")}

\* )

\*/

**private** $users;

/\*\*

\* Get id

\*

\* @return int

\*/

**public** **function** **getId**()

{

**return** $this**->**id;

}

/\*\*

\* Set name

\*

\* @param string $name

\*

\* @return Special

\*/

**public** **function** **setName**($name)

{

$this**->**name **=** $name;

**return** $this;

}

/\*\*

\* Get name

\*

\* @return string

\*/

**public** **function** **getName**()

{

**return** $this**->**name;

}

/\*\*

\* Set description

\*

\* @param string $description

\*

\* @return Special

\*/

**public** **function** **setDescription**($description)

{

$this**->**description **=** $description;

**return** $this;

}

/\*\*

\* Get description

\*

\* @return string

\*/

**public** **function** **getDescription**()

{

**return** $this**->**description;

}

/\*\*

\* Set skills

\*

\* @param string $skills

\*

\* @return Special

\*/

**public** **function** **setSkills**($skills)

{

$this**->**skills **=** $skills;

**return** $this;

}

/\*\*

\* Get skills

\*

\* @return string

\*/

**public** **function** **getSkills**()

{

**return** $this**->**skills;

}

/\*\*

\* Set users

\*

\* @param string $users

\*

\* @return Special

\*/

**public** **function** **setUsers**($users)

{

$this**->**users **=** $users;

**return** $this;

}

/\*\*

\* Get users

\*

\* @return string

\*/

**public** **function** **getUsers**()

{

**return** $this**->**users;

}

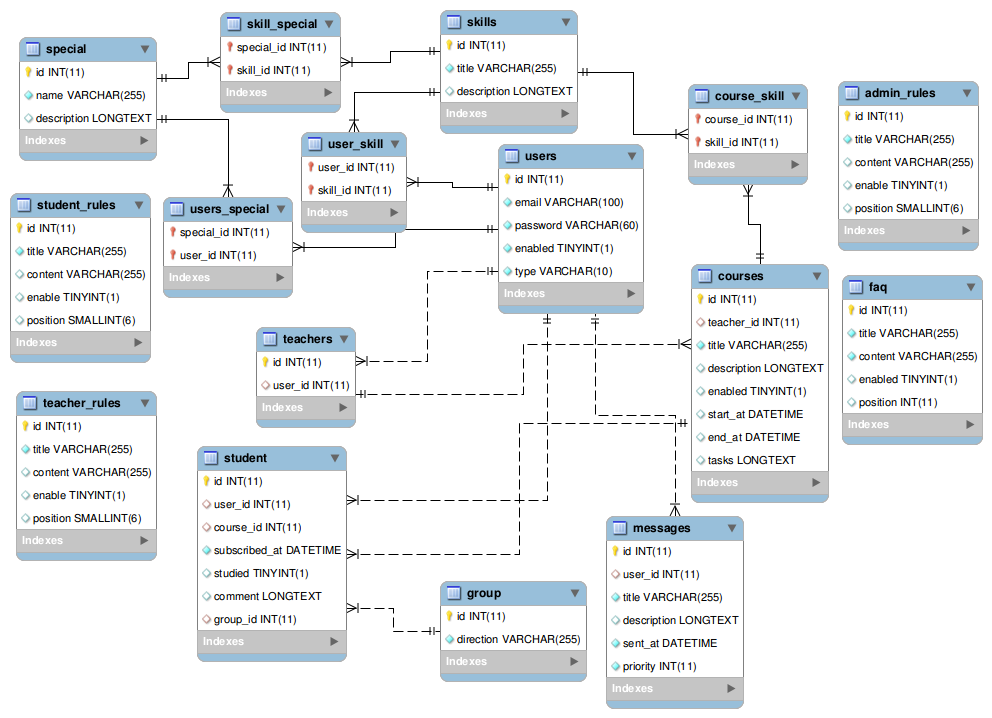
}

**7** **МОДЕЛИРОВАНИЕ ДАННЫХ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СРЕДЕ АВТОМАТИЗИРОВАННОГО СИНТЕЗА**

Для проектирования базы данных был использован инструмент MySQL Workbench, который позволяет визуально проектировать базу данных.

Схема спроектированнной базы данных представлена на рисунке 3.

Рисунок 3 – Схема базы данных



**8 СОЗДАНИЕ WEB-ПРИЛОЖЕНИЙ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СРЕДЕ АВТОМАТИЗИРОВАННОГО СИНТЕЗА**

Был создан проект в виде Web-приложения.

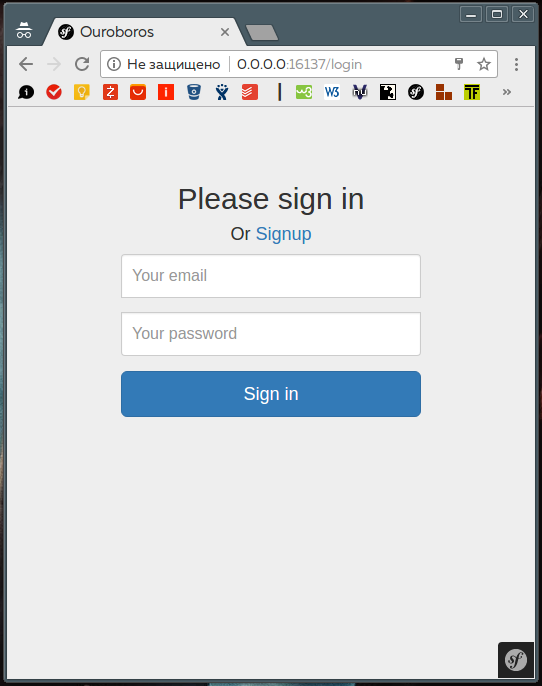


Рисунок 4 – Страница входа в систему

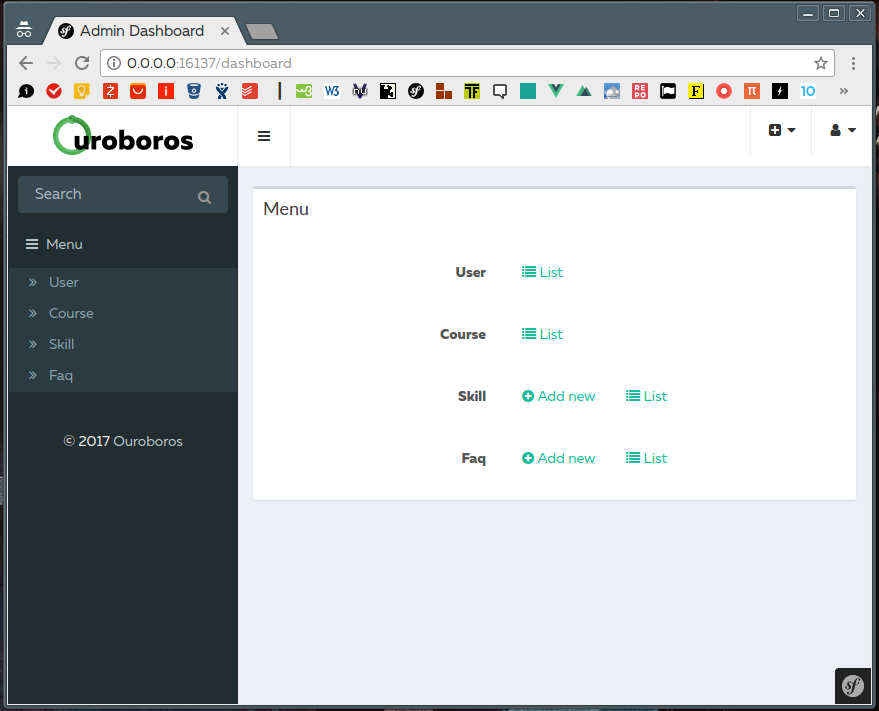


Рисунок 5 – Главный экран

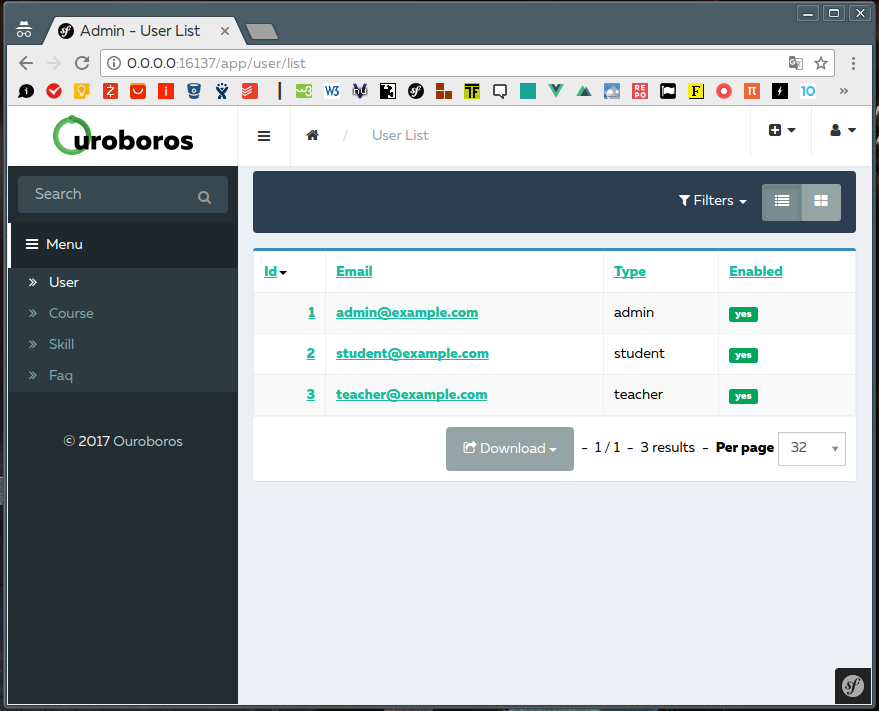


Рисунок 6 – Список пользователей системы

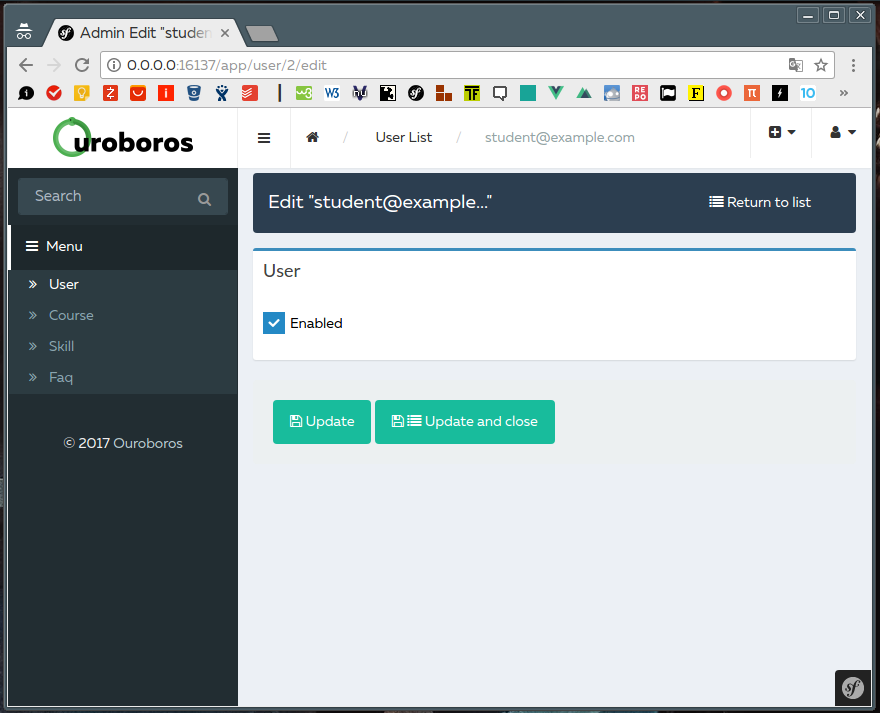


Рисунок 7 – Активация пользователя

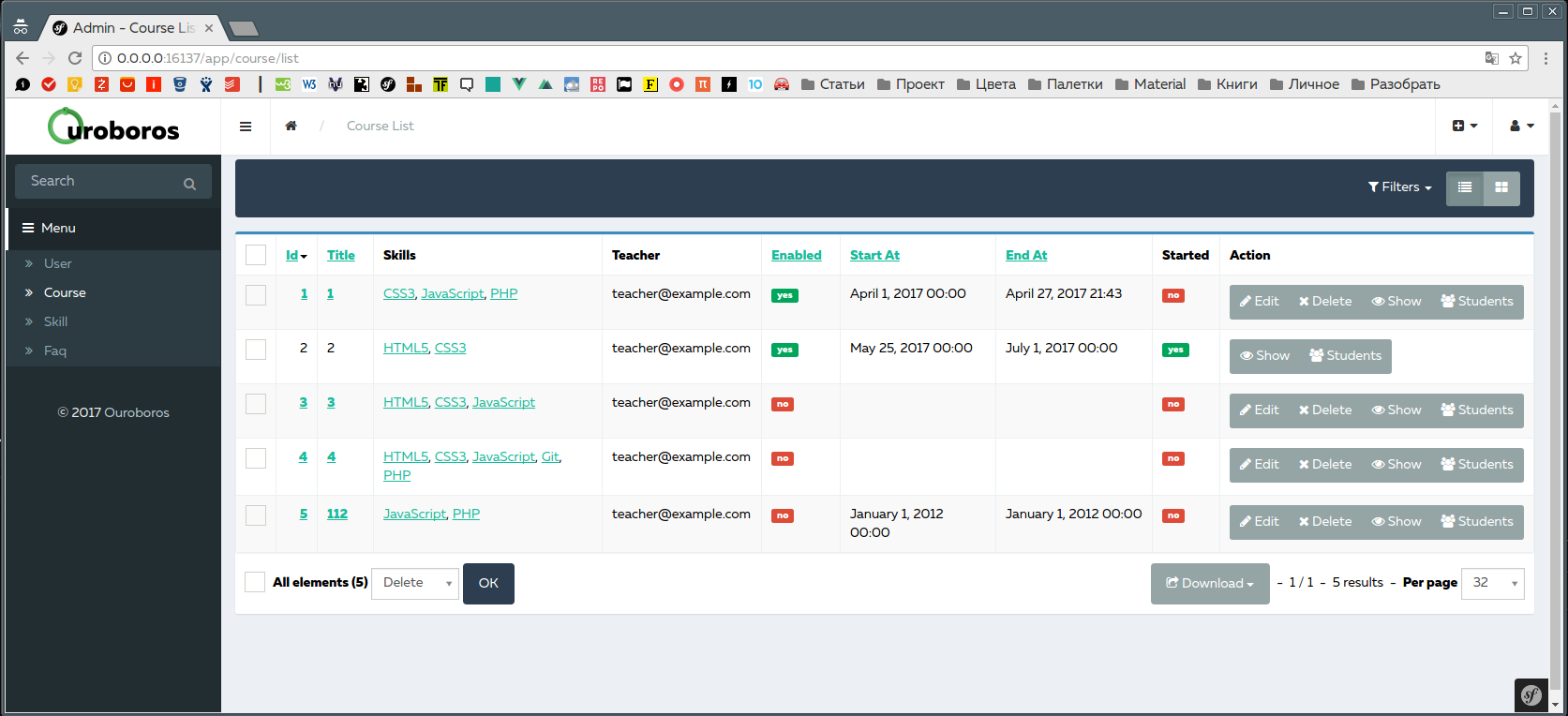


Рисунок 8 – Список курсов

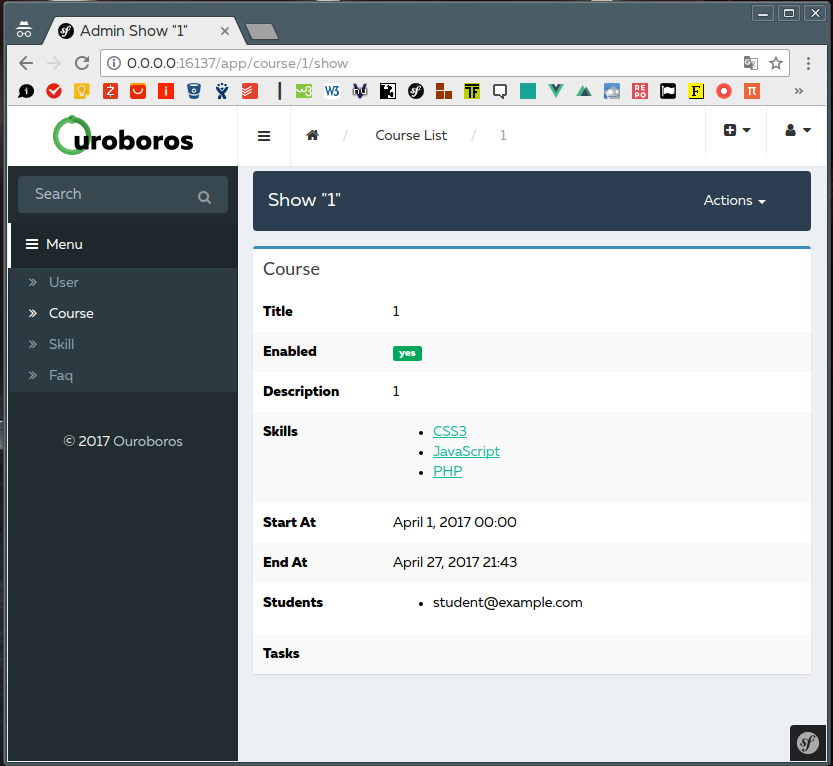


Рисунок 9 – Просмотр курса

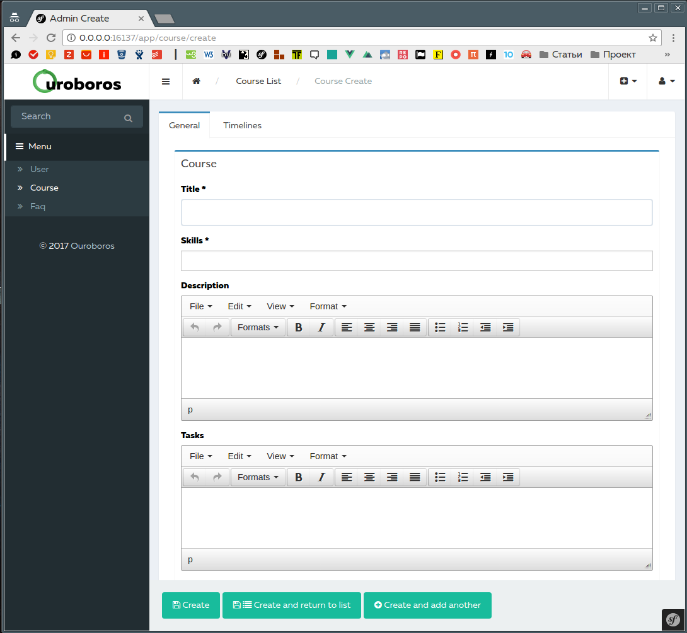


Рисунок 10 – Добавление курса

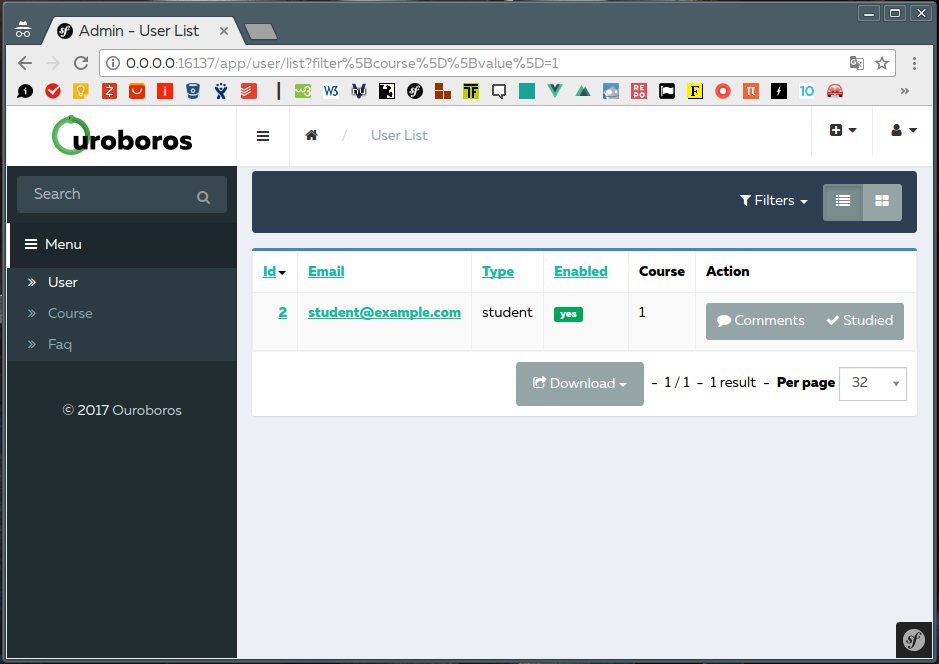


Рисунок 11 – Отметка о прохождении студентом курса

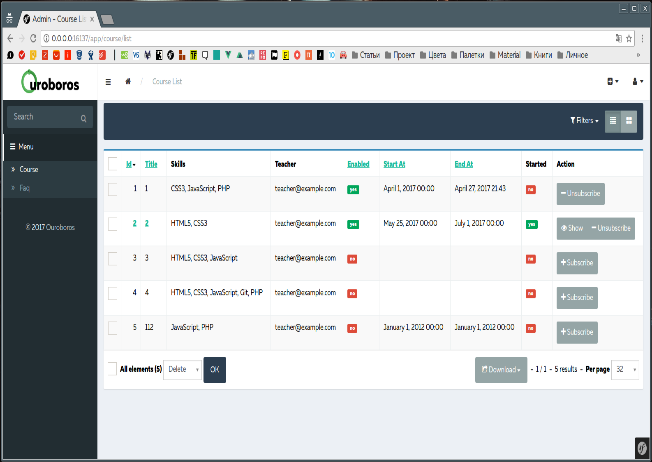


Рисунок 12 – Страница для подписки или отписки от курсов для студента