

Учебный центр

Sergey Makarov

21 апреля 2020 г.

Содержание

1	Концептуальная модель UI	1
1.1	Сценарии использования	1
1.1.1	Получение списков обучающихся по курсам	1
1.1.2	Получение истории обучения конкретного человека	1
1.1.3	Получение расписания преподавателя	1
1.1.4	Получение расписания учащегося	1
1.1.5	Управление данными обучающихся	1
1.1.6	Управление данными преподавателей	1
1.1.7	Управление курсами	2
1.2	Страницы приложения	2
1.2.1	Общие положения	2
1.2.2	Главная страница	2
1.2.3	Страница обучающегося	2
1.2.4	Страница преподавателя	2
1.2.5	Страница администратора курса	2
1.2.6	Страница курса	3
1.2.7	Страница регистрации	3
1.2.8	Страница авторизации	3
1.3	Схема навигации между страницами	3
2	База данных	4
2.1	Схема базы данных	4
2.2	Скрипт создания базы данных	4
2.3	Скрипт заполнения базы данных	6
2.4	Скрипт очистки базы данных	7

1 Концептуальная модель UI

1.1 Сценарии использования

1.1.1 Получение списков обучающихся по курсам

Чтобы получить список обучающихся на курсе, нужно авторизоваться как преподаватель и зайти на страницу курса, который этот преподаватель ведёт. На этой странице будет список ссылок на страницы обучающихся, записанных на курс.

1.1.2 Получение истории обучения конкретного человека

Чтобы получить историю обучения пользователя, нужно зайти на его страницу. Если обучающийся разрешил отображение своей истории обучения, она будет представлена на этой странице.

1.1.3 Получение расписания преподавателя

Получить расписание преподавателя можно зайдя на его страницу и нажав соответствующую кнопку.

1.1.4 Получение расписания учащегося

Получить расписание учащегося можно зайдя на его страницу и нажав соответствующую кнопку.

1.1.5 Управление данными обучающихся

Авторизовавшись, обучающийся может редактировать данные о себе, хранимые в системе. Для этого на его странице есть кнопка редактирования страницы, добавляющая к полям ФИО, почты, биографии и контактов формы для установки новых значений соответствующих полей, и кнопку применения изменений, фиксирующую изменения в базе. Кроме того, рядом с каждой формой при редактировании данных отображается чекбокс, определяющий, нужно ли показывать данную информацию на странице пользователя.

1.1.6 Управление данными преподавателей

Авторизовавшись, преподаватель может редактировать данные о себе, хранящиеся в системе. Для этого на его странице есть кнопка редактирования страницы, добавляющая к полям ФИО, почты, названия компании, биографии и контактов формы для установки новых значений соответствующих полей, и кнопку применения изменений, фиксирующую изменения в базе. Кроме того, рядом с каждой формой при редактировании данных отображается чекбокс, определяющий, нужно ли показывать данную информацию на странице пользователя.

1.1.7 Управление курсами

Авторизовавшись как администратор курса, пользователь может создавать новые курсы по ссылке со страницы администратора курса, либо же перейти по ссылкам с этой же страницы к любому из администрируемых курсов. На странице курса, помимо этого, есть кнопка редактирования описания курса, форма добавления новых администраторов со списком уже добавленных, и чекбокс, определяющий, доступен ли курс пользователям, отличным от администраторов.

1.2 Страницы приложения

1.2.1 Общие положения

В шапке каждой страницы приведены ссылки на главную страницу и страницу авторизации/выхода. При переходе по ссылке на страницу выхода пользователь попадает на ту же страницу, если она доступна неавторизованному пользователю, либо на главную.

1.2.2 Главная страница

На главной странице приведён список курсов в виде ссылок на них и поле поиска курса. В результате поиска на главной будут отображаться только курсы, удовлетворяющие условию поиска. Кроме того, на главной странице в шапке приведена ссылка на личную страницу пользователя и кнопка выхода, если пользователь авторизован, и ссылки на страницы авторизации и регистрации в противном случае. Большая часть функциональности приложения доступна только после авторизации.

1.2.3 Страница обучающегося

На странице обучающаяся приведено ФИО обучающегося, общая информация о нём, его контактные данные, если обучающийся разрешил их отображение, и набор курсов, которые обучающийся проходил или проходит, также если обучающийся разрешил отображение этих данных. Также на странице обучающегося есть кнопка составления расписания, которая позволяет вывести расписание обучающегося на заданный интервал. Если обучающийся авторизован, он может менять данные своей страницы и управлять их доступностью, как сказано ранее. При нажатии на кнопку редактирования страницы возле полей ФИО, общей информации и контактных данных появляются формы ввода новых значений и чекбокс, определяющий, нужно ли отображать эти данные на странице. Изменения применяются после нажатия кнопки сохранения изменений.

1.2.4 Страница преподавателя

На странице преподавателя отображается ФИО преподавателя, его контактная информация, если включено отображение этой информации, название компании, в которой он работает, и ссылки на курсы, занятия в которых он ведёт. Также на странице преподавателя есть кнопка составления расписания, которая позволяет вывести расписание преподавателя на заданный интервал. Если пользователь авторизован как преподаватель, он может управлять отображением контактной информации на своей странице и редактировать данные на этой странице. При нажатии на кнопку редактирования страницы возле полей ФИО, общей информации и контактных данных появляются формы ввода новых значений и чекбокс, определяющий, нужно ли отображать эти данные на странице. Изменения применяются после нажатия кнопки сохранения изменений.

1.2.5 Страница администратора курса

На этой странице предоставлен список ссылок на администрируемые пользователем курсы. Кроме того, есть ссылка создания курса, с помощью которой создаётся новый курс, на страницу которого администратор и попадает. По умолчанию созданный курс не виден никому, кроме администратора.

1.2.6 Страница курса

На странице курса приведено описание курса, диапазон дат его проведения, его интенсивность, расписание и список ссылок на страницы преподавателей, ведущих курс. Расписание представлено в виде списка дат и времени проведения занятий. В случае, если пользователь авторизован как обучающийся, на странице есть также кнопка записи на курс, либо, если он уже записан, кнопка отмены записи. Преподавателю, ведущему курс, доступен список ссылок на страницы обучающихся, записанных на этот курс. На странице курса есть кнопка редактирования описания курса, доступная только для администраторов курса, при нажатии этой кнопки появляется форма для нового описания и кнопка сохранения изменений. Помимо этого, администратор курса может назначать и отзывать других администраторов курса (для этого есть форма, куда пишется почта добавляемого администратора), и показывать/скрывать курс от других пользователей (чекбокс).

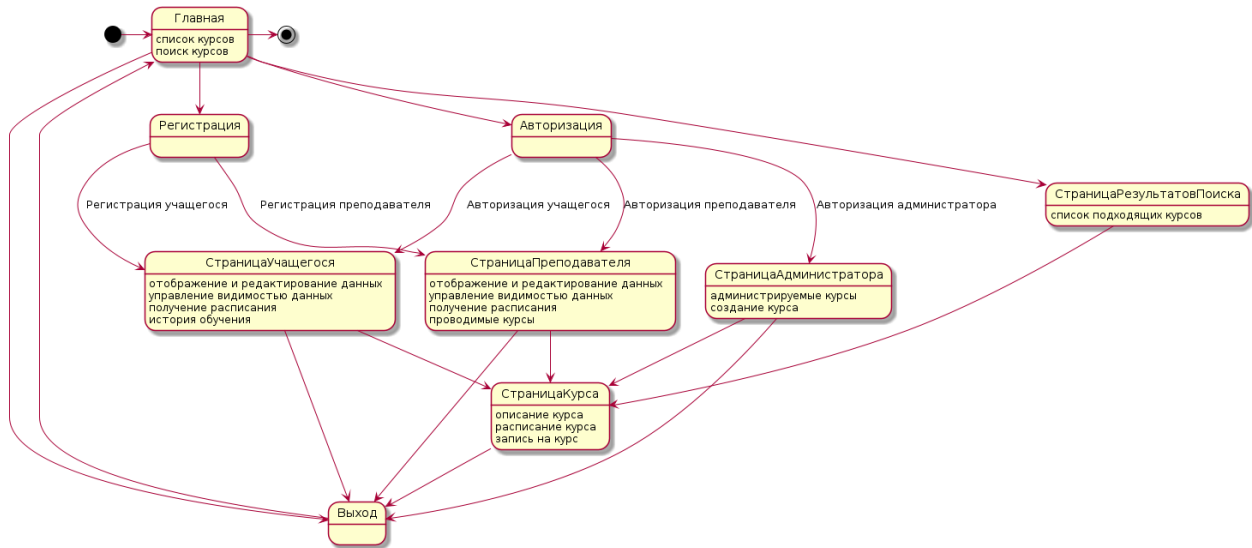
1.2.7 Страница регистрации

На странице регистрации можно зарегистрировать учётную запись для обучающегося или преподавателя. Учётная запись администратора курса создаётся вручную из соображений безопасности. Для регистрации обучающегося нужно предоставить электронную почту, логин и пароль, которые будут использоваться для входа, ФИО обучающегося и опциональную биографическую информацию, преподавателю дополнительно нужно предоставить название компании, в которой он работает.

1.2.8 Страница авторизации

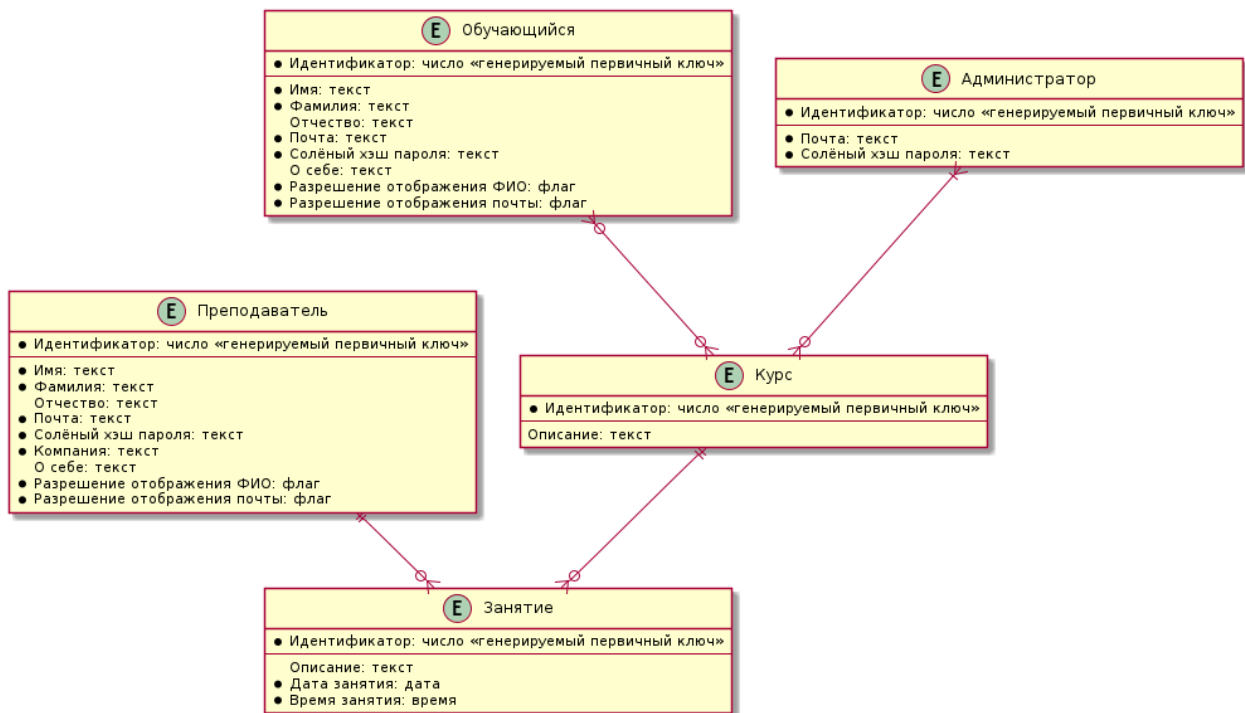
На странице авторизации можно войти как обучающийся, преподаватель, либо администратор курса. Во всех трёх случаях авторизация происходит с помощью логина и пароля. После авторизации как обучающийся, преподаватель или администратор курса пользователь попадает на страницу обучающегося, преподавателя или администратора курса соответственно.

1.3 Схема навигации между страницами



2 База данных

2.1 Схема базы данных



2.2 Скрипт создания базы данных

```
CREATE TABLE PUPIL (  
    id serial PRIMARY KEY,  
    firstname varchar(100) NOT NULL,  
    lastname varchar(100) NOT NULL,  
    patronymic varchar(100),  
    mail varchar(254) NOT NULL,  
    bio text,  
    pass_hash varchar(64) NOT NULL,  
    display_name boolean NOT NULL,  
    display_mail boolean NOT NULL);  
  
CREATE TABLE TEACHER (  
    id serial PRIMARY KEY,  
    firstname varchar(100) NOT NULL,  
    lastname varchar(100) NOT NULL,  
    patronymic varchar(100),  
    company_name varchar(256) NOT NULL,  
    mail varchar(254) NOT NULL,  
    bio text,  
    pass_hash varchar(64) NOT NULL,  
    display_name boolean NOT NULL,  
    display_mail boolean NOT NULL);  
  
CREATE TABLE ADMINISTRATOR (  
    id serial PRIMARY KEY,  
    mail varchar(254) NOT NULL,  
    pass_hash varchar(64) NOT NULL);  
  
CREATE TABLE COURSE (  
    id serial PRIMARY KEY,  
    admin_id integer REFERENCES ADMINISTRATOR,  
    description text,  
    is_displayed boolean NOT NULL);  
  
CREATE TABLE LESSON (  
    id serial PRIMARY KEY,  
    description text,  
    lesson_time timestamp NOT NULL,  
    course_id integer REFERENCES COURSE,  
    teacher_id integer REFERENCES TEACHER);  
  
CREATE TABLE ADMIN_COURSE (  
    id serial PRIMARY KEY,  
    admin_id integer REFERENCES ADMINISTRATOR,  
    course_id integer REFERENCES COURSE);  
  
CREATE TABLE PUPIL_COURSE (  
    id serial PRIMARY KEY,
```

```

pupil_id integer REFERENCES PUPIL,
course_id integer REFERENCES COURSE);

CREATE FUNCTION fix_courses() RETURNS TRIGGER AS $fix_courses$
DECLARE
    c RECORD;
    new_admin RECORD;
BEGIN
    FOR c IN
        SELECT id, description FROM COURSE
        WHERE COURSE.admin_id = OLD.id
    LOOP
        SELECT admin_id INTO new_admin FROM ADMIN_COURSE
        WHERE ADMIN_COURSE.course_id = course.id
        AND ADMIN_COURSE.admin_id != OLD.id;
        IF NOT FOUND THEN
            RAISE EXCEPTION 'Removing last admin of the course %', c.d;
        END IF;
        UPDATE COURSE SET admin_id = new_admin.admin_id WHERE id = c.id;
    END LOOP;
    RETURN OLD;
END;
$fix_courses$ LANGUAGE plpgsql;

CREATE TRIGGER fix_courses BEFORE DELETE ON ADMINISTRATOR
    FOR EACH ROW EXECUTE FUNCTION fix_courses();

```

2.3 Скрипт заполнения базы данных

```

SET DATESTYLE TO "DMY";

INSERT INTO PUPIL(firstname, lastname, mail, pass_hash, display_name, display_mail) VALUES('John',
'Doe', 'mail@example.com', 'fc4b5fd6816f75a7c81fc8eaa9499d6a299bd803397166e8c4cf9280b801d62c', TRUE,
INSERT INTO PUPIL(firstname, lastname, mail, pass_hash, display_name, display_mail) VALUES('Jane',
'Doe', 'jdoe@bing.com', '8070d7385ec2439ca5d8d415c003cfde188803fc20d7f89291883a8ac4d61c9d', TRUE, FALSE),
INSERT INTO PUPIL(firstname, lastname, mail, pass_hash, display_name, display_mail) VALUES('Jimm',
'Craig', 'craig@gmail.com', '8bc4b88b1827764692b001ccc6696a23204a9fac043135ebad830b2af6fc8924', TRUE, FALSE),
INSERT INTO PUPIL(firstname, lastname, mail, pass_hash, display_name, display_mail) VALUES('Kate',
'Robinson', 'kate1984@bing.com', '25032daca58d5d3e677fe090e5be9ffd2b1a7ef68d585dd05c3dcacf3fce41c5',
FALSE),

INSERT INTO TEACHER(firstname, lastname, company_name, mail, pass_hash, display_name, display_mail) VALUES('Kuznetsov', 'CS Center', 'mail@example.com', '6b3a55e0261b0304143f805a24924d0c1c44524821305f31d92778',
INSERT INTO TEACHER(firstname, lastname, company_name, mail, pass_hash, display_name, display_mail) VALUES('Tarasov', 'CS Center', 'taroleg@gmail.com', '44e9385f53069b91d4afaa12f93a6e3353f83cc8f1db1f030933264',
INSERT INTO TEACHER(firstname, lastname, company_name, mail, pass_hash, display_name, display_mail) VALUES('Moskvin', 'CS Center', 'chrome@google.com', 'c942bc47f4c98e6bda9666c229c1dc88e8ee73383d7c75de3dc',
INSERT INTO TEACHER(firstname, lastname, company_name, mail, pass_hash, display_name, display_mail) VALUES('Bugayenko', 'CS Center', 'yegor256@gmail.com', '520ac20de525cd5a8c8c39c39b46b71a090ce968ff60f8617375',
FALSE),

INSERT INTO ADMINISTRATOR(mail, pass_hash) VALUES('nicolas@nmattia.com', '84eab5c7b4f0530a708350454e7',

```

```

INSERT INTO ADMINISTRATOR(mail, pass_hash) VALUES('robert@stuffwithstuff.com', 'c074d9749aa92d447e63e');

INSERT INTO COURSE(description, admin_id, is_displayed) VALUES('Java programming', 1, TRUE);
INSERT INTO COURSE(description, admin_id, is_displayed) VALUES('Databases', 2, TRUE);

INSERT INTO LESSON(description, lesson_time, course_id, teacher_id) VALUES('Introduction',
'12-03-2020 16:20:00', 1, 2);
INSERT INTO LESSON(description, lesson_time, course_id, teacher_id) VALUES('Introduction',
'14-03-2020 16:55:00', 2, 3);
INSERT INTO LESSON(description, lesson_time, course_id, teacher_id) VALUES('Object-oriented programming',
'19-03-2020 16:20:00', 1, 4);
INSERT INTO LESSON(description, lesson_time, course_id, teacher_id) VALUES('History of databases',
'12-03-2020 16:55:00', 2, 4);

INSERT INTO ADMIN_COURSE(admin_id, course_id) VALUES(1, 1);
INSERT INTO ADMIN_COURSE(admin_id, course_id) VALUES(2, 2);

INSERT INTO PUPIL_COURSE(pupil_id, course_id) VALUES(1, 1);
INSERT INTO PUPIL_COURSE(pupil_id, course_id) VALUES(2, 2);
INSERT INTO PUPIL_COURSE(pupil_id, course_id) VALUES(3, 2);
INSERT INTO PUPIL_COURSE(pupil_id, course_id) VALUES(4, 1);

```

Проверим заполнение таблиц:

```
SELECT * FROM PUPIL;
```

```
SELECT * FROM TEACHER;
```

```
SELECT * FROM ADMINISTRATOR;
```

```
SELECT * FROM COURSE;
```

```
SELECT * FROM LESSON;
```

2.4 Скрипт очистки базы данных

```
DROP TRIGGER fix_courses ON ADMINISTRATOR;
```

```
DROP FUNCTION fix_courses();
```

```
DROP TABLE PUPIL_COURSE;
```

```
DROP TABLE ADMIN_COURSE;
```

```
DROP TABLE PUPIL CASCADE;
```

```
DROP TABLE COURSE CASCADE;
```

```
DROP TABLE LESSON CASCADE;
```

```
DROP TABLE TEACHER CASCADE;
```

```
DROP TABLE ADMINISTRATOR CASCADE;
```