МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ФГБО ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ

(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)»

ИЖЕНЕРНО-ЭКОНОМИЧЕСКИЙ ИНСТИТУТ

Кафедра «Прикладная и бизнес-информатика»

ОТЧЕТ ПО КУРСОВОМУ ПРОЕКТУ

по дисциплине «Управление разработкой информационных систем»

Тема: «Конструктор расписания учебных занятий для образовательных учреждений среднего профессионального образования»

Студентка группы Стешец Е.М. ИЭс-161-16

(Ф.И.О.)

Руководитель к.т.н., Петров С.А.

(уч. степень, звание, Ф.И.О.)

|  |  |  |
| --- | --- | --- |
| Сдана на проверку |  |  |
| Возвращена на доработку |  |  |
| Допущена к защите |  |  |
| Оценка |  |  |
|  |

Москва 2018г.

Оглавление

[1.Описание предметной области 3](#_Toc514935625)

[2. Обоснование выбора объекта автоматизации 6](#_Toc514935626)

[2.1. Назначение, цели создания информационной системы «Конструктор расписания». 6](#_Toc514935627)

[2.2. Общее описание 6](#_Toc514935628)

[2.3. Требования к системе. 6](#_Toc514935629)

[2.4. Требования к функциональным характеристикам. 7](#_Toc514935630)

[2.5. Функциональное назначение системы. 7](#_Toc514935631)

[2.6. Не функциональное назначение системы. 8](#_Toc514935632)

[2.7. Диаграмма вариантов использования. 8](#_Toc514935633)

[3. Описание выбранных для реализации технологий 10](#_Toc514935634)

[4. Проектирование информационной системы 12](#_Toc514935635)

[4.1. Диаграмма базы данных «Конструктора расписания». 14](#_Toc514935636)

[5. Реализация информационной системы 23](#_Toc514935637)

[Заключение 33](#_Toc514935638)

[Литература 34](#_Toc514935639)

## 1.Описание предметной области

Предметной областью информационной системы (ИС) была выбрана деятельность «Конструктора расписания учебных занятий для образовательных учреждений среднего профессионального образования».

С помощью данной ИС осуществляется облегчение работы персонала. «Конструктор расписания» предназначен для внесения, хранения, изменения и предоставления информации, так как данные находятся в электронном виде, человек с ролью Администратора (методист) может зайти с любого устройства, зная логин и пароль, и, осуществить все нужные действия, связанные с дисциплинами, группами, преподавателями и так далее. Также данная ИС сокращает время составления расписания, потому что в бумажном виде это происходит гораздо дольше.

В составлении расписания множество тонкостей и нюансов, которые не учитываются разработчиками или учесть просто не возможно, поэтому данная ИС лишь облегчает процесс работы методиста, и его невозможно автоматизировать целиком.

Федеральный закон Российской Федерации от 27 июля 2006 г. N 149-ФЗ «Об информации, информационных технологиях и о защите информации» даёт следующее определение: «информационная система — совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств».

**Основные задачи информационных систем:**

* **Интерпретация данных**. Под интерпретацией понимается процесс определения смысла данных, результаты которого должны быть согласованными и корректными. Обычно предусматривается многовариантный анализ данных.
* **Диагностика.** Под диагностикой понимается процесс соотношения объекта с некоторым классом объектов и/или обнаружение неисправности в некоторой системе. Неисправность - это отклонение от нормы. Такая трактовка позволяет с единых теоретических позиций рассматривать и неисправность оборудования в технических системах, и заболевания живых организмов, и всевозможные природные аномалии.
* **Мониторинг.** Основная задача мониторинга - непрерывная интерпретация данных в реальном времени и сигнализация о выходе тех или иных параметров за допустимые пределы.
* **Проектирование.** Проектирование состоит в подготовке спецификаций на создание «объектов» с заранее определёнными свойствами. Под спецификацией понимается весь набор необходимых документов - чертёж, пояснительная записка и т.д. Основные проблемы здесь - получение чёткого структурного описания знаний об объекте и проблема «следа».
* **Прогнозирование.** Прогнозирование позволяет предсказывать последствия некоторых событий или явлений на основании анализа имеющихся данных. Прогнозирующие системы логически выводят вероятные следствия из заданных ситуаций.
* **Планирование.** Под планированием понимается нахождение планов действий, относящихся к объектам, способным выполнять некоторые функции. В таких ЭС используются модели поведения реальных объектов с тем, чтобы логически вывести последствия планируемой деятельности.

Обучение. Под обучением понимается использование компьютера для обучения какой-то дисциплине или предмету. Системы обучения диагностируют ошибки при изучении какой-либо дисциплины с помощью ЭВМ и подсказывают правильные решения.

* **Управление.** Под управлением понимается функция организованной системы, поддерживающая определенный режим деятельности. Такого рода ЭС осуществляют управление поведением сложных систем в соответствии с заданными спецификациями.
* **Поддержка принятия решений.** Поддержка принятия решения - это совокупность процедур, обеспечивающая лицо, принимающее решения, необходимой информацией и рекомендациями, облегчающие процесс принятия решения. Эти ЭС помогают специалистам выбрать и/или сформировать нужную альтернативу среди множества выборов при принятии ответственных решений.

Информационная резервирования гостиничных номеров должна обеспечить:

* Ввод, изменение, удаление данных об образовательном учреждении;
* Ввод, изменение, удаление данных о подразделении;
* Ввод, изменение, удаление данных об образовательных программах, учебных периодах, преподавателях, дисциплинах, учебных группах и кабинетах
* Ввод, изменение, удаление данных об учебных периодах,
* Ввод, изменение, удаление данных о преподавателях,
* Ввод, изменение, удаление данных о дисциплинах,
* Ввод, изменение, удаление данных об учебных группах
* Ввод, изменение, удаление данных о кабинетах
* Составление из введенных данных расписания, его изменение или удаление
* Справочная информация
* Вход для сотрудника, используя логин и пароль.

## 2. Обоснование выбора объекта автоматизации

## 2.1. Назначение, цели создания информационной системы «Конструктор расписания».

**Назначение ИС:**

Информационная система предназначена для упрощения работы методиста, а так же сокращения затрат времени. Данный работник образовательного учреждения имеет роль Администратора системы и обеспечивает внесение, редактирование и удаление информации.  
**Цели создания ИС:**

Целью является разработка программного обеспечения для автоматизации деятельности сотрудников учебного учреждения СПО. Она автоматизирует работу по составлению расписания, а именно: внесению данных об образовательном учреждении, подразделении, образовательных программах, учебных периодах, преподавателях, дисциплинах, учебных группах и кабинетах.  
**Характеристика объектов автоматизации:**

В данном курсовом проекте в качестве исследуемой организации рассматривается образовательное учреждение среднего профессионального образования, для осуществления работы которого, составляется расписание.

## 2.2. Общее описание

Информационная система представляет собой хранение данных, которое обеспечивает доступность, целостность и конфиденциальность.

## 2.3. Требования к системе.

Основное требование к системе – дружественный интерфейс пользователя. Кроме того, следует учесть, что система ориентирована на конечного пользователя, который не должен знать структуру программного кода, внутренние представление данных, средства программирования, которые использовались при разработке системы.

Кроме того, работа в приложении должна быть построена наиболее эффективно, то есть должны быть реализованы самые необходимые пользователю функции. Пользователь должен иметь доступ ко всем данным в БД, а также иметь возможность для добавления, редактирования, удаления.   
С информационной системой «Конструктор расписания» должен работать квалифицированный сотрудник, который имеет опыт работы в составлении расписания.

## 2.4. Требования к функциональным характеристикам.

Система должна обеспечить возможность выполнения перечисленных ниже свойств:

* Гибкость – настраивается на любую структуру, для широкого круга деятельности;
* Безопасность – ограничение общего доступа к заранее определенным ресурсам БД;
* Целостность и доступность – документы в базе данных должны быть представлены в общепринятых в учреждении форматах, защищенных от изменений.

## 2.5. Функциональное назначение системы.

Программное изделие должно выполнять следующие функции:

* Ввод, изменение, удаление данных об образовательном учреждении;
* Ввод, изменение, удаление данных о подразделении;
* Ввод, изменение, удаление данных об образовательных программах, учебных периодах, преподавателях, дисциплинах, учебных группах и кабинетах
* Ввод, изменение, удаление данных об учебных периодах,
* Ввод, изменение, удаление данных о преподавателях,
* Ввод, изменение, удаление данных о дисциплинах,
* Ввод, изменение, удаление данных об учебных группах
* Ввод, изменение, удаление данных о кабинетах
* Составление из введенных данных расписания, его изменение или удаление
* Справочная информация
* Вход для сотрудника, используя логин и пароль.

## 2.6. Не функциональное назначение системы.

* ИС должна быть написана на (СУБД) MySQL, веб-сервере Apache, язык программирования PHP и фреймворк Yii2.
* ИС должна работать на ОС Windows.

## 2.7. Диаграмма вариантов использования.

Субъект (actor) — это некто или нечто (человек, машина и т.д.) взаимодействующее с системой. Субъект взаимодействует с прецедентом, ожидая получить некий полезный результат.

Типичным графическим изображением субъекта является «штриховой человечек». В общем случае субъект может быть показан в виде прямоугольного символа класса. Подобно обычному классу субъект может обладать атрибутами и операциями (связанными с событиями, сообщения о которых он отправляет и получает). На рисунке 1 показан субъекты: Администратор и Гость.

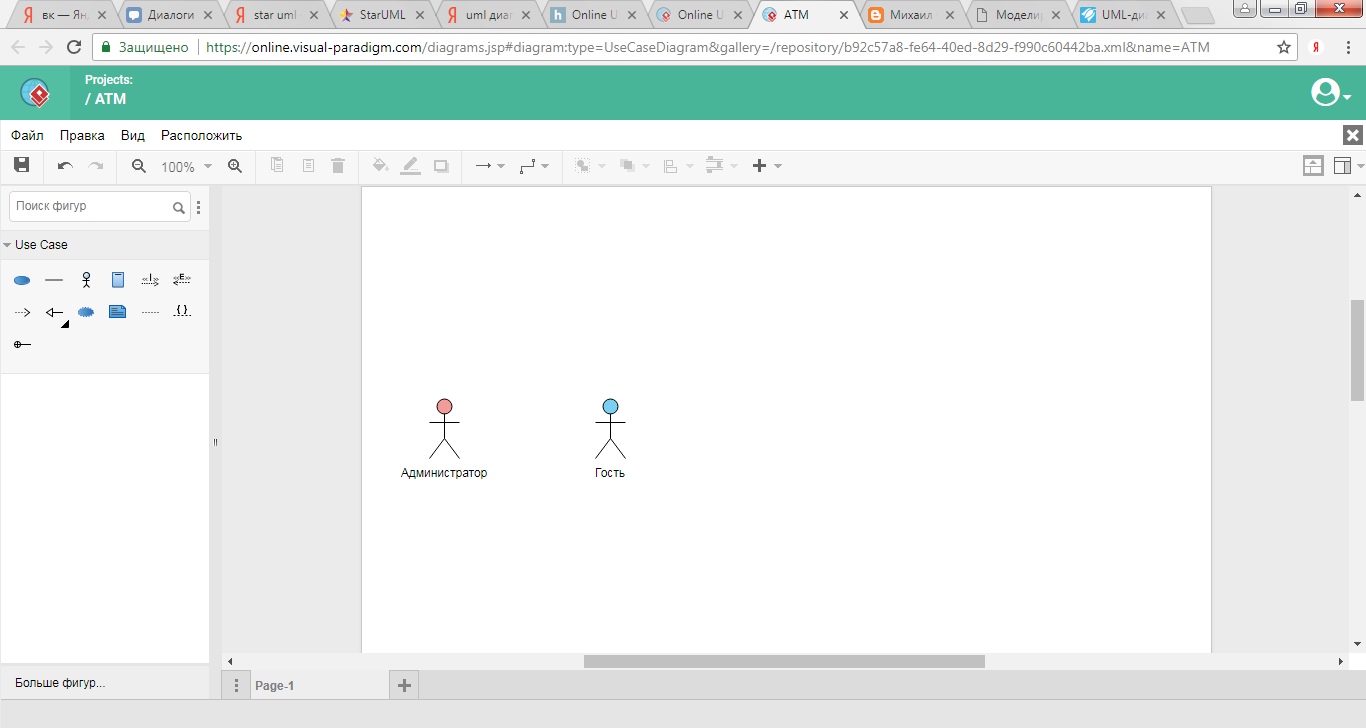


Рис.1 Субъекты системы «Конструктор расписания»

Прецеденты удовлетворяют функциональные требования за счет предоставления субъекту полезного результата. При этом не имеет значения, в какой последовательности решает бизнес-аналитик свои задачи: сначала обозначает субъектов, а затем прецеденты, или наоборот. Прецедент наглядно показывает варианты использования системы. На рисунке 2 показаны прецеденты моделируемой системы.

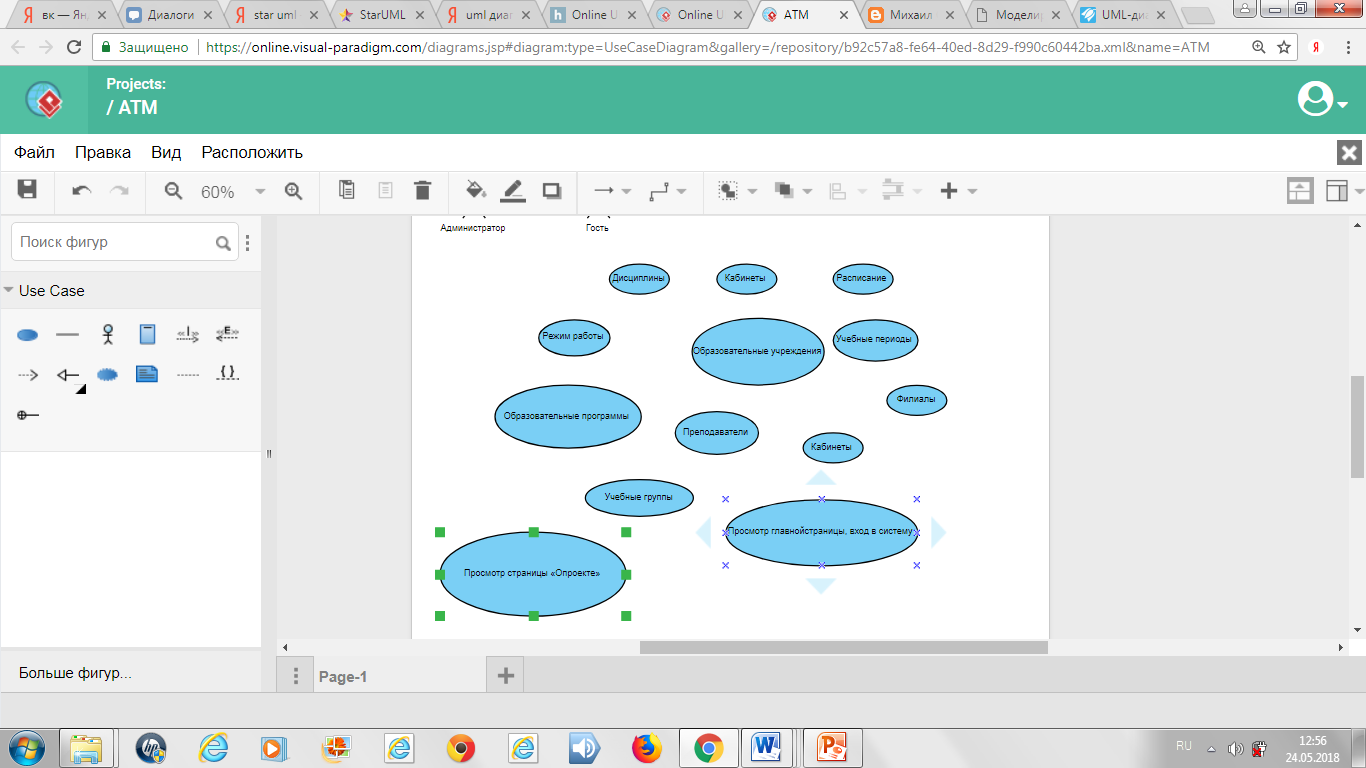


Рис. 2 Прецеденты системы «резервирования гостиничных номеров»

Диаграмма вариантов использования – это наглядное графическое представление субъектов и прецедентов и их взаимодействий в системе вместе с любыми дополнительными определениями и спецификациями. Она представляет собой не просто некую схему, а является полностью документированной моделью предполагаемого поведения системы. Диаграмма прецедентов для моделируемой системы представлена на   
рисунке 3.

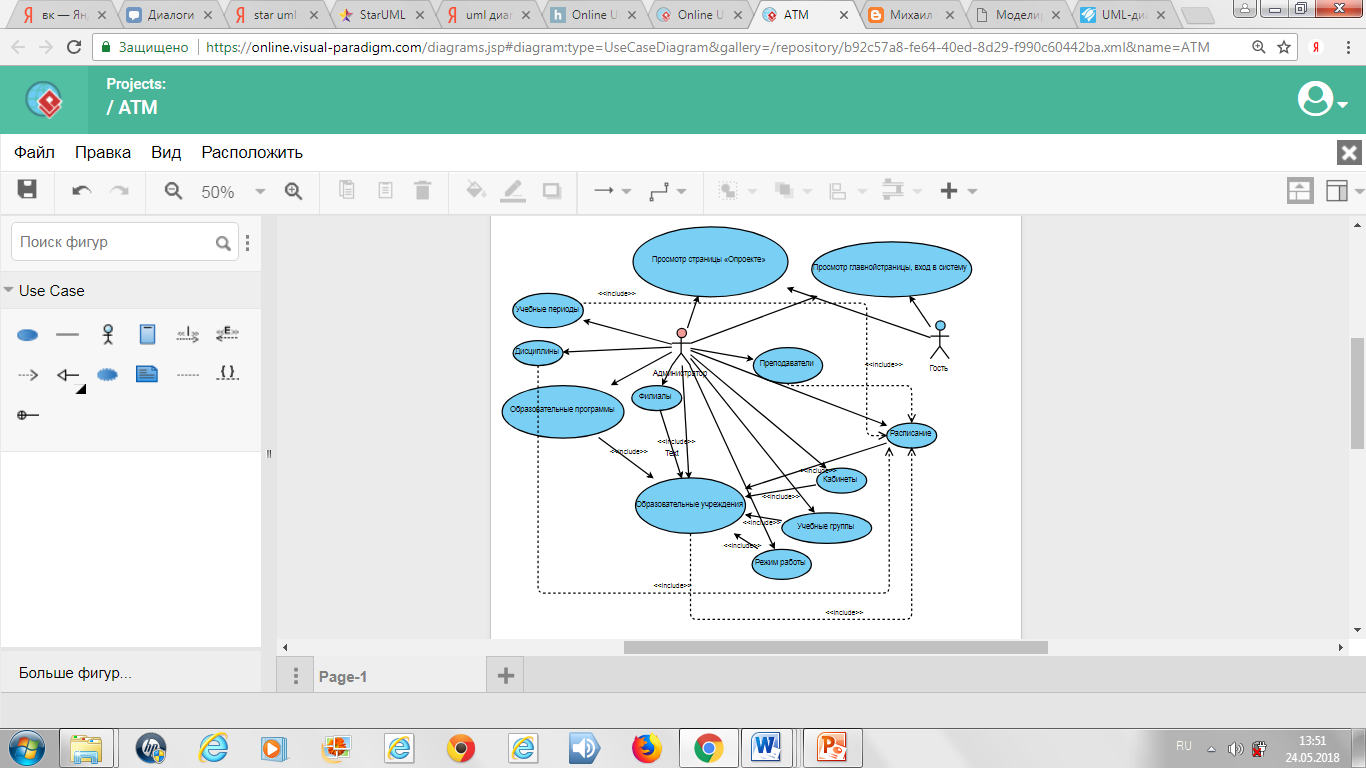


Рис. 3 Диаграмма вариантов использования для системы «Конструктор расписания»

## 3. Описание выбранных для реализации технологий

В качестве технологий для разработки системы выступают:

* (СУБД) MySQL,
* веб-сервер Apache,
* язык программирования PHP
* фреймворк Yii2

1) Фреймворк Yii2 — в качестве среды разработки программы «Конструктор расписания учебных занятий для образовательных учреждений среднего профессионального образования », был выбран Yii2, так как наиболее популярный среди фреймворков.

Yii2 – быстрый и профессиональный PHP фреймворк ,обладает такими характеристиками как: MVC, DAO/ActiveRecord, тестированием, открытым кодом, высокой производительностью. Используется пространство времен, поэтому компоненты, виджеты и модели изолированы друг от друга. При такой организации структуры проекта код писать значительно проще чем в первой части Yii.

К плюсам данного фреймворка относятся: легкость в освоении API, простота и быстродействие, полноценное использование MVC архитектуры, удобство установки, легкость тестирования, большое количество расширений.

Еще в данном фреймворке учтена зашита от наиболее распространенных угроз: XSS, XSRF, SQL injection и другие. Если использовать совокупность современных методов защиты информации, то никакой опасный код не попадет в систему.

2) Apache - основными достоинствами считаются надёжность и гибкость конфигурации, позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, и т.д. Он работает на таких операционные системы как MS Windows, Mac OS, BSD, Linux и еще некоторые.

Из-за того что Apache отличается гибкостью в работе, имеет множество настроек, хранящихся в конфигурационном файле, а так же есть возможность в подключении внешних модулей, интегрируется с другим ПО и языками программирования (php, Python, Ruby, ASP, Perl) он и используется таким количеством сайтов.

3) MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. Данная БД используется для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

4) PHP - скриптовый язык общего назначения, позволяющий разрабатывать веб-приложения, и в настоящее время поддерживающийся большим количеством хостинг-провайдеров.

## 4. Проектирование информационной системы

Проектирование информационных систем всегда начинается с определения цели проекта. Основная задача любого успешного проекта заключается в том, чтобы на момент запуска системы и в течение всего времени ее эксплуатации можно было обеспечить:

* требуемую функциональность системы и степень адаптации к изменяющимся условиям ее функционирования;
* требуемую пропускную способность системы;
* безотказную работу системы в требуемом режиме, иными словами - готовность и доступность системы для обработки запросов пользователей;
* простоту эксплуатации и поддержки системы;
* необходимую безопасность.

Производительность является главным фактором, определяющим эффективность системы. Хорошее проектное решение служит основой высокопроизводительной системы.

Проектирование информационных систем 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.

В реальных условиях проектирование - это поиск способа, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений.

К любому проекту предъявляется ряд абсолютных требований, например, максимальное время разработки проекта, максимальные денежные вложения в проект и т.д. Одна из сложностей проектирования состоит в том, что оно не является такой структурированной задачей, как анализ требований к проекту или реализация того или иного проектного решения.

Yii2 – php фреймворк, с открытым исходным кодом, который лежит на Github.

В Yii2 приложения организованы в соответствии с MVC. Шаблон MVC описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и, конечно же, реализуется.

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

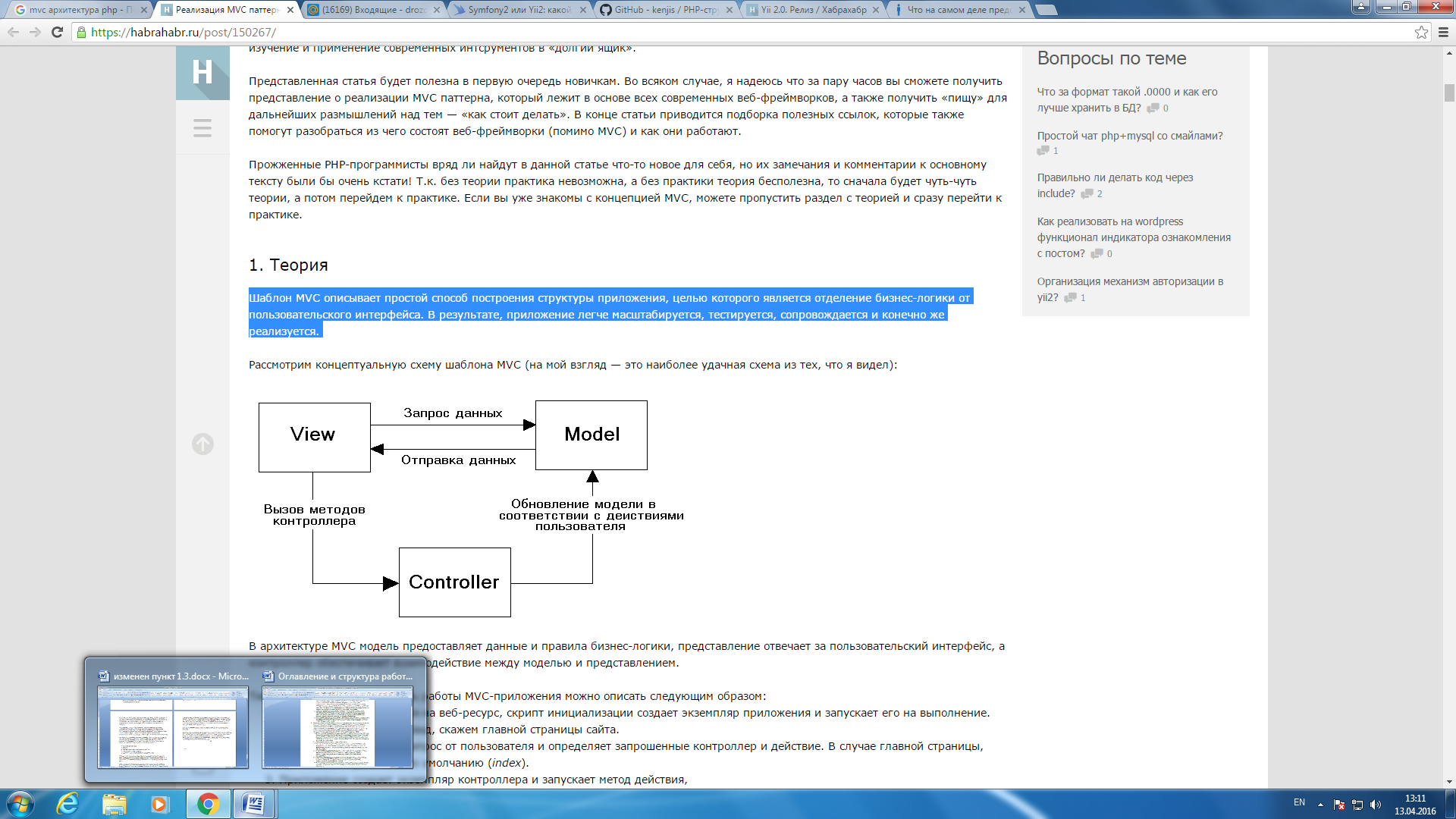


Рис.4 Модель MVC

Models – часть MVC архитектуры, являются объектами, представляющими коммерческие данные, правила и логику. Можно составлять классы модели путем расширения Yii\Base\Model или его дочерних классов.

Views – виды – часть MVC архитектуры, код который отвечает за представления данных конечным пользователям. В веб- приложениях обычно представляется как виды шаблонов (views templates) , которые по сути PHP скрипты, в основном содержащие HTML и PHP коды, отвечающие за представление и внешний вид. Виды управляются компонентами приложения View, который содержит часто используемые методы для упорядочивания видов и их рендеринга.

Controllers – часть MVC архитектуры, объект классов, унаследованных от yii\base\Controller и отвечающая за обработку запроса приложениями (applications), контроллеры проанализируют входные данные, передадут их в модели (models), вставят результаты модели в представления(views), в итоге сгенерируют исходящие ответы.

Контроллеры состоят из действий, которые являются основными блоками, к которым может обращаться конечный пользователь и запрашивать исполнение того или иного функционала. В контроллере может быть одно или несколько действий.

## 4.1. Диаграмма базы данных «Конструктора расписания».

Целью построения диаграммы базы данных является получение графического представления физической структуры исследуемой предметной области.

Сущности описывают объекты, являющиеся предметом деятельности предметной области, и субъекты, осуществляющие деятельность в рамках предметной области. Свойства объектов и субъектов реального мира описываются с помощью атрибутов.

Взаимоотношения между сущностями иллюстрируются с помощью связей. Правила и ограничения взаимоотношений описываются с помощью свойств связей. Обычно связи определяют либо зависимости между сущностями, либо влияние одной сущности на другую.

В спроектированной базе данных, получились таблицы:

• Branches – в данной таблице находятся филиалы(территории) колледжей;

• Branch\_educationprogramm – эта таблица создана для связи таблиц Филиалы – образовательные программы;

• Cabinets – в этой таблице указываются кабинеты, находящиеся на конкретной территории;

• Disciplines – указываются дисциплины ( разделы МДК);

• Discipline\_educationprogramm – таблица, используемая для связи Образовательные программы – Дисциплины;

• Discipline\_traningperiod – здесь связываются таблицы Дисциплины – Учебные периоды;

• Educationprogramms – здесь перечислены образовательные программы учащихся ;

• Eo – таблица для внесения образовательных учреждений;

• Grups – здесь осуществляется внесение групп учащихся;

• Migration – для удобства разработки;

• Raspisanie – сводная таблица, в которой находятся данные из всех остальных таблиц.

• Raspisaniesettings – название таблицы с расписанием;

• Teachers – перечисление преподавателей, работающих в данном подразделении.

• Teacher\_discipline – таблица для связи Преподаватели – дисциплины;

• Traningperiods – перечисление учебных периодов;

• User – таблица возможных пользователей, имеющих вход в систему;

• Workmode – режим работы (звонки).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя** | **Тип** | **Ключ** | **Атрибуты** | **Дополнительно** | **Логическое имя** |
| **Таблица «Branches»** | | | | | |
| Id | Int(10) | PK | unsigned | Auto\_increment | идентификатор |
| FullName | Text |  |  |  | полное имя |
| ShortName | Varchar(255) |  |  |  | сокращенное имя |
| Address | Text |  |  |  | адрес |
| Eo | Int(10) |  | unsigned |  | образовательное учреждение |
| **Таблица «Branch\_educationprogramm»** | | | | | |
| Id | Int(10) | PK | unsigned | Auto\_increment | идентификатор |
| Branch\_Id | Int(10) |  | unsigned |  | внешний ключ для идентификации  филиала |
| Educationprogramm\_Id | Int(10) |  | unsigned |  | внешний ключ для идентификации образовательных программ |
| **Таблица «Cabinets»** | | | | | |
| Id | Int(10) | PK | unsigned | Auto\_increment | идентификатор |
| NumberCabinet | Varchar(255) |  |  |  | номер кабинета |
| NameCabinet | Text |  |  |  | название кабинета |
| **Таблица «Disciplines»** | | | | | |
| Id | Int(10) | PK | Unsigned | Auto\_increment | идентификатор |
| FullName | Text |  |  |  | полное название дисциплины |
| ShortName | Varchar(255) |  |  |  | сокращенное название дисциплины |
| CodeDisciplines | Varchar(255) |  |  |  | код дисциплины |
| CountClock | Int(11) |  |  |  | общее количество часов |
| LessonClock | Int(11) |  |  |  | количество часов лекционных |
| PrakticClock | Int(11) |  |  |  | количество часов учебных |
| **Таблица «Discipline\_educationprogramm»** | | | | | |
| Educationprogramm\_Id | Int(10) |  | Unsigned |  | внешний ключ для идентификации образовательных программ |
| Discipline\_Id | Int(10) |  | Unsigned |  | внешний ключ для идентификации дисциплин |
| **Таблица «Discipline\_traningperiod»** | | | | | |
| Discipline\_Id | Int(10) | PK | Unsigned |  |  |
| Trainingperiod\_Id | Int(10) | PK | Unsigned |  |  |
| Count\_clock | Int(11) |  |  |  | количество часов |
| **Таблица «Educationprogramms»** | | | | | |
| Id | Int(10) | PK | Unsigned | Auto\_increment | идентификатор |
| codeSpec | Varchar(255) |  |  |  | код специальности |
| FullNameSpec | Text |  |  |  | полное наименование специальности |
| ShortNameSpec | Varchar(255) |  |  |  | сокращенное наименование |
| CountMonth | Int(11) |  |  |  | количество в месяцах |
| **Таблица «EO»** | | | | | |
| Id | Int(10) | PK | unsigned | Auto\_increment | идентификатор |
| FullName | Text |  |  |  | полное название образовательного учреждения |
| ShortName | Varchar(255) |  |  |  | сокращенное название |
| Address | Text |  |  |  | адрес образовательного учреждения |
| **Таблица «Groups»** | | | | | |
| Id | Int(10) | PK | unsigned | Auto\_increment | идентификатор |
| FullName | Text |  |  |  | полное название группы |
| ShortName | Varchar(255) |  |  |  | сокращенное название группы |
| YearStart | Year(4) |  |  |  | год начала обучения |
| Branch\_educationprogramm\_Id | Int(10) |  |  |  | внешний ключ для идентификации таблицы «Филиалы – Образовательные программы» |
| **Таблица «Raspisanie»** | | | | | |
| Id | Int(10) | PK | unsigned | Auto\_increment | идентификатор |
| NumberToday | Int(11) |  |  |  |  |
| StatusRecord | Int(11) |  |  |  |  |
| Workmode\_Id | Int(10) |  | Unsigned |  | Внешший ключ для идентификации режима работы |
| Group\_Id | Int(10) |  | Unsigned |  | Внешний ключ для идентификации групп |
| Teacher\_Id | Int(10) |  | Unsigned |  | Внешний ключ для идентификации учителей |
| Settingraspis\_Id | Int(10) |  | Unsigned |  | Внешний ключ для идентификации названия расписания |
| Discipline\_Id | Int(10) |  | Unsigned |  | Внешний ключ для идентификации дисциплин |
| **Таблица «Raspisaniesettings»** | | | | | |
| Id | Int(10) | PK | Unsigned | Auto\_increment | идентификатор |
| NameRaspis | Varchar(255) |  |  |  | имя расписания |
| DateCreate | Datetime |  |  |  | дата создания |
| DateUpdate | Datetime |  |  |  | дата изменения |
| **Таблица «Teachers»** | | | | | |
| Id | Int(10) | PK | Unsigned | Auto\_increment | идентификатор |
| FirstName | Varchar(255) |  |  |  | фамилия |
| Name | Varchar(255) |  |  |  | имя |
| Lastname | Varchar(255) |  |  |  | отчество |
| Cabinet\_Id |  |  |  |  | внешний ключ для идентификации кабинетов |
| **Таблица «teacher\_discipline»** | | | | | |
| Teacher\_Id | Int(10) |  | unsigned |  | внешний ключ для идентификации учителей |
| Discipline\_Id | Int(10) |  | Unsigned |  | внешний ключ для идентификации дисциплин |
| **Таблица «Traningperiods»** | | | | | |
| Id | Int(10) | PK | unsigned | Auto\_increment | идентификатор |
| NamePeriod | Varchar(255) |  |  |  | название периода |
| NumberSemestr | Int(11) |  |  |  | номер семестра |
| Kurs | Int(11) |  |  |  | курс |
| DateStart | Date |  |  |  | дата начала |
| DateStop | Date |  |  |  | дата окончания |
| **Таблица «User»** | | | | | |
| Id | Int(11) |  |  | Auto\_increment | идентификатор |
| Created\_at | Int(11) |  |  |  | создать |
| Update\_at | Int(11) |  |  |  | изменить |
| Username | varchar(255) |  |  |  | имя пользователя |
| Auth\_key | Varchar(32) |  |  |  | аутентификационный ключ |
| Email\_confirm\_token | Varchar(255) |  |  |  |  |
| Password\_hash | Varchar(255) |  |  |  |  |
| Password\_reset\_token | Varchar(255) |  |  |  |  |
| Email | Varchar(255) |  |  |  | электронная почта |
| Status | Smallint(6) |  |  |  | статус |
| **Таблица «Workmode»** | | | | | |
| Id | Int(10) | PK | Unsigned | Auto\_increment | идентификатор |
| nameTimeInterval | Varchar(255) |  |  |  | название интервала |
| ShortName | Varchar(255) |  |  |  | сокращенное название |
| timeStart | Time |  |  |  | время начала |
| timeStop | Time |  |  |  | время окончания |
| Brunch\_Id | Int(10) |  | unsigned |  | внешний ключ для идентификации филиала |

На рисунке 5 представлена диаграмма базы данных информационной системы «Конструктор расписания».

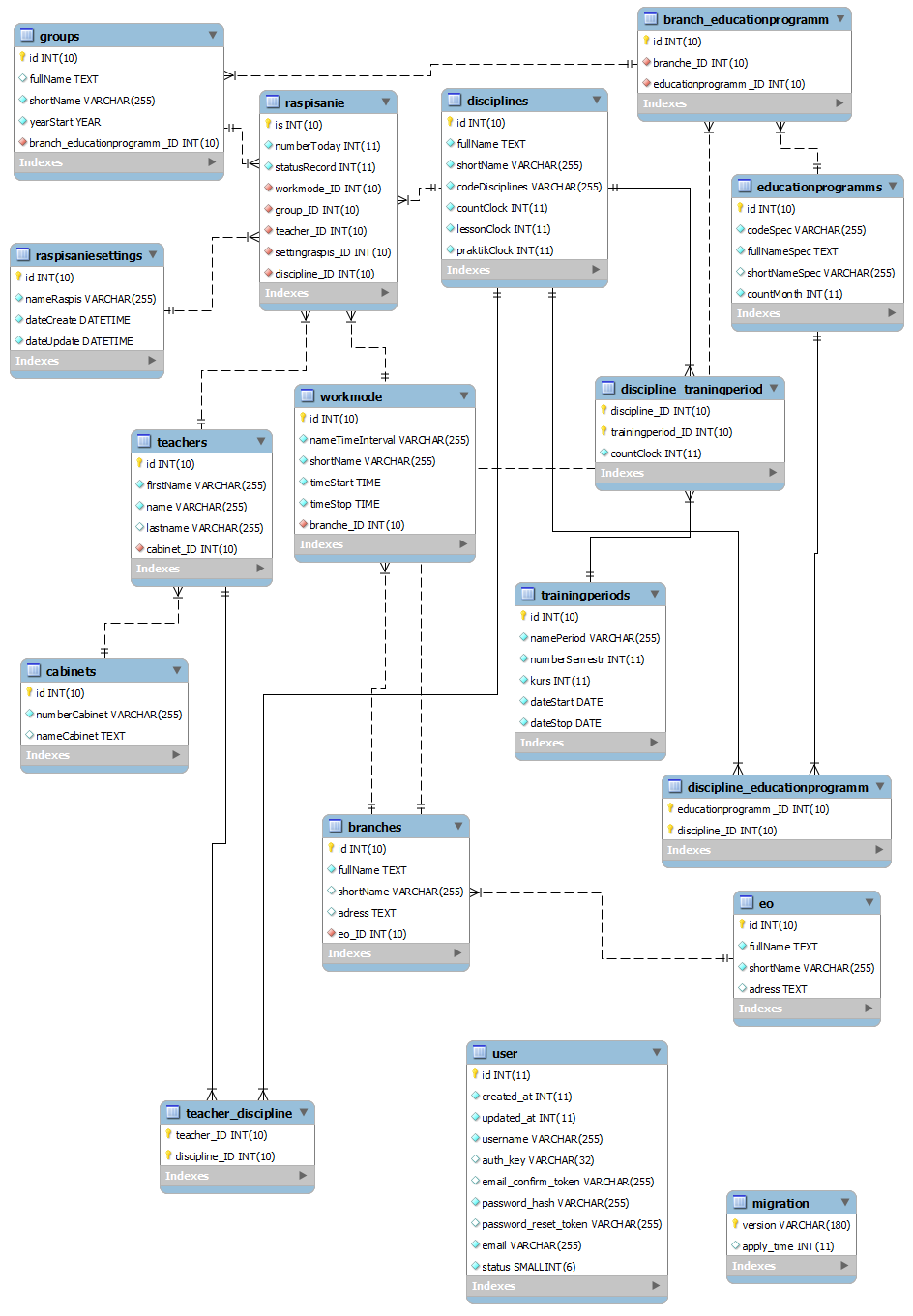


Рис. 5 Диаграмма базы данных

## 5. Реализация информационной системы

Работа программы начинается с окна авторизации пользователя, где пользователь вводит логин и пароль.

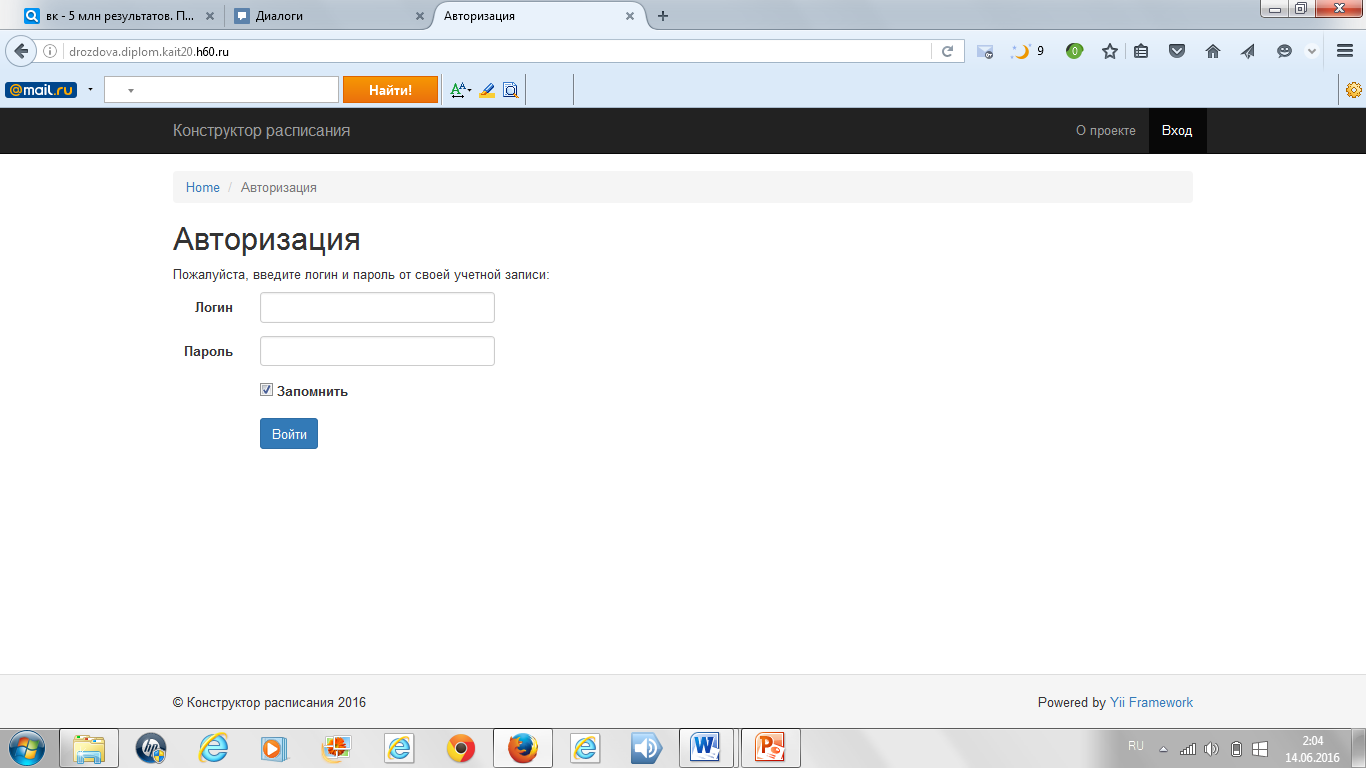


Рис.6 Форма авторизации пользователя.

После нажатие кнопки “Войти”, при указании правильного логина и пароля, открывается окно

После входа появляется страница с образовательными учреждениями.

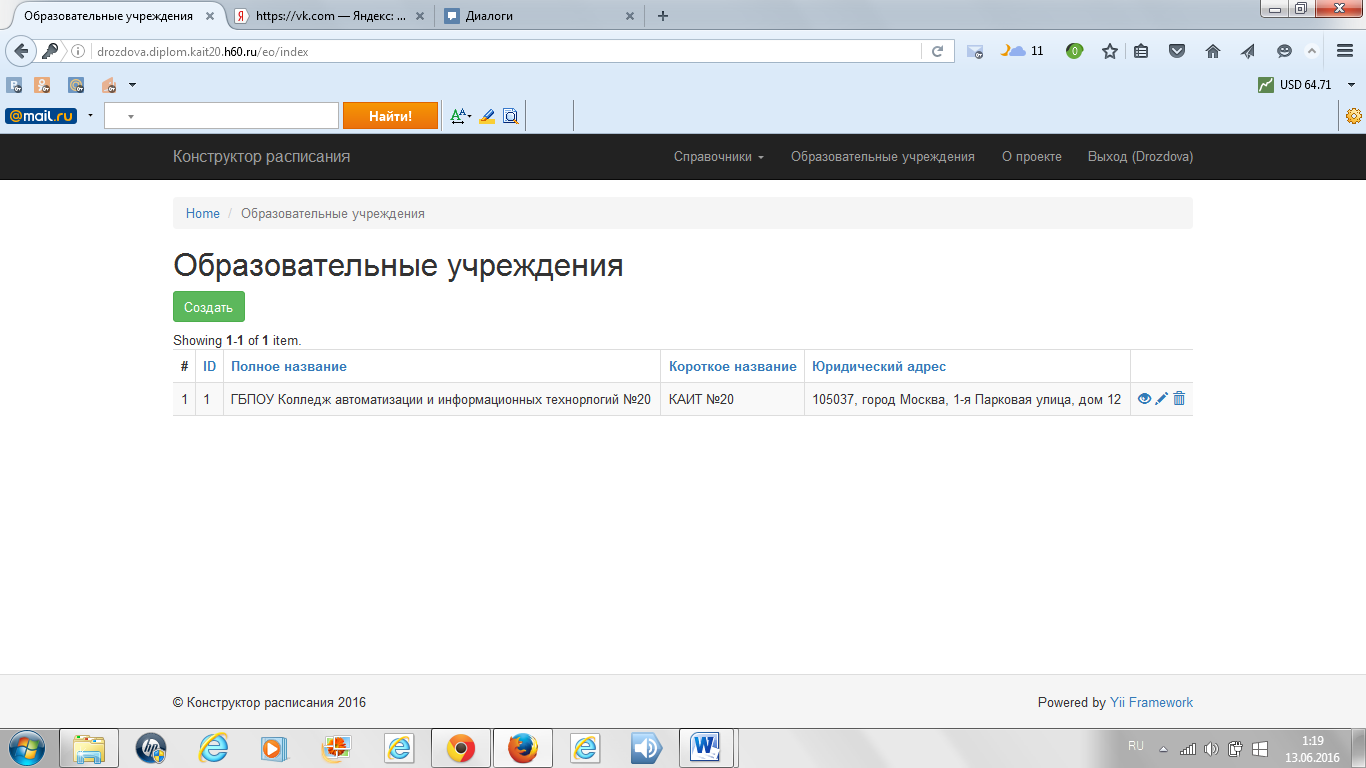


Рис.7 Окно сайта после авторизации.

Вверху находится панель управления системой (меню), которое имеет пункты: справочники, образовательные учреждения, о проекте, выход.

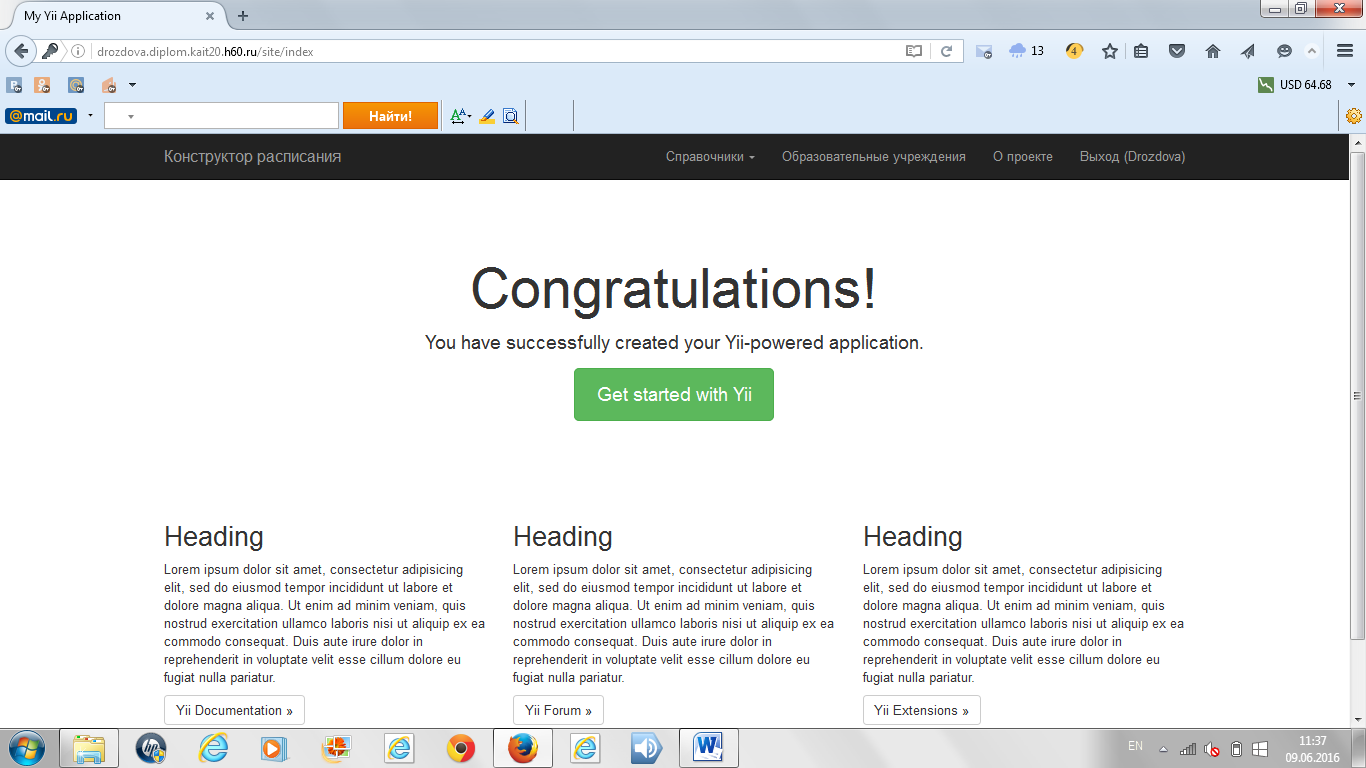


Рис.8 Меню сайта.

Для того, чтобы создать расписание для колледжа Администратор должен нажать на пункт «Образовательные учреждения», после чего открывается окно и есть кнопка «Создать».

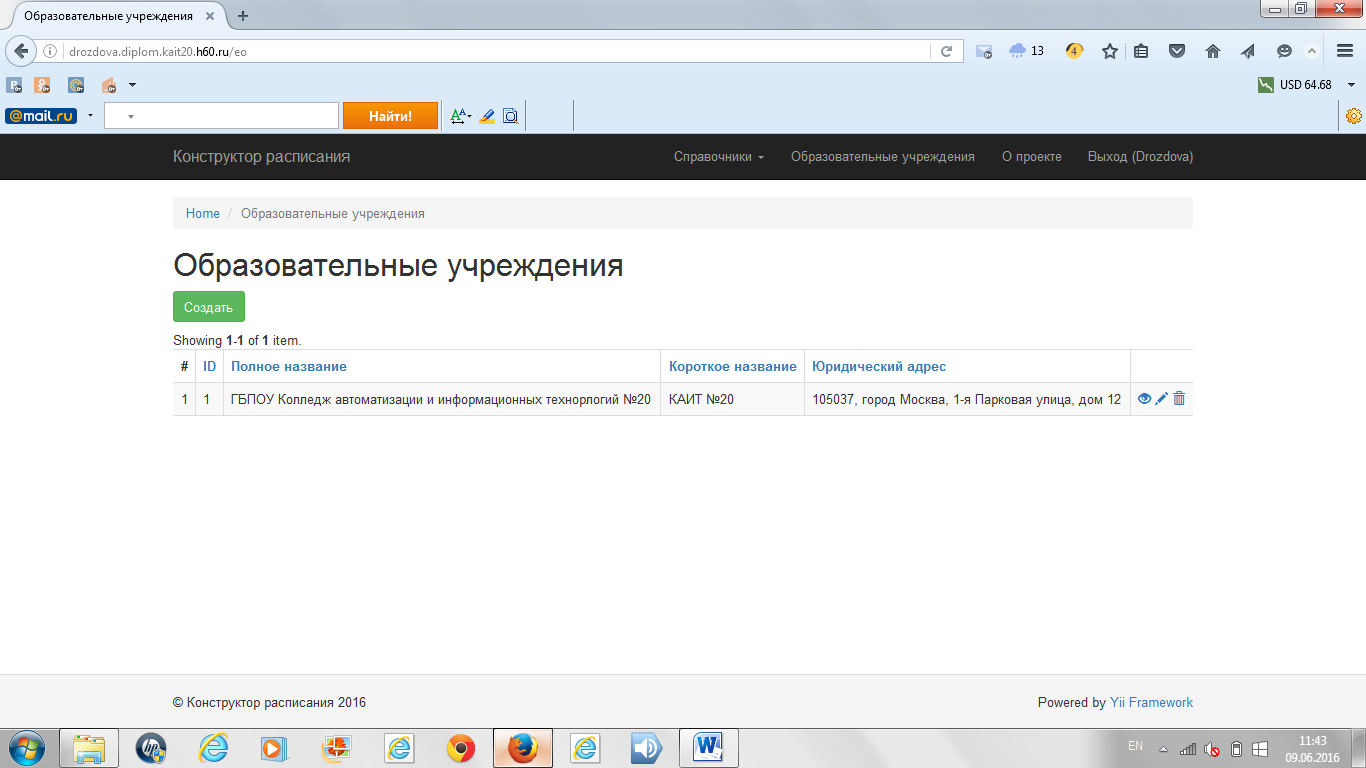


Рис.9 Создание Образовательного учреждения.

После нажатия кнопки «Создать» открывается окно с формой, которая имеет поля: полное название, краткое название, юридический адрес.

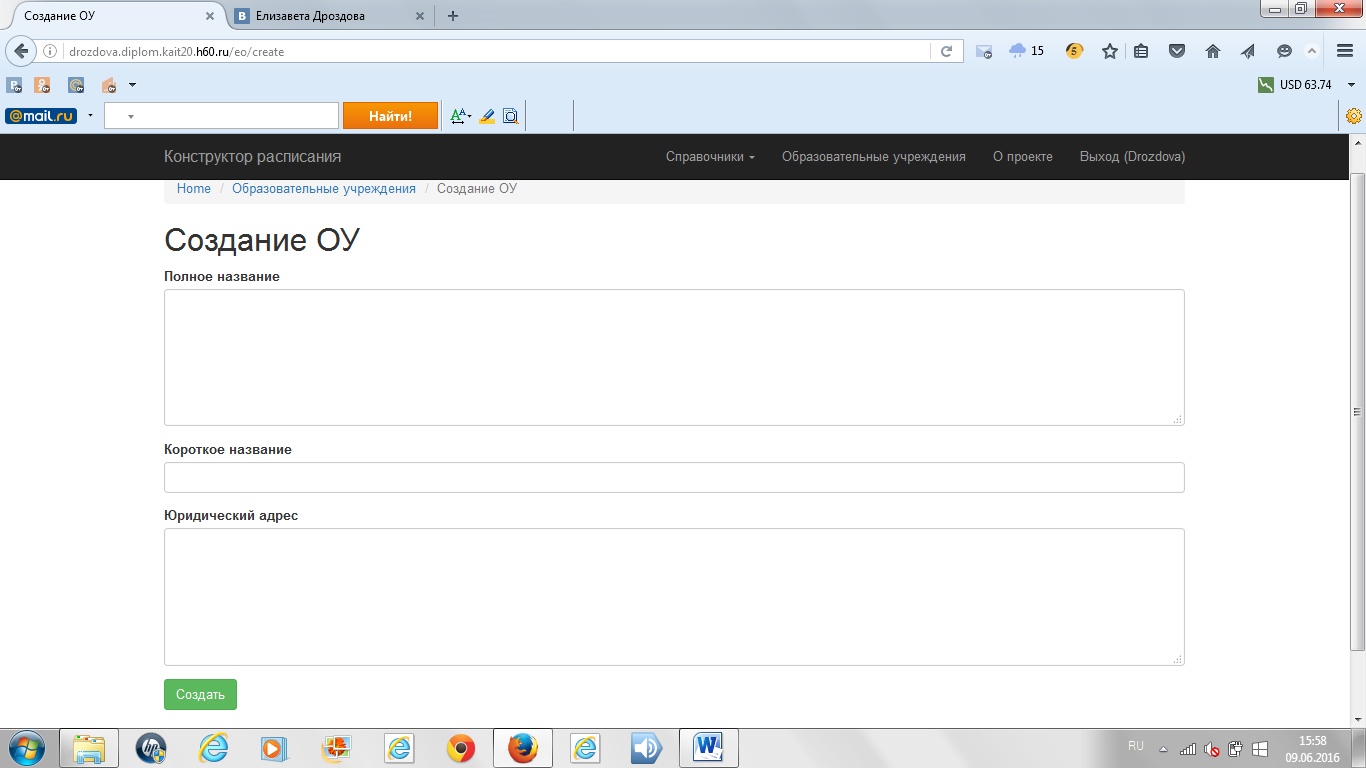


Рис.10 Форма для создания образовательного учреждения.

Чтобы все сохранилось, Администратор должен нажать кнопку «Создать». После этого появляется заполненная форма и две кнопки «Изменить» и «Удалить», также кнопка для создания филиалов.

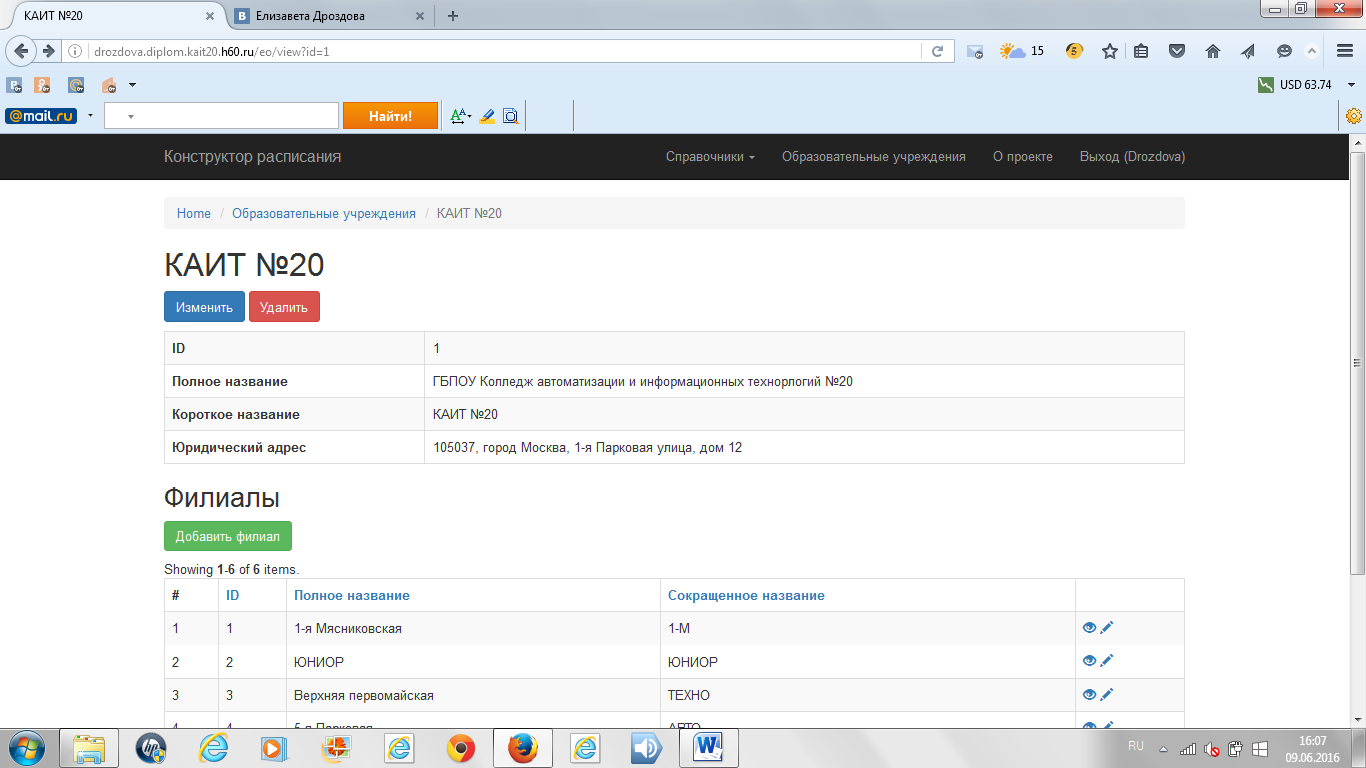


Рис.11 Просмотр учреждения.

Следующим шагом должно быть добавление филиалов (территорий) колледжа. При нажатии на кнопку «Добавить филиал» нужно заполнить поля: полное название, сокращенное название, адрес нахождения, а затем выбрать образовательные программы для данного филиала. После нажатия кнопки «Создать» появится таблица с заполненными филиалами.

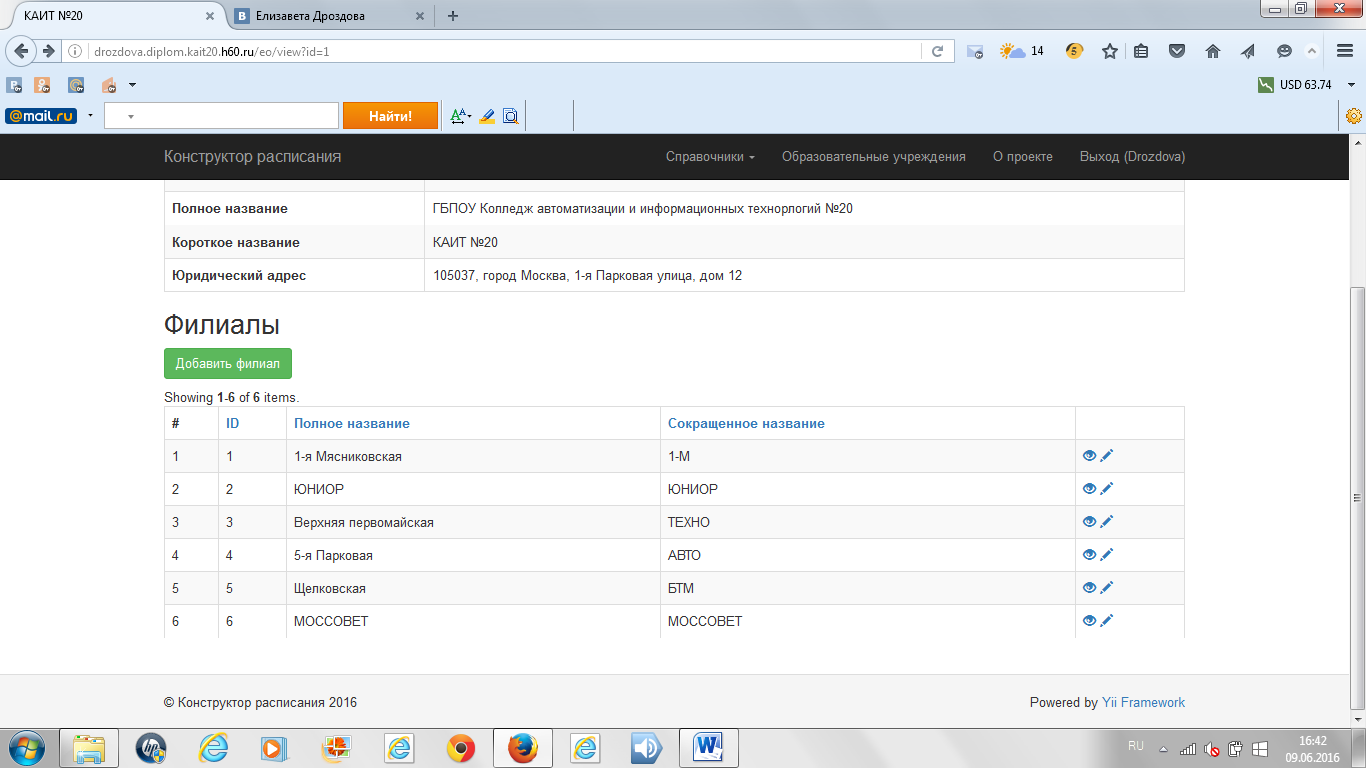


Рис.12 Просмотр филиалов.

Для того, что бы при создании филиала можно было выбрать образовательную программу нужно их заполнить. Это можно сделать, нажав на пункт «Справочники» и выбрать «Образовательные учреждения».

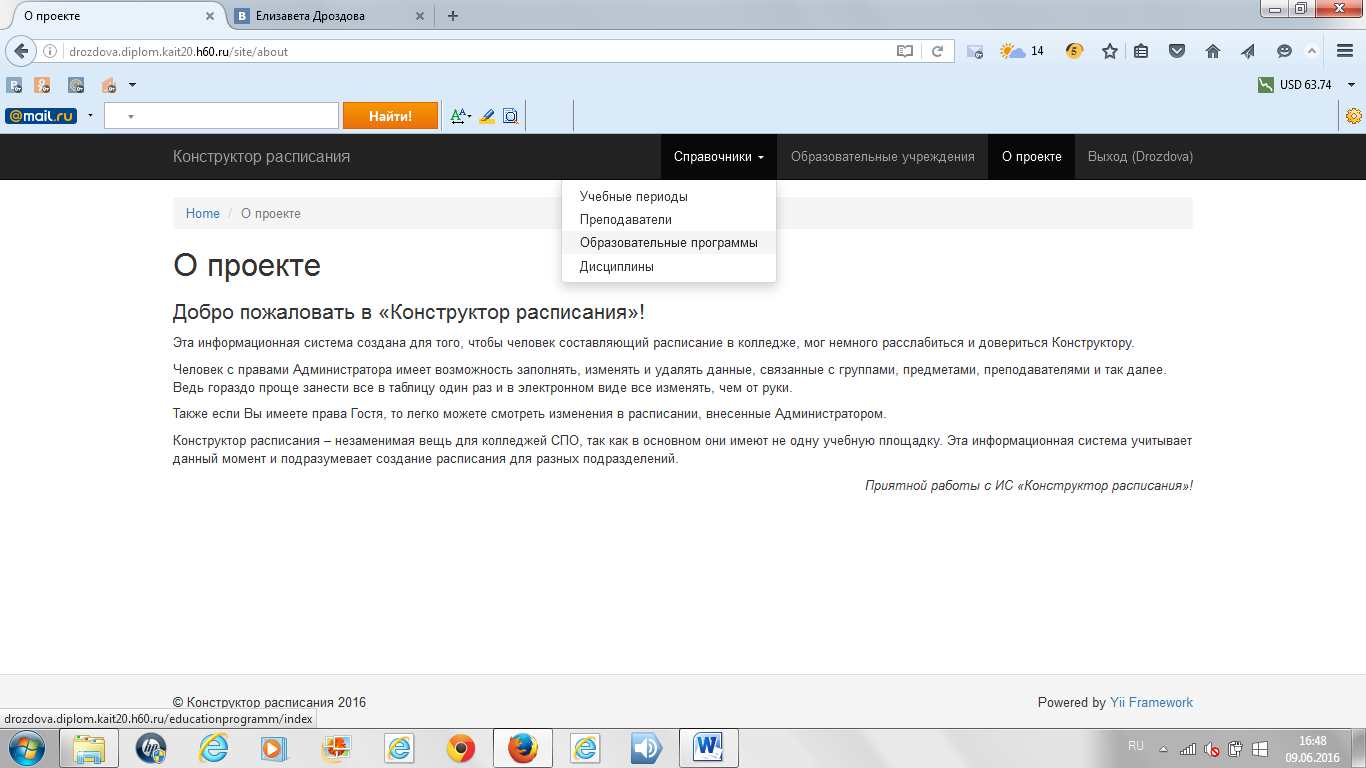


Рис.13 Справочники.

При нажатии кнопки «Добавить образовательную программу» открывается форма в которой нужно заполнить поля: код специальности, полное название специальности, короткое название специальности, срок обучения в месяцах.

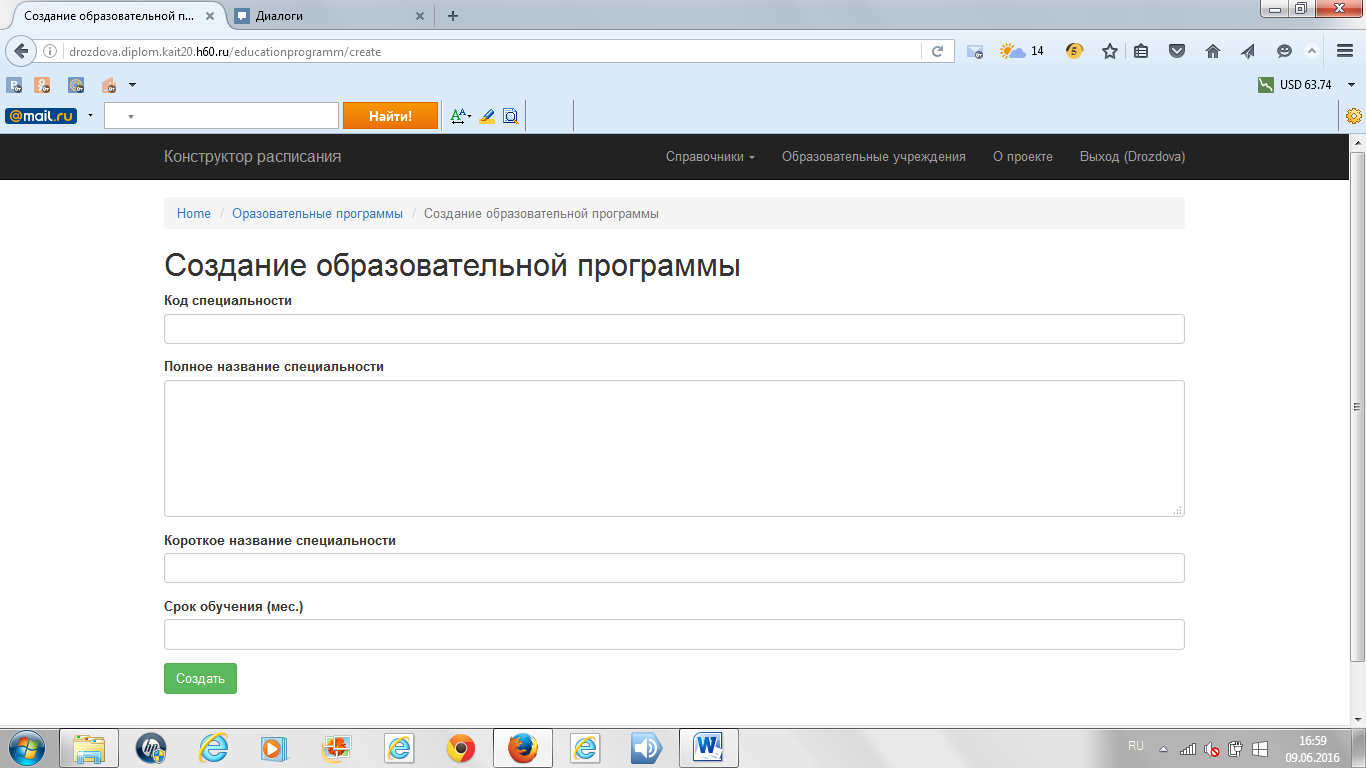


Рис.14 Создание образовательной программы.

После создания образовательных программ можно просматривать их в виде таблицы.

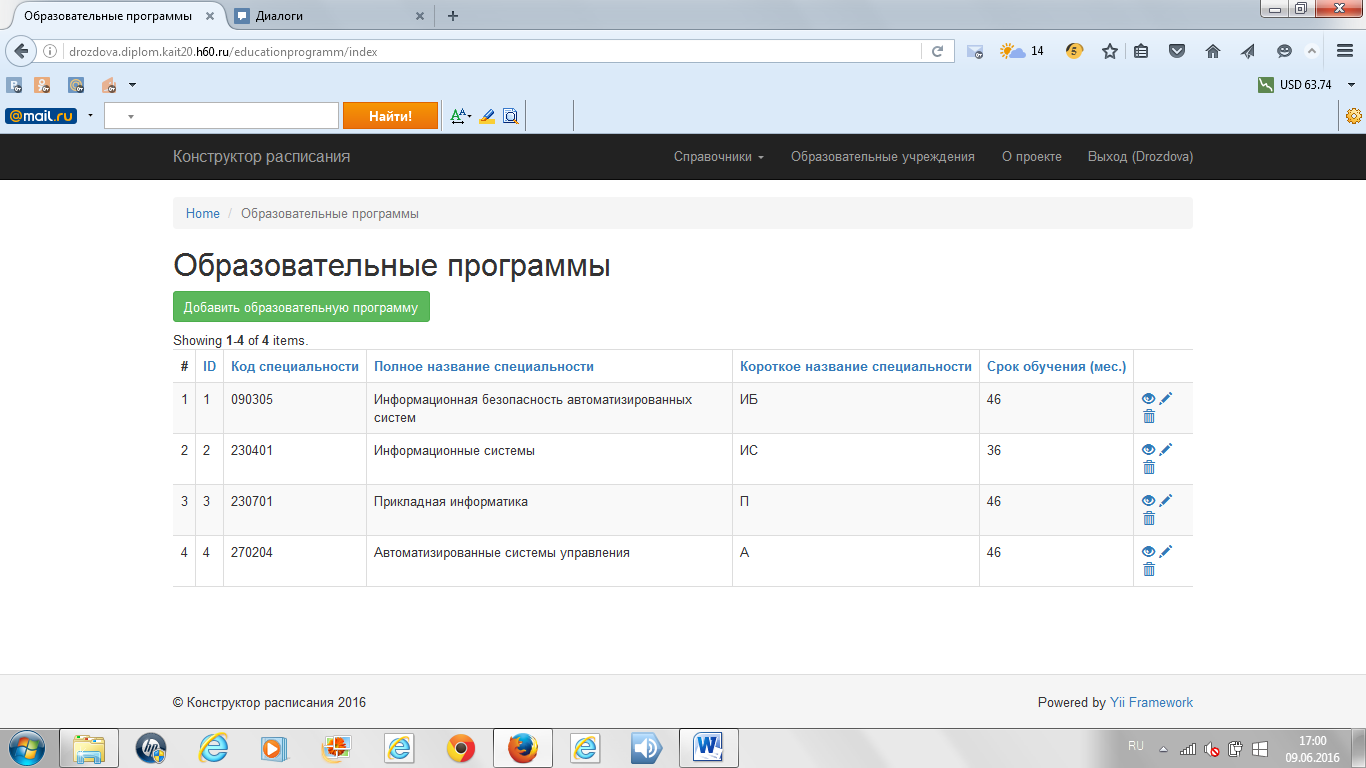


Рис.15 Просмотр образовательных программ.

Для дальнейшей работы по созданию расписания нужно заполнить остальные данные в пункте меню «Справочники».

Администратор выбирает пункт «Учебные периоды», в открывшемся окне нажимает кнопку «Добавить учебный период», далее открывается форма с полями: название учебного периода, номер семестра, курс. После нажимает кнопку «Создать». Когда все периоды (семестры) заполнены, можно просмотреть их в виде таблицы.

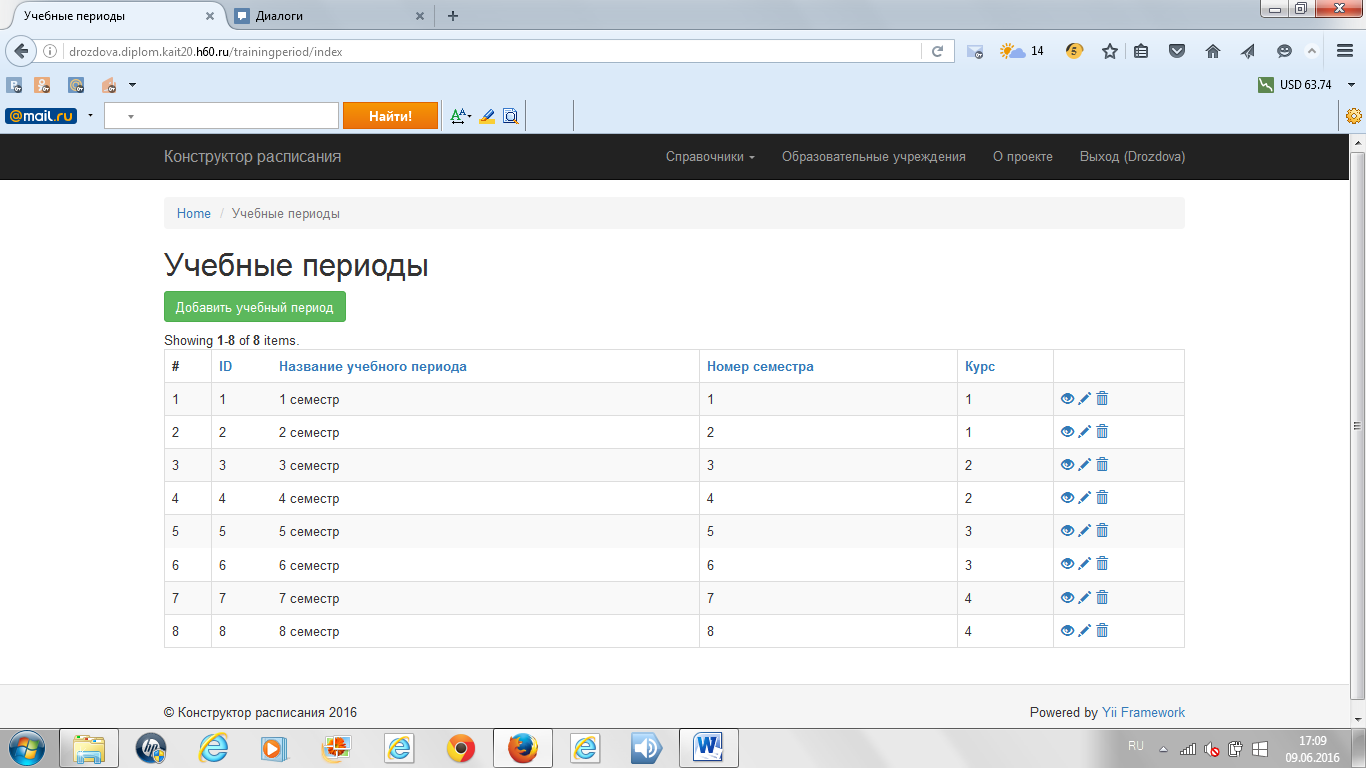


Рис.16 Просмотр учебных периодов.

Заполнение пункта «Преподаватели» происходит следующим образом: Администратор нажимает кнопку «Добавить преподавателя», затем заполняет Ф.И.О., нажимает кнопку «Создать». По итогу можно просматривать данные о преподавателях в виде таблицы.

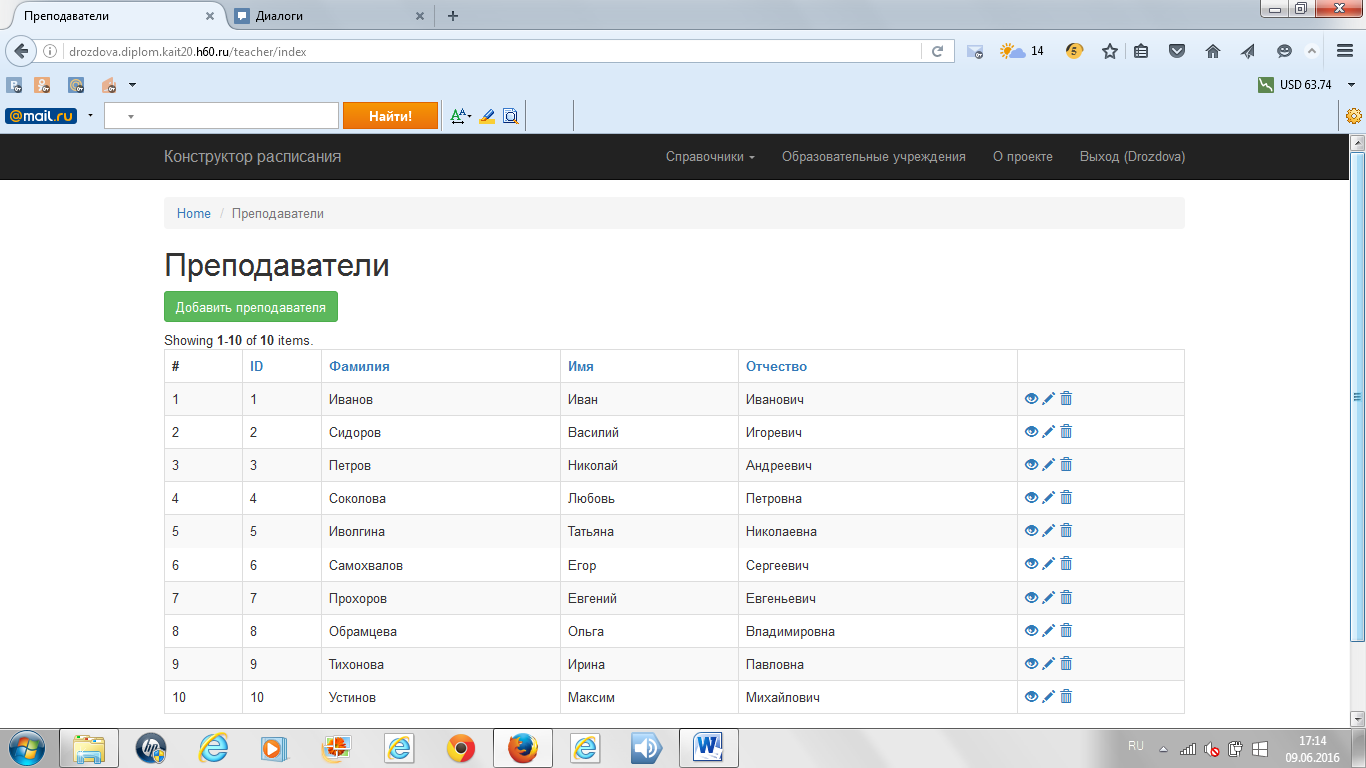


Рис.17 Просмотр существующих преподавателей.

Нажав пункт «Дисциплины» и кнопку «Создать дисциплину», Администратор заполняет, выбирает из списка семестр (к которому относится дисциплина), заполняет полное название, сокращенное название, код по учебному плану, общее количество часов, часы лекций, часы практики, выбирает из списка образовательные программы и преподавателя. Далее все дисциплины можно просмотреть в виде таблицы.

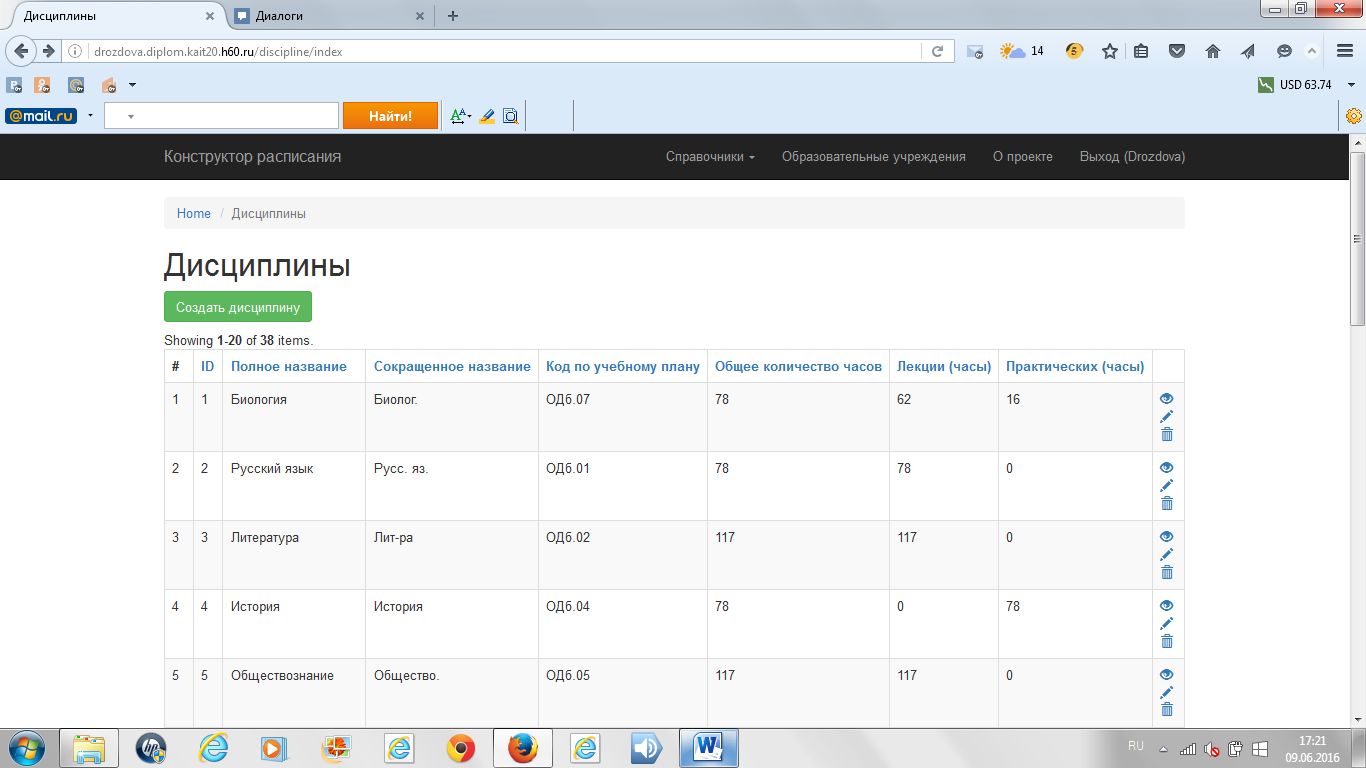


Рис.18 Просмотр дисциплин.

Теперь Администратор выбирает в меню «Образовательные учреждения» и там нажимает на кнопку просмотра учреждения, которая изображена в виде глаза.

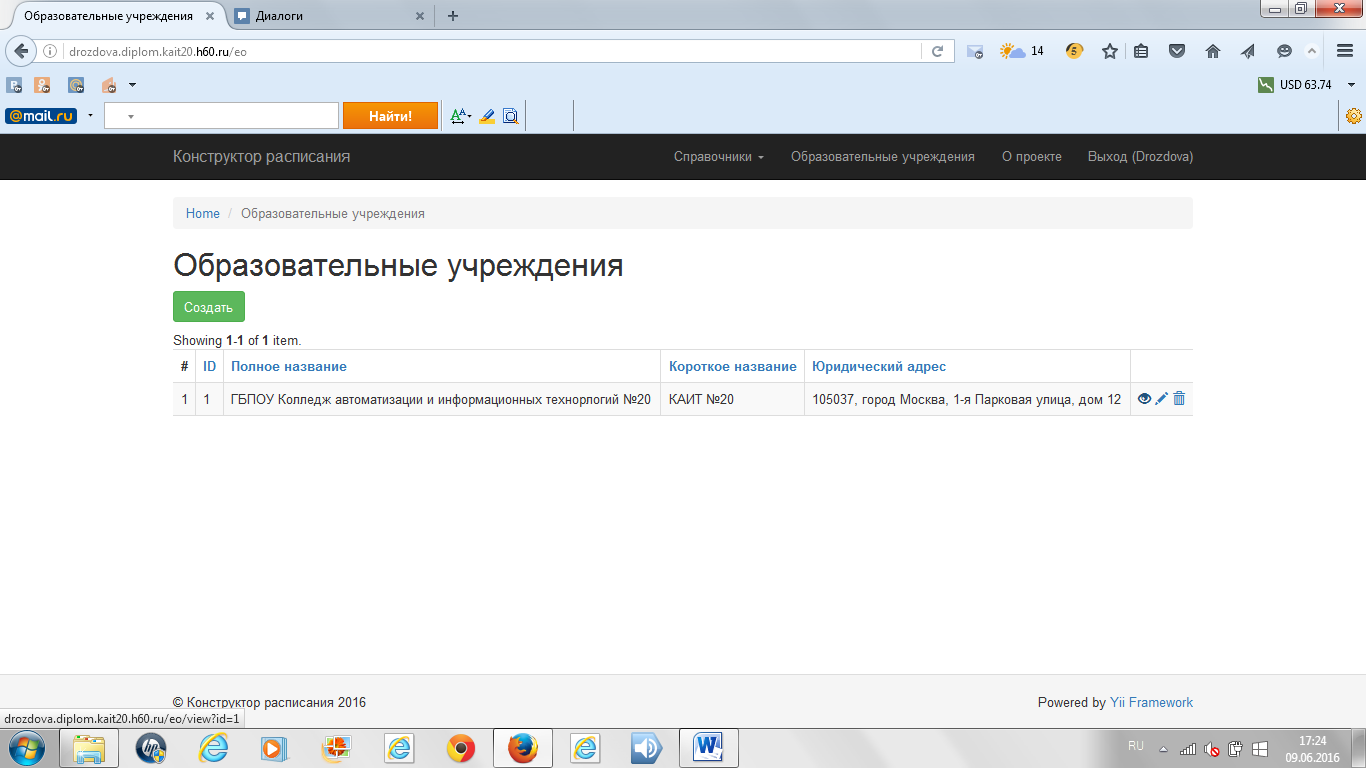
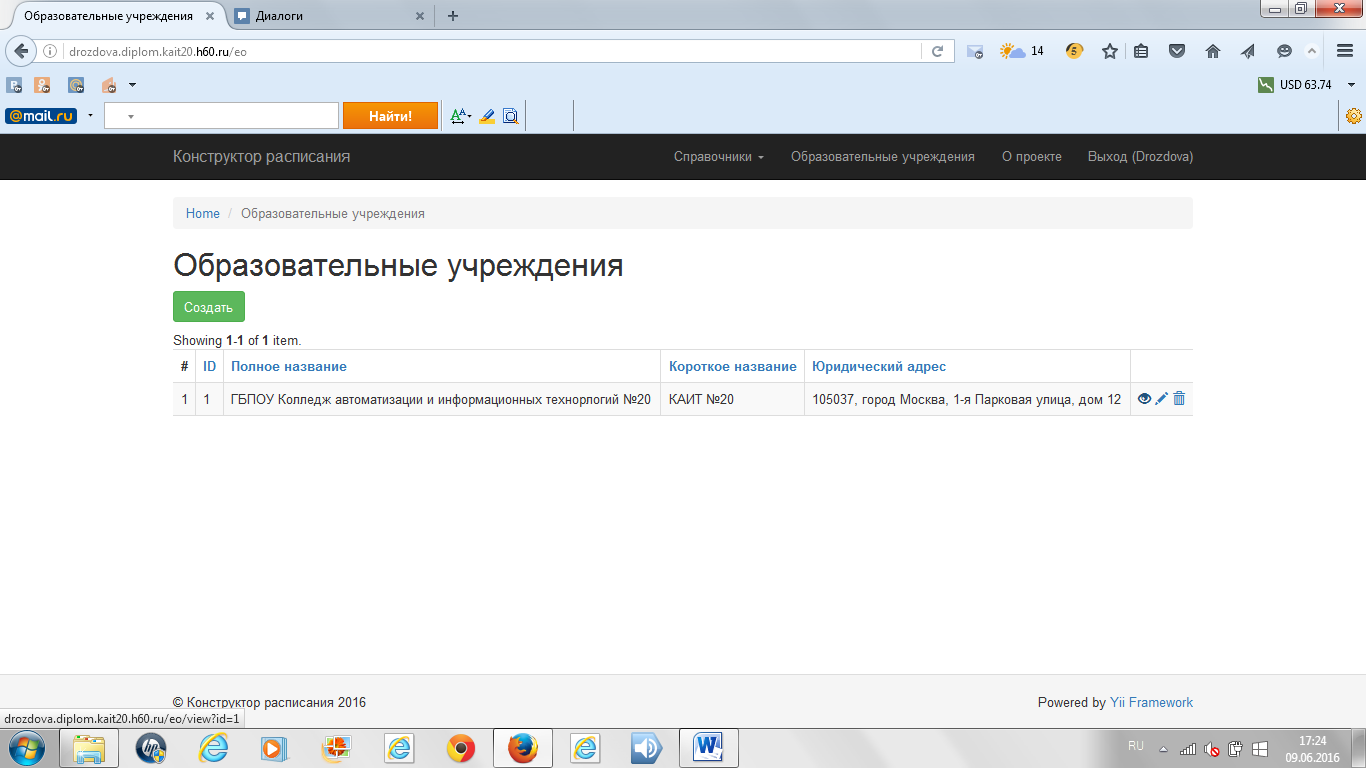


Рис.19 Обозначение просмотра страниц.

Там выбрать тот филиал, для которого создается расписание, и нажать на кнопку в виде глаза.

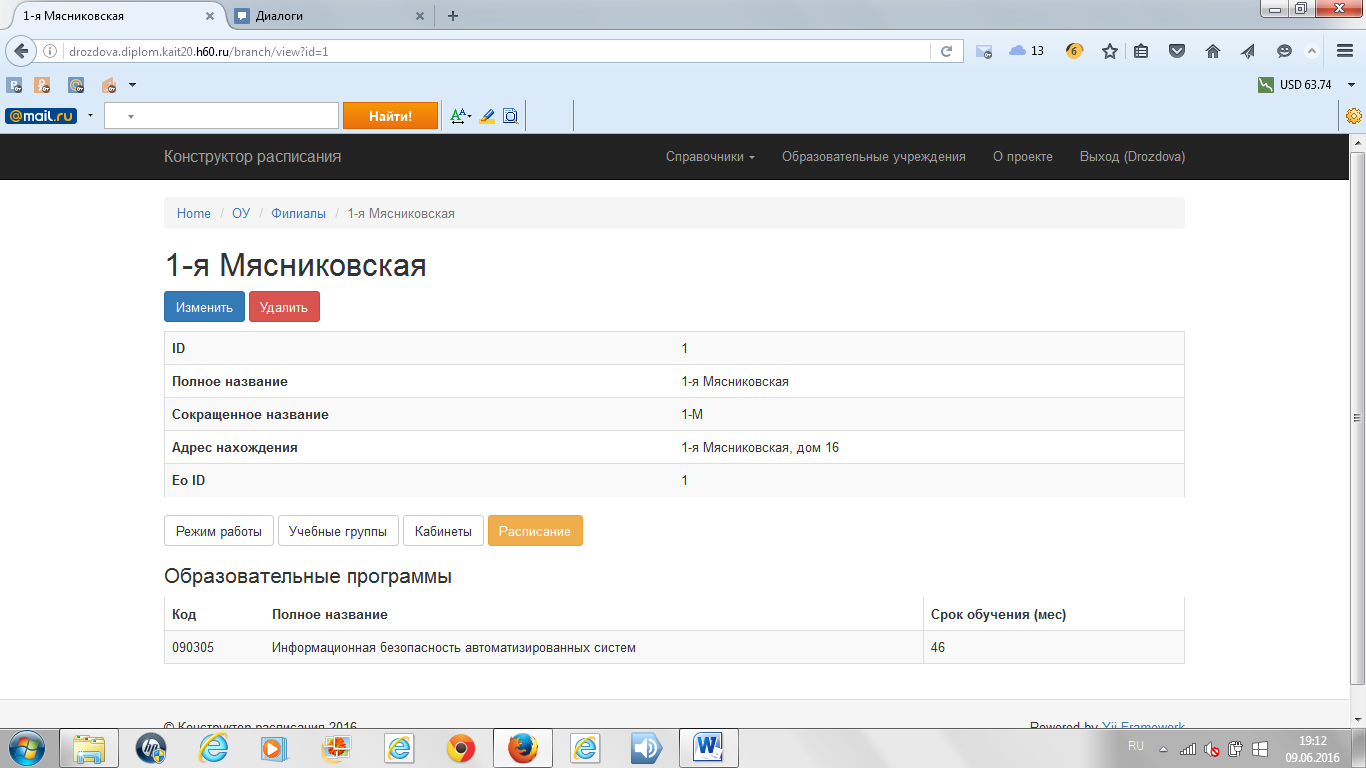


Рис.20 Выбор филиала.

В открывшимся окне, Администратор видит данные о филиале, образовательные программы и 4 кнопки: «Режим работы», «Учебные группы», «Кабинеты», «Расписание».

Для того, чтобы получить в итоге расписание нужно заполнить: «Режим работы», «Учебные группы», «Кабинеты».

При создании режима работы заполняются поля: название временного интервала, сокращенное название, время начала и время окончания.

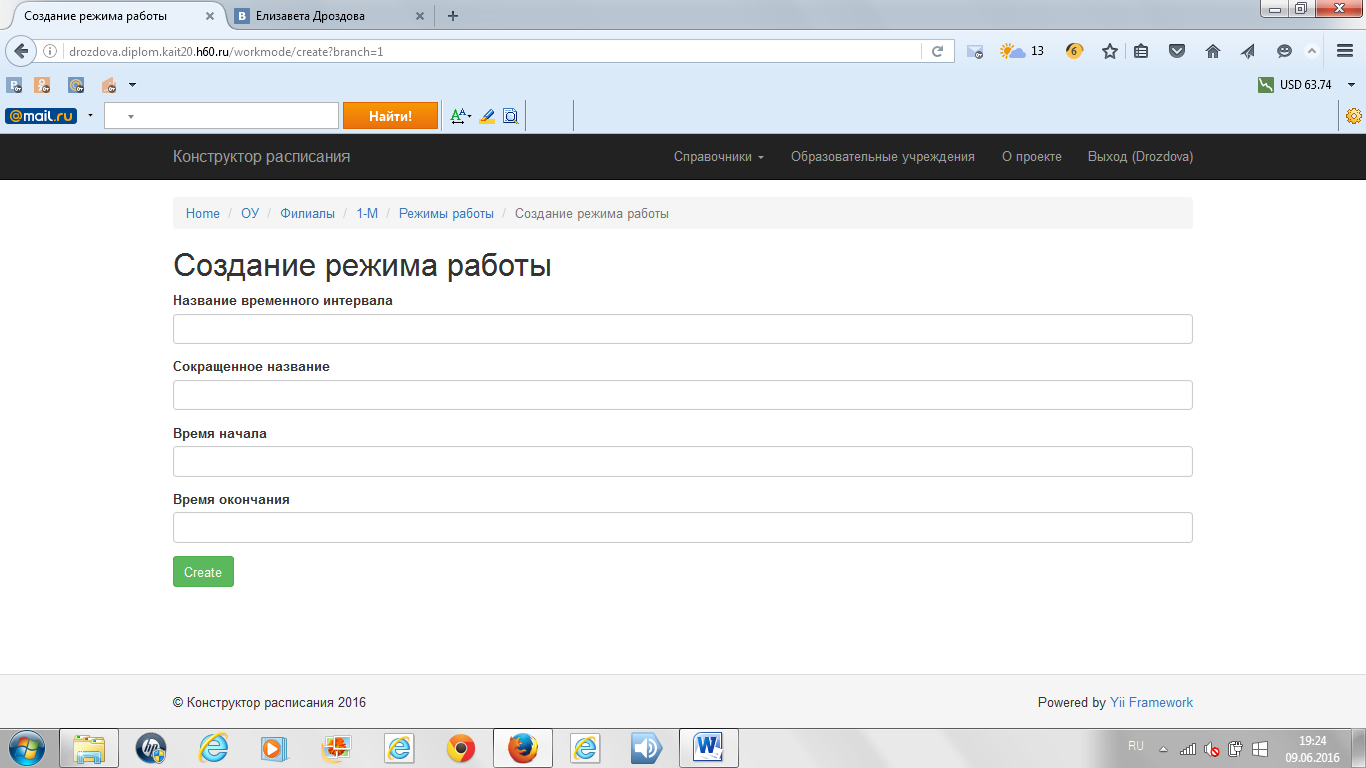


Рис.21 Создание режима работы.

При создании учебных групп заполняются поля: полное название группы, сокращенное название, год начала обучения, выбор образовательной программы.

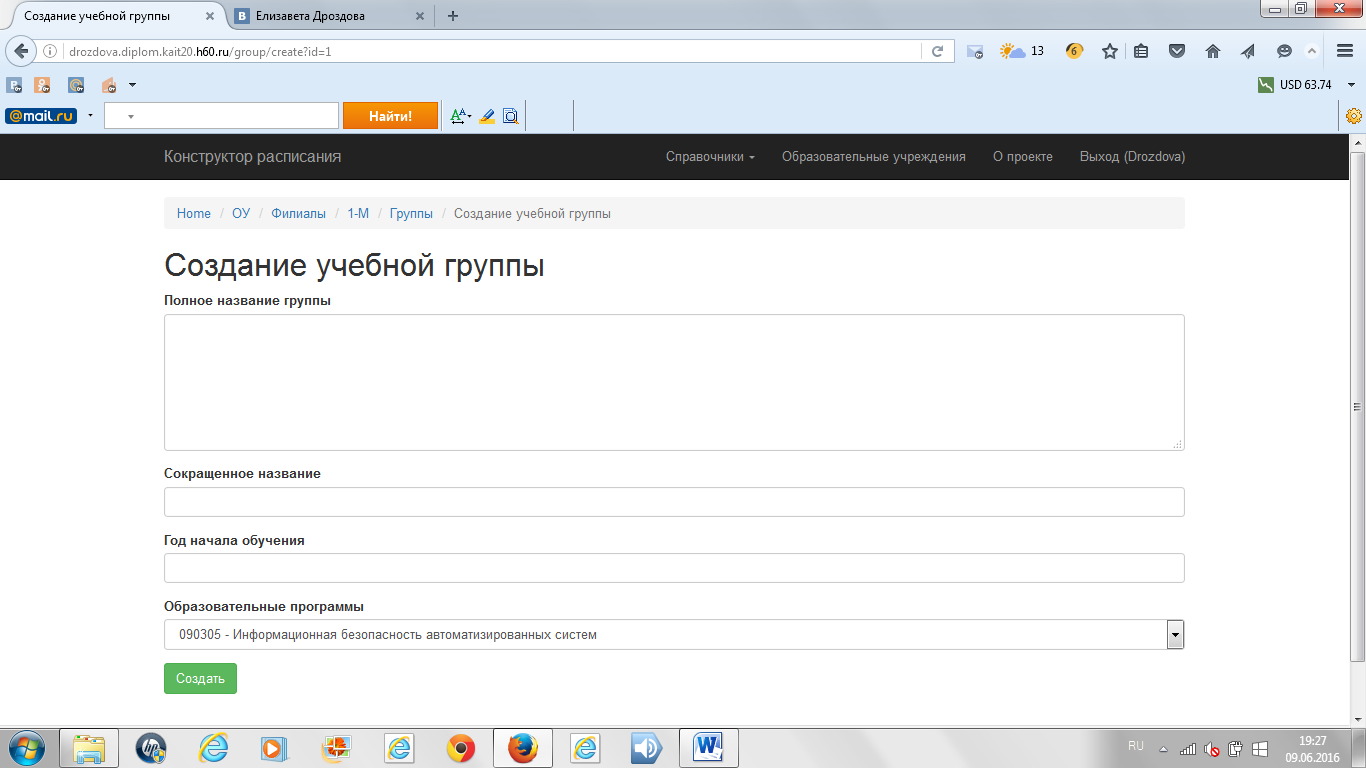


Рис.22 Форма создания учебной группы.

При создании кабинетов заполняются поля: номер кабинета, название кабинета.

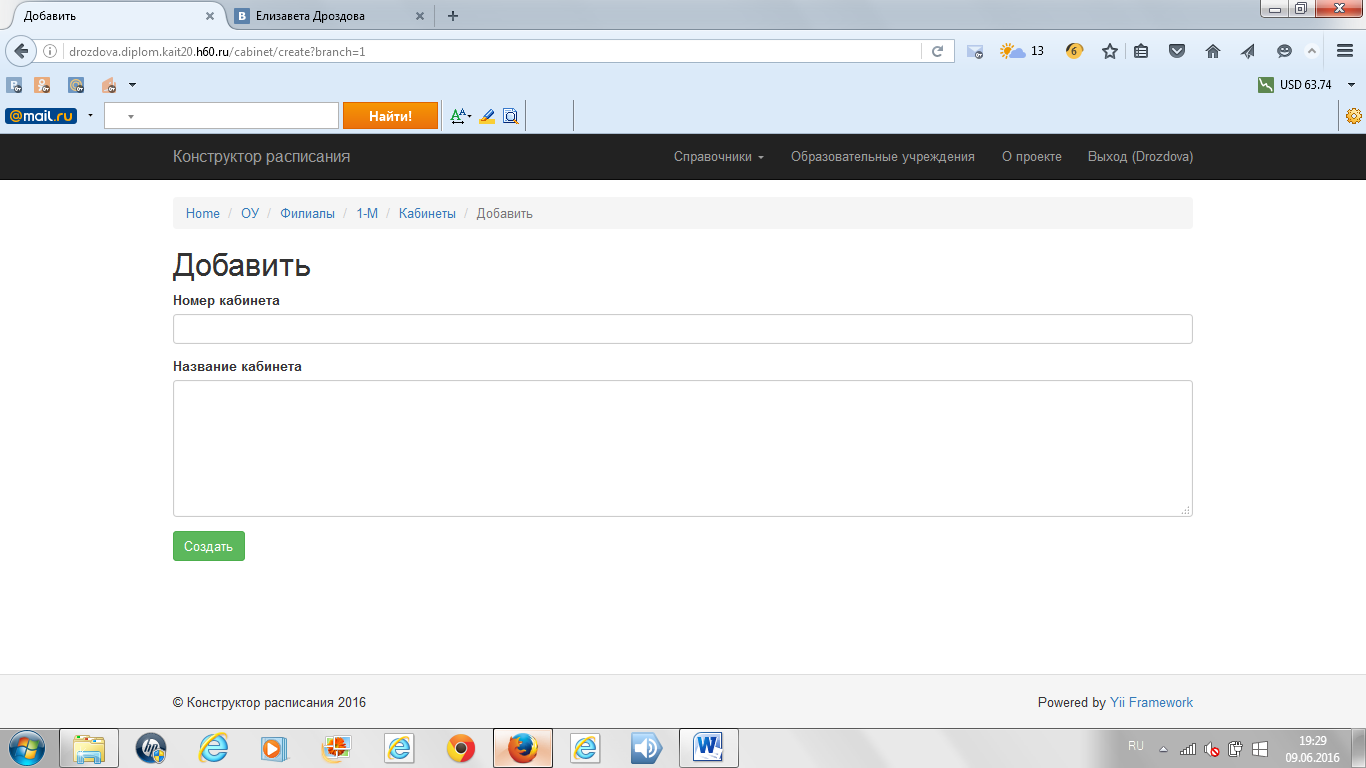


Рис.23 Добавление нового кабинета.

После того, как заполнены все формы, в том порядке, который указан можно нажимать на кнопку «Расписание», там нужно задать ему название, а далее в режиме просмотра нажать кнопку «Открыть».



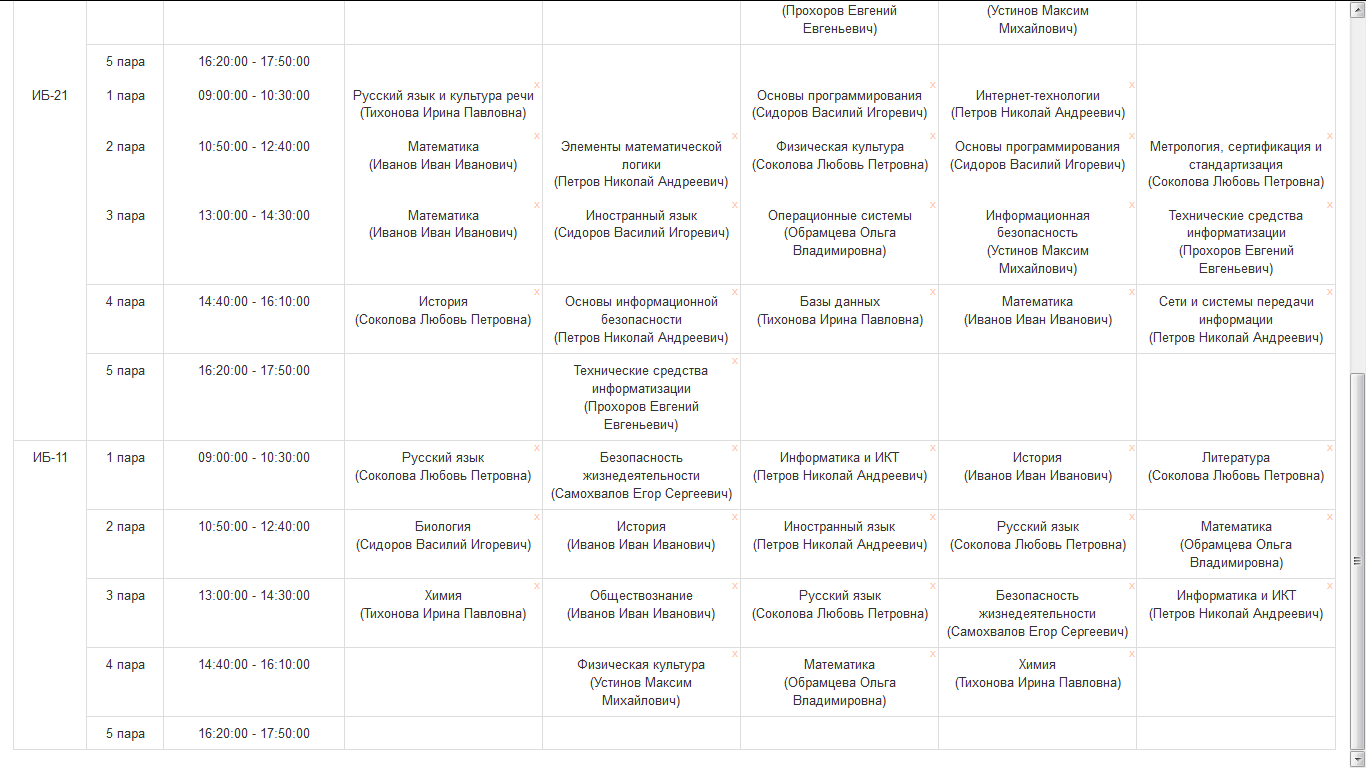


Рис.24-25 Сетка с расписанием.

Для того чтобы сформировать сетку с расписанием нужно нажать на пустую ячейку, после чего появится модальное окно со списком дисциплин для определенной специальности. Список зависит и меняется в зависимости от уже выбранных.

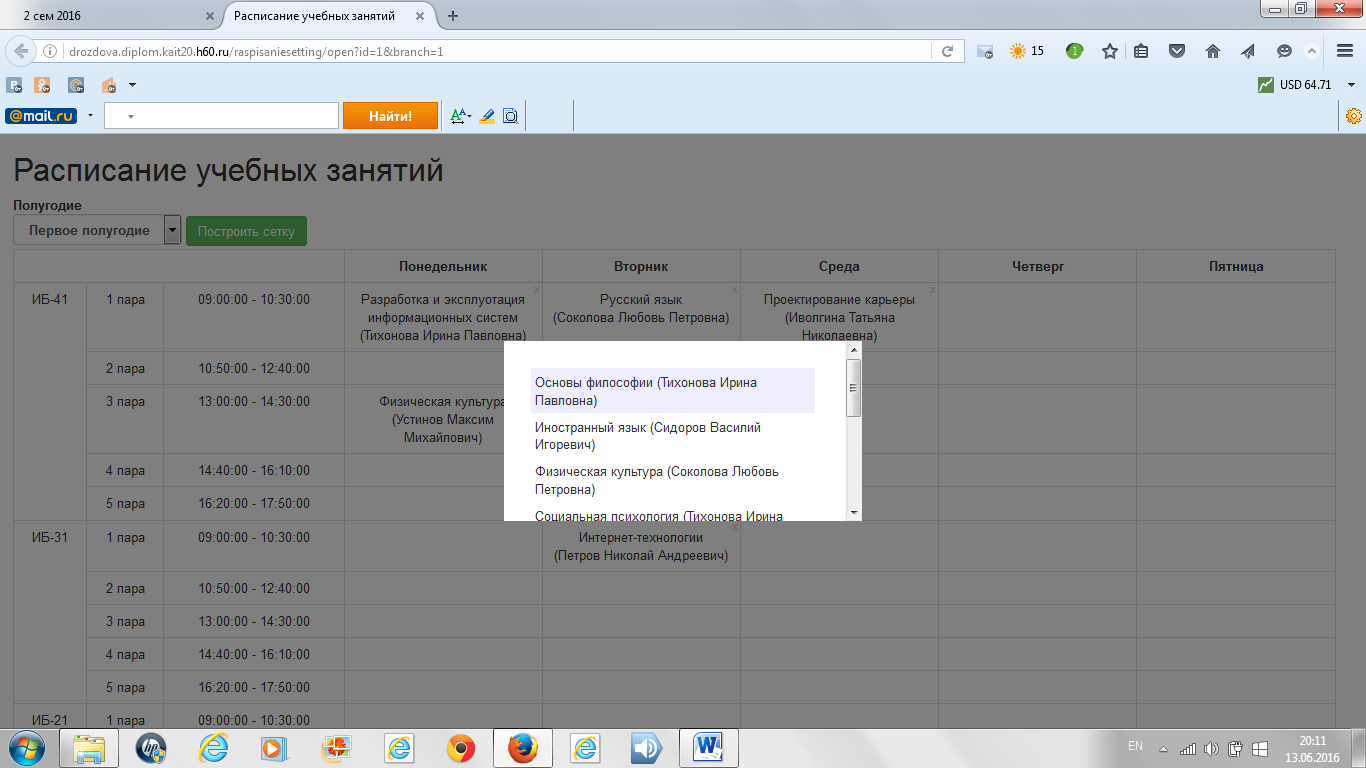


Рис.26 Модальное окно.

После нажатия на дисциплину, она отображается в сетке и отправляется в базу данных. Для того чтобы удалить ее достаточно нажать на маленький красный крестик.

## Заключение

В результате курсового проекта была спроектирована и реализована информационная система «Конструктор расписания учебных занятий для образовательных учреждений среднего профессионального образования». Данная система удовлетворяет всем требованиям, предъявленным в задании, и реализует большинство необходимых функций. Приложение выполняет следующий перечень требований:

* представление данных в удобном для пользователя виде;
* добавление, удаление и изменение данных;
* просмотр даты и времени, и т.д.

Данная программа разработана и предназначена для автоматизации рабочего места методиста, обеспечивающая предоставление и хранение полной информации о расписании и его составляющих.

Для того чтобы программа функционировала, необходима операционная система Windows. Методист должен иметь навыки работы с ПК, и знать основы OC Windows.

Программа включает в себя интуитивно понятный и практичный пользовательский интерфейс, подходящий под OC Windows. Для работы с программой пользователю необходимо владеть элементарными навыками работы с данной оболочкой.

# Литература

1. Описание информационной системы: [Электронный ресурс]// Википедия. URL: https://ru.wikipedia.org/wiki/Информационная\_система.
2. Описание языка программирования: [Электронный ресурс]//URL: http://www.yiiframework.com/ - официальный сайт фреймворка.
3. Описание языка программирования: [Электронный ресурс]//URL: http://yiiframework.ru/ - yii2 по русски.
4. Описание языка программирования: [Электронный ресурс]//URL: http://www.php.su/ - о языке программирования PHP (включая учебники и самоучители)