

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа №64 г. Владивостока»

Разработка программного модуля интеграции ГИС «РО» (государственной
информационной системы «Региональное образование»)
и информационного портала образовательной организации

Автор работы:
ученик 11 «а» класса Руденко В.О.
Научный руководитель:
учитель информатики МБОУ «СОШ
№64» Руденко Е.О.

г. Владивосток
2023

Оглавление

Введение	3
1. Теоретическая часть.....	5
1.1 Технологии, используемые при разработке веб-приложений.....	6
1.2 Языки описания содержимого Web-страниц.	9
1.3 Протоколы, на базе которых происходит обмен сообщениями в Web-приложениях.	10
1.4 Использование библиотек и фреймворков в процессе разработки программного обеспечения	10
2. Практическая часть	12
2.1. Разработка модели информационной системы.....	12
2.2. Разработка модели базы данных ИС и анализ пакета расписания «msxml»	13
2.3. Разработка программного компонента импорта расписания в информационную систему.	14
2.4. Разработка web-приложения с функцией выбора способа отображения расписания.	17
Заключение	21
Список литературы	22

Введение

Формирование цифровой образовательной среды в образовательной организации насущная необходимость, поскольку школа несет особую миссию, которая заключается в подготовке всесторонне развитого выпускника, обладающего необходимым набором компетенций и компетентностей, готового к продолжению образования в высокоразвитом информационном обществе.

Цифровая образовательная среда образовательной организации предполагает набор ИКТ-инструментов, использование которых должно носить системный порядок и удовлетворяет требованиям ФГОС к формированию условий реализации основной образовательной программы начального общего, основного общего и среднего общего образования, способствует достижению обучающимися планируемых личностных, метапредметных, предметных результатов обучения [1].

Кроме того, цифровая образовательная среда образовательной организации должна стать единым пространством коммуникации для всех участников образовательных отношений, действенным инструментом управления качеством реализации образовательных программ, работой педагогического коллектива.

Доступ к актуальной информации о расписании уроков позволяет обучающимся правильно планировать учебный процесс, вместе с тем, функциональные ограничения программного продукта «Сетевой город/регион. Образование» не позволяют обучающимся видеть расписание других классов или учителя.

Целью настоящего проекта является разработка комплекса программных продуктов, позволяющих организовать доступ к информации о расписании уроков на сайте информационного портала образовательной

организации с применение web-технологии, с возможностью выбора формата отображения расписания и функцией быстрого импорта расписания из ГИС «РО» (государственной информационной системы «Региональное образование»), работающей на базе продуктов компании ООО «ИРТЕХ», семейства «Сетевой город/регион. Образование».

Задачи:

1. изучить основные теоретические и практические аспекты предметной области.

2. осуществить разработку модель функционирования информационной системы и информационных каналов именованной информацией (поставки информации) между ее компонентами.

3. разработать web-портал (страницу web-портала) для публикации открытой информации о расписании уроков с функцией выбора вида отображения расписания;

4. разработать программный компонент, реализующий функционал импорта расписания в информационную систему, использующий формат экспорта/импорта данных, совместимый с ГИС «РО».

1. Теоретическая часть

Программное обеспечение в современном мире становится все сложнее, поскольку на него возлагается решение все более сложных задач. Коммерческие компании и государственные организации стремятся автоматизировать все больше своих процессов, как внутренних, так и тех, что связаны с общением с внешним миром. Необходимые при этом программные системы обычно должны поддерживать одновременную работу многих географически распределенных пользователей с централизованными и интегрированными хранилищами данных организаций. При этом должны обеспечиваться эффективность обработки запросов, высокая надежность работы, предоставление результатов в удобном пользователям виде, разграничение областей доступа и разрешенных операций для разных групп пользователей и т.д. Разработка таких приложений (в тех случаях, когда подходящей системы нет на рынке готовых продуктов), их внедрение и поддержка становятся все дороже из-за возрастающего количества предоставляемых ими функций, ограничений, которым они должны удовлетворять, составляющих их компонентов, возможных связей и взаимодействий между ними.

Есть, тем не менее, факторы, которые помогают значительно снизить расходы на создание подобных систем, один из них — широчайшее распространение Интернет. Если программное обеспечение использует для связи между своими элементами базовые протоколы Интернет (TCP/IP и HTTP) и предоставляет пользовательский интерфейс в виде страничек HTML, которые можно просматривать в любом браузере, то практически каждый его потенциальный пользователь не имеет технических препятствий для обращения к этому ПО. Не нужно прокладывать сети,

тратить усилия на настройку связи с серверами, разрабатывать различные клиентские компоненты для разных операционных систем, устанавливать клиентам специальное оборудование и программные компоненты, и т.д. Интернет предоставляет готовую инфраструктуру для создания крупномасштабных программных систем, в рамках которых десятки тысяч компонентов работают совместно и миллионы людей пользуются предоставляемыми ими услугами.

Таким образом, возник особый класс программных систем — Web-приложения. Это программные системы, использующие для связи протоколы Интернет, а в качестве пользовательского интерфейса — HTML-страницы. Сейчас они стали одним из самых востребованных видов ПО. Однако, чтобы сделать потенциальные выгоды от использования Интернет реальными, необходимы технологии разработки Web-приложений, которые позволяли бы строить их на компонентной основе, минимизируя затраты на интеграцию отдельных компонентов, их развертывание и поддержку в рабочем состоянии.

1.1 Технологии, используемые при разработке веб-приложений

Общая схема архитектуры Web-приложений представлена на Рисунке 1. Она включает ряд характерных элементов Web-приложений, но далеко не каждое Web-приложение содержит их все.

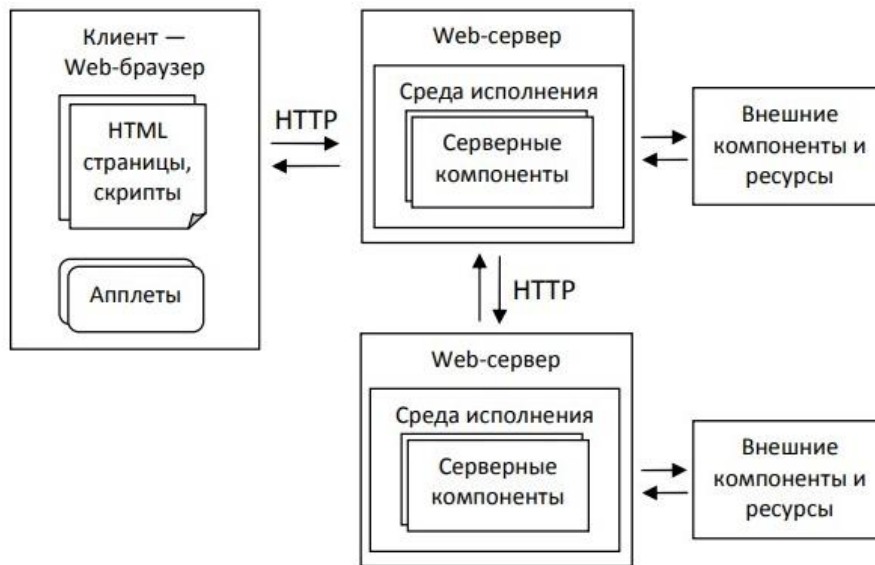


Рисунок 1. Общая архитектура Web-приложений

Клиентская часть предназначена для взаимодействия приложения с пользователем и реализуется в рамках Web-браузера в виде набора HTML-страниц (HTML — Hypertext Markup Language, язык разметки гипертекста).

Серверная часть предназначена для хранения нужной для работы приложения информации и обработки запросов пользователей. Серверная часть реализуется в виде набора компонентов, работающих в рамках Web-сервера, сервера, обрабатывающего запросы по протоколу HTTP (Hypertext Transfer Protocol, протокол передачи гипертекста), или тесно взаимодействующих с ним. Запрос пользователя, упакованный в HTTP-запрос, поступает Web-серверу на обработку, поскольку он направляется браузером по определенному URI (Uniform Resource Identifier, унифицированному идентификатору ресурса) — адресу, показываемому в адресной строке браузера, для обращения к любым ресурсам в Web.

Для поиска в глобальной сети машины, на которой размещен данный ресурс, используется протокол DNS (Domain Name Service, служба доменных имен). С помощью DNS адрес домена, т.е. базовая часть адреса ресурса, чаще всего удобная для восприятия человека (примеры:

www.google.com, www.msu.ru и пр.), транслируется в IP-адрес машины, к которой нужно обращаться со всеми запросами на соответствующие адреса.

Пользовательский интерфейс на стороне клиента реализуется с помощью Web-страниц, оформленных на языке HTML, часто с использованием таблиц стилей и скриптовых языков для динамической модификации представления страниц в браузере в зависимости от действий пользователя или конфигурации браузера (включающей его тип, набор подключенных дополнительных модулей, некоторые характеристики платформы, на которой он работает). Все подобные техники динамической модификации HTML-страниц часто объединяют под названием динамического HTML (DHTML) или клиентских сценариев (client-side scripting). Иногда, когда возможностей динамического HTML для реализации нужных свойств интерфейса не хватает, используются апплеты — приложения, работающие в рамках виртуальной машины, входящей в браузер в виде подключаемого модуля. По сравнению с обычными приложениями апплеты по соображениям безопасности обычно ограничены в доступе к локальным ресурсам (устройствам, файлам, и пр.), находящимся на машине, где работает браузер, но имеют возможность использовать внешние, доступные в Интернет ресурсы.

На стороне сервера также имеются компоненты, выполняющие функции пользовательского интерфейса. Они принимают и обрабатывают HTTP-запрос, выделяя из него информацию, существенную для данного приложения, а также компонуют или генерируют HTML-страницы (вместе с встроенными в них скриптами) для передачи их клиенту, на основе результатов обработки запроса, пришедшего от пользователя, или конфигурации браузера (насколько ее характеристики присутствуют в HTTP-запросе).

Кроме того, на стороне сервера в работе Web-приложения часто участвуют компоненты, задачей которых является собственно обработка запросов — выполнение поиска на сайте, регистрация заказа в магазине, регистрация перевода денег с кредитной карты и пр. В зависимости от используемых технологий такие компоненты могут работать как в рамках процесса Web-сервера, так и в отдельных процессах.

Очень часто обработка запросов связана с обращением к внешним по отношению к Web-приложению ресурсам — базам данных, службам каталогов и т.д.

Иногда внешние ресурсы представляют собой Web-службы. Для программиста обращение к Web-службам часто ничем не отличается от работы с другими видами ресурсов, а реализация обращений к ним выполнена в виде пересылки сообщений в HTTP-запросах к серверам, на которых находятся компоненты, реализующие такие службы. Современные корпоративные системы могут использовать десятки и сотни различных Web-служб, реализуя на их основе огромное количество функций.

Поскольку Web-приложения приобрели огромную популярность, при их разработке используется настолько много разнообразных технологий и языков, что одно их перечисление занимает много места.

1.2 Языки описания содержимого Web-страниц.

Язык разметки гипертекста (Hypertext Markup Language, HTML) служит основой для построения пользовательского интерфейса Web-приложений. Это базовый язык описания содержимого Web-страниц. Основные элементы HTML — теги и их атрибуты. Тег может соответствовать структурному элементу документа (заголовку, абзацу, таблице), элементу оформления или ссылке на другой документ, картинку,

анимацию и пр. Атрибуты тега описывают свойства соответствующих элементов.

Каскадные таблицы стилей (Cascading Style Sheets, CSS) являются языком для описания представления документов, написанных на одном из языков разметки. В таблице стилей определяются правила, по которым элементам такого документа сопоставляется стиль его представления (задающий цвет текста и фона, шрифт, форматирование абзаца и пр.). Таблицы стилей позволяют разделить описание структуры и представления документа, давая тем самым возможность отображать один и тот же документ по-разному в зависимости от конфигурации браузера или пользовательских настроек .

1.3 Протоколы, на базе которых происходит обмен сообщениями в Web-приложениях.

Протокол передачи гипертекста (Hypertext Transfer Protocol, HTTP) является основным протоколом, используемым для передачи сообщений и данных в Web-приложениях. Он предназначен для передачи прежде всего текстовых сообщений между слабо связанными сервером и клиентом. HTTP удобен тем, что большинство сетей заранее настроено на прием сообщений по этому протоколу, поскольку браузеры используются повсеместно; для многих других протоколов нужна специальная настройка, соответствующие порты по умолчанию закрыты во многих сетях. Поэтому использование Web-приложений не требует специальной настройки сетей .

1.4 Использование библиотек и фреймворков в процессе разработки программного обеспечения

С целью упрощения и ускорения процедуры разработки программного обеспечения можно использовать готовые сценарии, библиотеки или фреймворк.

Библиотека (library) – это наборы подпрограмм, модулей или пакетов, главной задачей которых является упрощение рутинных процессов программирования и поддержка сложных с технической точки зрения проектов.

Фреймворк (framework)— это набор шаблонов, заготовок. Каркас будущего проекта, на который разработчик может нанизывать дополнительные функции и фишки, которые нужны проекту. Использование фреймворков при разработке проекта упрощает и ускоряет процесс реализации программы.

Framework и Library, согласно данным из Google – инструменты для коддинга. Готовые инструменты, которые помогают при решении многих задач в программировании и разработке. Но возникает вопрос – это одинаковые объекты или нет [2].

Исходя из определений, данных в Google, оба варианта схожи между собой. Но библиотека имеет функции для вызова родительским кодом. Тем, что присутствует в языке изначально.

2. Практическая часть

2.1. Разработка модели информационной системы

На первоначальном этапе необходимо разработать модель работы информационной компонент системы. Для достижения цели требуется разработать компонент, организующий импорт данных, полученных из ГИС РО (СГО) в информационную систему образовательной организации и web-портал, позволяющие просматривать информацию неавторизованным пользователям, а также базу данных, в которой будет организовано хранение расписания уроков.

Модель работы приложения приведена на Рисунке 2. Заместитель директора по учебной работе или иное лицо, уполномоченное на загрузку данных из ГИС РО, импортирует пакет текущего расписания из ГИС РО в формате «.nsxml», после чего, используя компонент импорта данных импортирует его в информационную систему.

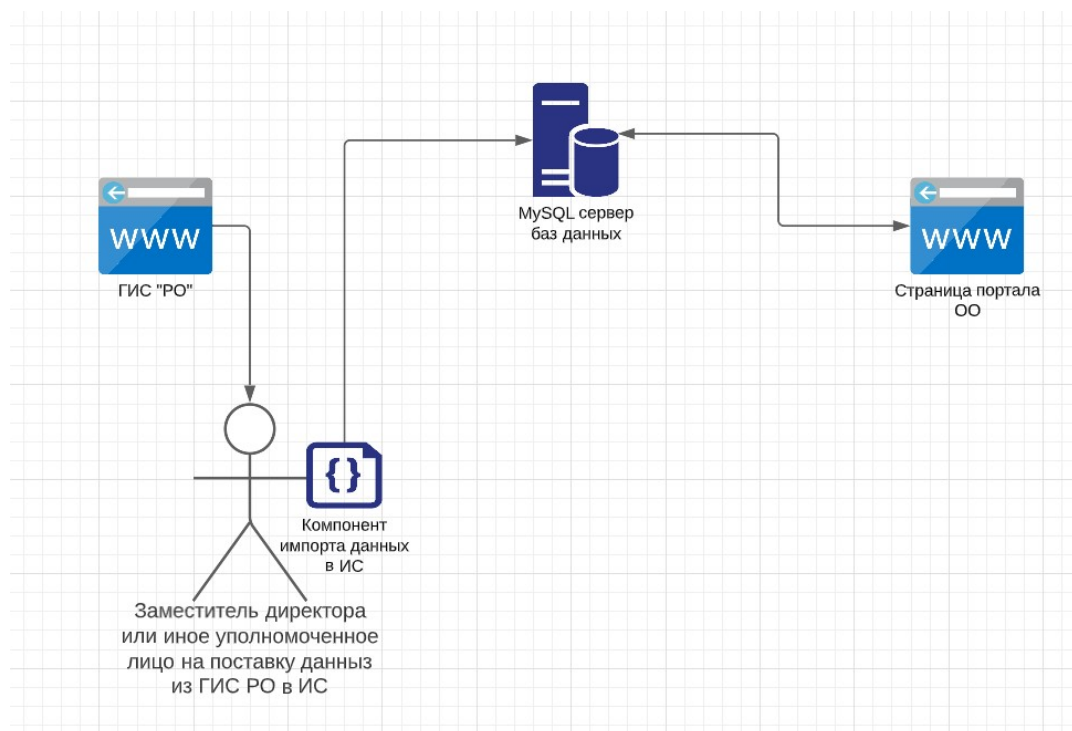


Рисунок 2

Пользователям портала без необходимости авторизации доступна функция выбора вида отображения расписания: по классам или по учителям, и дальнейший выбор интересующего класса/учителя, в соответствии со схемой, представленной на Рисунке 3.

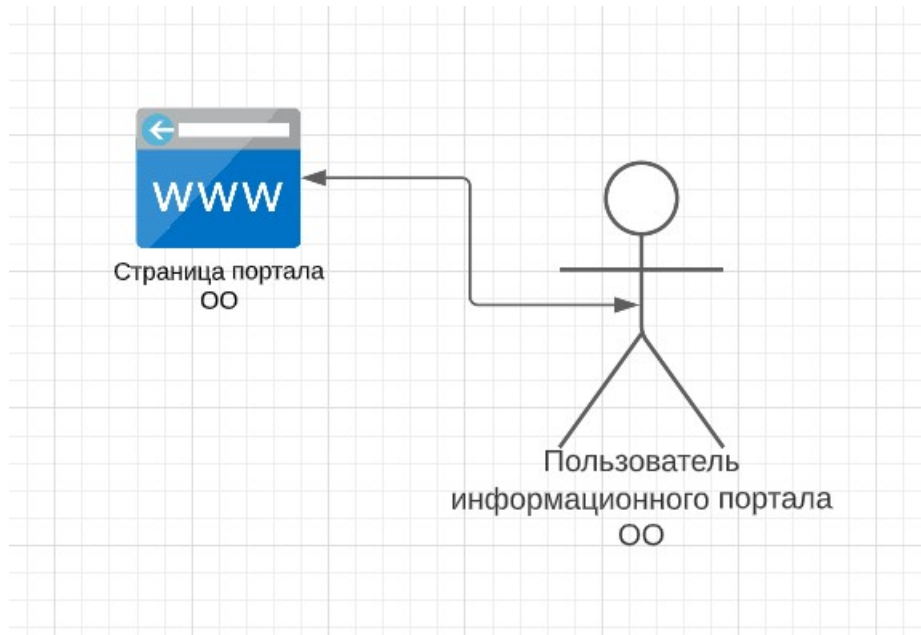


Рисунок 3

2.2. Разработка модели базы данных ИС и анализ пакета расписания «msxml»

В результате анализа файла экспорта/импорта расписания ИС «Сетевой город/регион. Образование» было выявлена совместимость формата хранения информации с форматом XML. На первом этапе моделирования базы данных было необходимо проанализировать структуру файла и выделить основные сущности, такими стали:

- «lessontimes» - расписание уроков;
- «teachers» - перечень учителей;
- «subjects» - сопоставление учителей и предметов;
- «rooms» - учебные кабинеты организации;

- «plan» – перечень предметов (учебный план);
- «timetable» - расписание уроков;

соответственно, база данных, предназначенная для хранения информации, полученной при экспорте расписания из ГИС РО, имеющего данную структуру, должна иметь аналогичную модель данных. Диаграмма «сущность-связь», отражающая модель данных базы данных изображена на Рисунке 4.

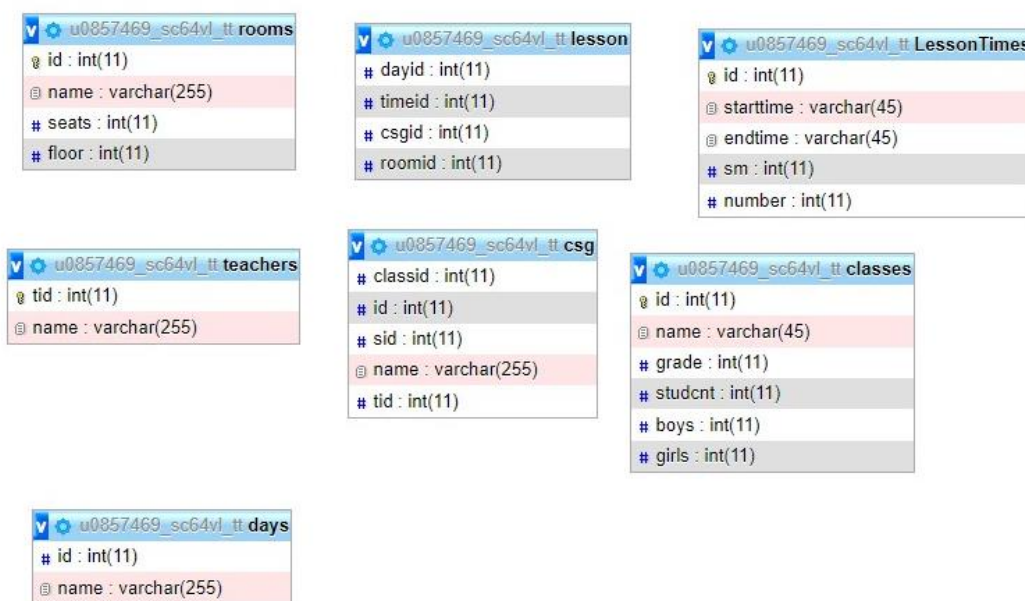


Рисунок 4

2.3. Разработка программного компонента импорта расписания в информационную систему.

Для разработки клиентской части информационной системы, реализующей функционал импорта данных файла расписания из ГИС «РО», было принято решение использовать язык программирования С#, и .NET Фреймворк.

На основании того, что файл импорта/экспорта данных ГИС «РО» имеет формат, совместимый с разметкой XML, было принято решение

разработать графический интерфейс приложения (GUI), позволяющий выбирать, используя стандартный обозреватель windows, файлы с расширением «.nsxml» [3], элементы графического интерфейса приложения приведены на Рисунке 5 и Рисунке 6.

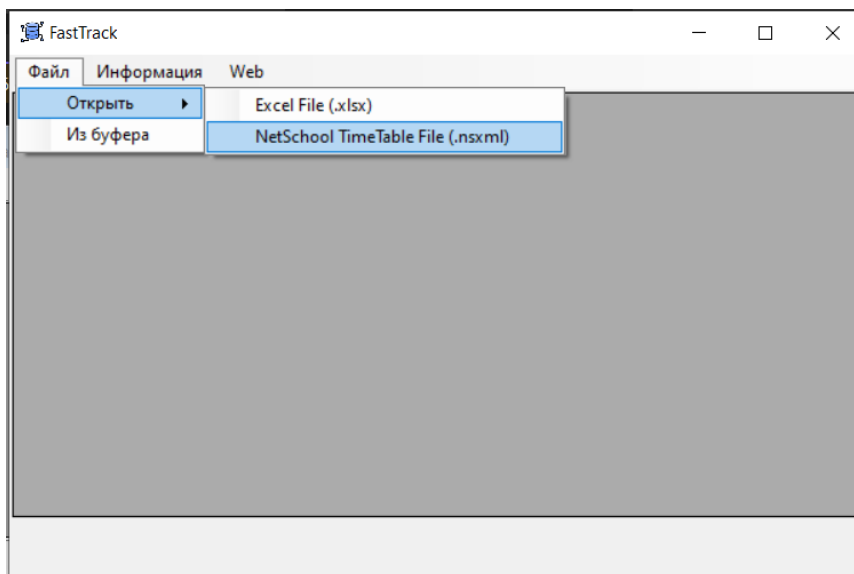


Рисунок 5

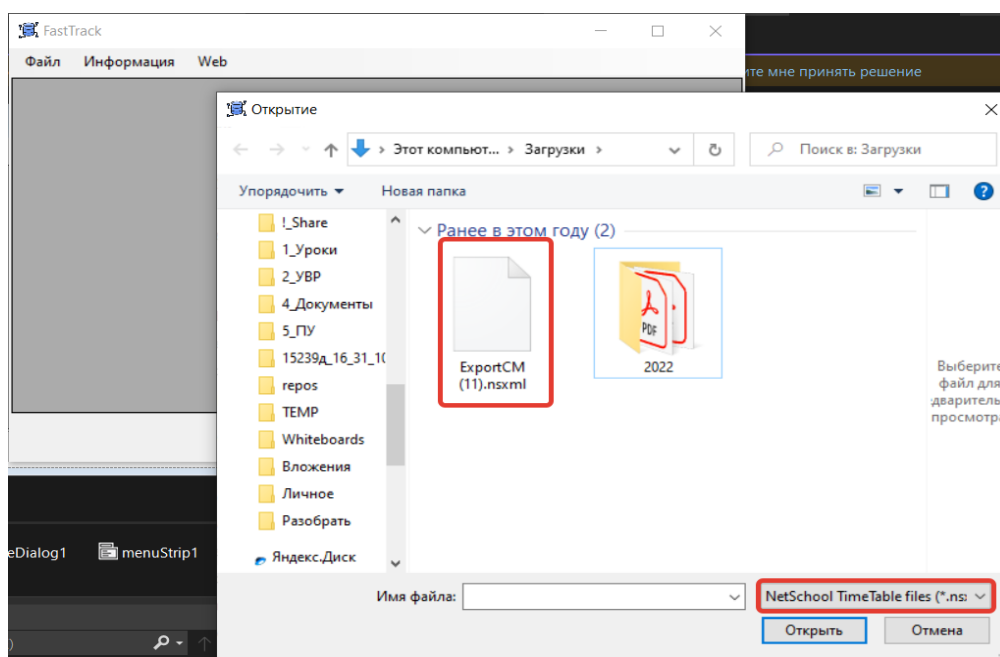


Рисунок 6

После выбора файла происходит подключение к базе данных, как показано на Рисунке 7, удаление содержимого базы данных, как показано на Рисунке

2.4. Разработка web-приложения с функцией выбора способа отображения расписания.

Одним из требований, предъявляемых к разрабатываемому web-приложению было соответствие уже установленным принципам дизайна интерфейса портала образовательной организации (www.sc64v1.ru). В основе дизайнерской линии находился HTML-фреймворк Bootstrap, разрабатываемый организацией Bootstrap Core Team и свободно копируемый. Для соответствия web-приложения дизайну портала мной была взята эталонная страница портала с расположенными на ней элементами управления и электронная справка по использованию фреймворка Bootstrap [5].

Используя язык программирования PHP было организовано отображение элементов управления (переключателя) для выбора вида отображения расписания, что продемонстрировано на Рисунке 9 и Рисунке 10.

```
echo '<div> ';
echo '
    <a style="margin: 20px;" href="?type='. 0 .'&group='.$subject.'" class="btn btn'. ($type == 0 ? '' : '-outline' ) .
    '-danger" role="button" aria-pressed="true">По классам</a>
    <a style="margin: 20px;" href="?type='. 1 .'&group='.$subject.'" class="btn btn'. ($type == 1 ? '' : '-outline' ) .
    '-danger" role="button" aria-pressed="true">По учителям</a>
  </div>
```

Рисунок 9

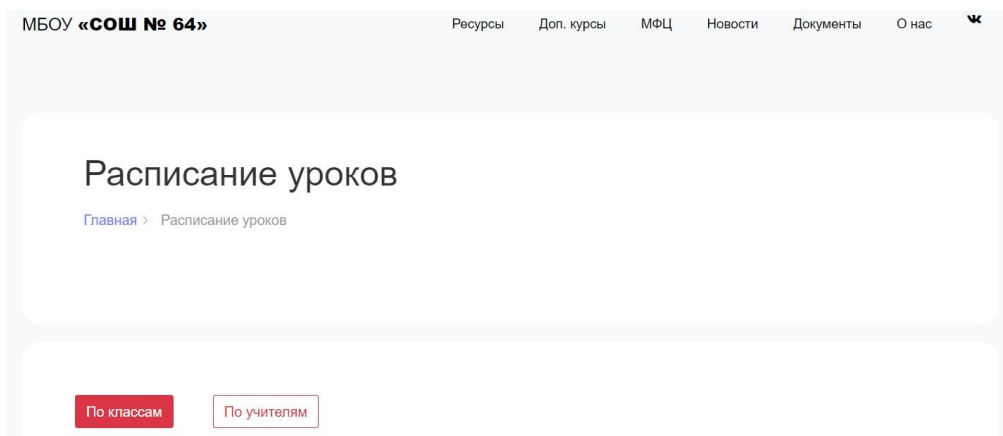


Рисунок 10

При этом, информация о выбранном способе отображения как полученная методом «GET» переменная в адресной строке, если вид отображения расписания не задан, то используется значение по умолчанию: по классам в соответствии с заданными настройками, представленными на Рисунке 11.

```
if(isset($_GET['type'])){  
    if($_GET['type'] == '1')  
    {  
        //echo 'true';  
        $type = '1';  
    }  
    else if($_GET['type'] == '2')  
    {  
        //echo 'true';  
        $type = '2';  
    }  
    else  
    {  
        $type = '0';  
    }  
} else  
{  
    $type = '0';  
}
```

Рисунок 11

На следующем этапе, пользователю предоставляется возможность выбрать класс (в случае, если был выбран способ отображения расписания по классам), как показано на Рисунке 12

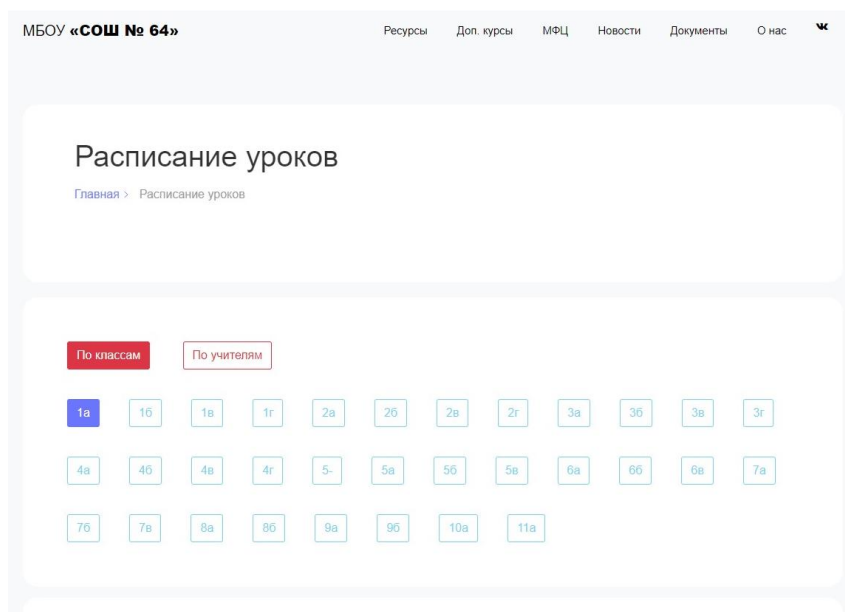


Рисунок 12

или учителям (в случае, если был выбран способ отображения по учителю), как показано на рисунке 13, по умолчанию выбирается первая позиция в списке.

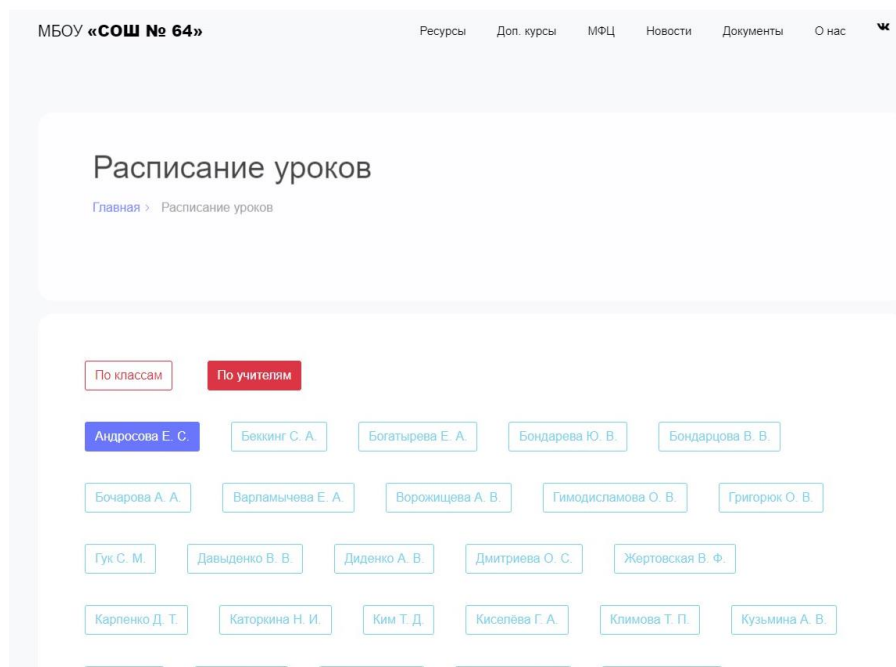


Рисунок 13

При выборе необходимых параметров (или заполнении данных по умолчанию) формируется SQL-запрос к базе данных типа «SELECT» [6], с целью получения информации о расписании и дальнейшее отображение его при помощи динамично формируемого при помощи PHP HTML-содержимого, в соответствии с рисунком 14.

Редактировать файл - table.php

Имя файла
~/www/js64vl.ru/table.php

Кодировка
UTF-8

Шрифт 12px | Режим php | Тема Chrome | Нумерация | Перенос Off

```
232
233
234 echo
235
236
237 <section class="ftco-section bg-light" style="padding: 0px;">
238   <div class="container">
239     <div class="ftco-search">
240       <div class="tab-pane fade show active" id="stuffinfo1" role="tabpanel" aria-labelledby="day-1-tab" style="padding-bottom: 15px; color:black;">
241         <div class="speaker-wrap row ftco-animate d-flex fadeInUp ftco-animate" style="border-radius: 20px; margin: 0;">
242           ;
243         </div>
244       </div>
245     </div>
246   </section>
247
248   if ($stype == 0)
249   {
250     $q = "SELECT * FROM `lesson`, `csg` where `csg`.`id`= lesson.csgid and `csg`.`id` in (select `csg`.`id` from `csg` where csg.classid = (select id from classes where classes.name= '". $group. "') ) ORDER BY `1`";
251   }
252   else if ($stype==1)
253   {
254     $q = "
255     SELECT * FROM `lesson`, `csg` where `csg`.`id`= lesson.csgid and `csg`.`id` in (select `csg`.`id` from `csg` where csg.tid = (select tid from teachers where teachers.name= '". $group. "') ) ORDER BY `lesson`.`day`";
256   }
257
258   //echo $q;
259   $results = mysqli->query( $q );
260
261
262   while ($row = mysqli_fetch_array($results)) {
263     echo($row['dayid']. ' ' . $row['timeid']. ' ' . $row['roomid']. ' ' . $row['tid']. ' ' . $row['dayid']. ' ' . $row['name']. '<br>');
264   }
265
266 }
267
268
```

Рисунок 14

Заключение

В ходе работы над данным проектом были изучены основные теоретические и практические аспекты предметной области, выполнено проектирование и разработка модели функционирования информационной системы и информационных каналов имени информацией (поставки информации) между ее компонентами, разработаны такие компоненты информационной системы, как страница web-портала (web-приложение) для публикации открытой информации о расписании уроков с функцией выбора вида отображения расписания, компонент, реализующий функционал импорта расписания в информационную систему, использующий формат экспорта/импорта данных, совместимый с ГИС «РО», таким образом была достигнута цель – разработан комплекса программных продуктов, позволяющих организовать доступ к информации о расписании уроков на сайте информационного портала образовательной организации с применение web-технологии, с возможностью выбора формата отображения расписания и функцией быстрого импорта расписания из ГИС «РО» (государственной информационной системы «Региональное образование»), работающей на базе продуктов компании ООО «ИРТЕХ», семейства «Сетевой город/регион. Образование», рассматриваем, как часть цифровой образовательной среды образовательной организации.

Список литературы

1. Формирование цифровой образовательной среды образовательной организации в условиях реализации ФГОС / Текст – электронный // Информационный портал «ИНФОРМАТИЗАЦИЯ В ШКОЛЕ»- URL: <https://itsch.ru/formirovanie-cifrovoj-sredy-fgos/> / (дата обращения: 24.02.2023). – Режим доступа: свободный.
2. Фреймворк и библиотека в JavaScript: разница, основные элементы/ Текст – электронный // Интернет-журнал «OTUS»- URL: <https://otus.ru/journal/frejmwork-i-biblioteka-v-javascript-raznica-osnovnye-elementy/> / (дата обращения: 2.03.2023). – Режим доступа: свободный.
3. Работа с XML с помощью классов System.Xml / Текст – электронный // Справочный портал «metanit.com» - URL: <https://metanit.com/sharp/tutorial/16.2.php/> / (дата обращения: 2.03.2023). – Режим доступа: свободный.
4. MySQL и C# — работаем с базой данных / Текст – электронный // Информационный портал «Программирование на C, C# и Java» - URL: <https://vscode.ru/prog-lessons/mysql-c-sharp.html/> / (дата обращения: 8.03.2023). – Режим доступа: свободный.
5. Build fast, responsive sites with Bootstrap / Текст – электронный // Официальный интернет-сайт организации Bootstrap - URL: <https://getbootstrap.com/> / (дата обращения: 8.03.2023). – Режим доступа: свободный.
6. Работа с MySQL в PHP / Текст – электронный // ООО «Интерактивные обучающие технологии»- URL: <https://htmlacademy.ru/blog/php/mysql/> / (дата обращения: 5.03.2023). – Режим доступа: свободный.