

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Математико-механический факультет

Кафедра системного программирования

РАЗРАБОТКА ВЕБ-СЕРВИСА «ЭЛЕКТРОННЫЙ  
ДНЕВНИК» ДЛЯ РЕАЛИЗАЦИИ ОБРАТНОЙ СВЯЗИ  
«ШКОЛА-РОДИТЕЛЬ»

Выпускная квалификационная работа

Краснопёрова Алексея Александровича

Научный руководитель	..... / подпись /	доцент, к.ф.-м.н. Кияев В.И.
Рецензент	..... / подпись /	асп. каф. сис. прог. Тимофеев Н.М.
“Допустить к защите” заведующий кафедрой	..... / подпись /	д.ф.-м.н., проф. Терехов А.Н.

Санкт-Петербург  
2010

SAINT-PETERSBURG STATE UNIVERSITY  
Mathematics & Mechanics Faculty

Software Engineering Chair

“PARENT-SCHOOL” FEEDBACK REALIZATION IN “E-  
JOURNAL” WEB-SERVICE

by

Alexey Krasnoperov

Bachelor’s graduation paper

Supervisor	.....	PhD, As. Prof. V. I. Kiyaev
Reviewer	.....	Postgraduate N. M. Timofeev
“Approved by”	.....	Professor A. N. Terekhov
Head of Chair		

Saint-Petersburg  
2010

## Оглавление

<b>1. Введение</b> .....	<b>3</b>
<b>2. Постановка задачи</b> .....	<b>5</b>
<b>3. Обзор</b> .....	<b>6</b>
3.1. Обзор существующих решений .....	6
3.2. Обзор средств разработки.....	9
3.2.1. Система Управления Базами Данных.....	9
3.2.2. Работа с базами данных и генерация HTML.....	10
3.2.3. AJAX.....	11
3.3. Кроссбраузерность и стандарты веб-разработки .....	13
<b>4. Функциональность и архитектура</b> .....	<b>15</b>
4.1. Основные идеи.....	15
4.2. Описание функциональности .....	15
4.3. Описание архитектуры .....	16
4.3.1. Классификация пользователей.....	16
4.3.2. Пользовательские модули.....	18
4.3.3. Структура Базы Данных.....	19
4.3.4. Модель-Представление-Контроллер(Model-View-Controller) .....	20
4.4. Особенности реализации.....	23
4.4.1. Использование «толстых» моделей вместо «тонких».....	23
4.4.2. Ajax-компоненты.....	23
4.4.3. Влияние веб-сервера на быстроедействие.....	24
4.4.4. Противодействие SQL-инъекциям и XSS-атакам .....	25
4.4.5. Периодические задания и нюансы настройки демона-планировщика Cron.....	27
4.5. Ускорение проекта .....	28
<b>5. Обзор веб-сервиса «Электронный дневник»</b> .....	<b>30</b>
5.1. Интерфейс экранной формы «Родитель».....	30
5.2. Интерфейс экранной формы «Учитель».....	31
<b>6. Заключение</b> .....	<b>34</b>
<b>Список литературы</b> .....	<b>36</b>

## 1. Введение

В настоящее время существует проблема коммуникации между родителями и школой, в которой учится их ребенок. Зачастую ребенок не хочет или не может показывать плохие оценки родителям, поэтому он идет на различные ухищрения: исправляет оценки в дневнике на более «правильные», ведет разные дневники для плохих и хороших оценок, «забывает» сообщить о родительских собраниях, создает припятствия для личного общения классного руководителя с родителями.

Также традиционные методы коммуникации не удобны родителям: родительские собрания занимают слишком много времени и часто проводятся в неприемлимое для родителей время. Родителям «хороших» учеников приходится ехать в школу и в течение двух часов слушать замечания, адресованные не их ребенку, только для того, чтобы в итоге услышать, что с их ребенком все в порядке и им можно не беспокоиться. А родители «плохих» учеников могли бы отреагировать на все негативные события в жизни ребенка заблаговременно, если бы они информировались оперативней. Ведь часто случается, что ребенку просто требуется небольшая помощь в освоении материала или в решении каких-либо психологических проблем, но родители даже не подозревают об этом, и в итоге ситуацию оказывается не просто исправить.

Эта проблема не является новой и поэтому на рынке программного обеспечения уже существует ряд продуктов, так или иначе пытающихся устранить коммуникационный провал. За решение этой задачи взялись как небольшие частные фирмы, так и организации типа банков и университетов.

Ведущими электронными дневниками на сегодняшний день являются MGUshka.ru [1], 1dnevnik.ru [2], dnevnik.ru [3], ballov.net [4]. Кроме приведенных выше примеров существует множество других, менее известных и распространенных систем.

Изучив охват этих проектов и проведя блиц-опрос школ, я могу сделать вывод, что далеко не все школы информированы о существовании электронных дневников и активно их используют. А так как качество образования оставляет желать лучшего и повсеместное использование электронных дневников в школах могло бы поспособствовать улучшению ситуации, есть все основания считать обоснованным реализацию собственного электронного дневника.

По статистике из открытых источников [15] можно увидеть, что, даже в условиях демографической ямы, количество школьников России превышает 12 миллионов человек. Следовательно, при реализации системы «Электронный дневник», необходимо учитывать возможность высоких нагрузок в процессе её эксплуатации.

## 2. Постановка задачи

Целью данной дипломной работы является реализация веб-сервиса «Электронный Дневник» с изучением сопутствующих его реализации и обеспечению бесперебойного высоконагруженного функционирования технологий.

Таким образом, необходимо:

- изучить необходимый минимум для реализации качественного веб-сервиса:
  - стандарты веб-разработки;
  - шаблоны проектирования ;
- изучить способы ускорения высоконагруженных веб-приложений;
- выбрать конечную концепцию электронного дневника;
- реализовать веб-сервис «Электронный Дневник».

## 3. Обзор

### 3.1. Обзор существующих решений

В данный момент на рынке существует большое количество компаний, предоставляющих сервис электронных дневников. Каждая из них обладает своими достоинствами и недостатками.

Так, например, проект МГУшка [1] предлагает родителям сервис SMS-дневников. Каждый вечер в определенное время родителям приходит SMS-сообщение, содержащее информацию о полученных их ребенком оценках и о домашнем задании на следующий день. Данная модель не является удобной для родителей – трудно оценить тенденции в образовании школьника, понять случайная ли это двойка или провал в знаниях. Кроме того, чтобы выяснить какие предметы нужно срочно подтягивать для получения хорошей триместровой оценки, родителям приходится самостоятельно выписывать оценки из каждого SMS-сообщения, что тоже не является приемлемым из-за малого «периода жизни» SMS-сообщений у активных пользователей или пользователей с памятью телефона, не позволяющей хранить много сообщений.

Также на рынке имеются масштабные решения, такие как [dnevnik.ru](http://dnevnik.ru) [3], своим функционалом повторяющие полноценные социальные сети. Пользователям помимо основной функциональности электронного дневника предлагается также сервис хранения файлов, фотографий, видеозаписей, возможность объединения людей в сообщества, создание различных событий и прочие, полезные на первый взгляд,

возможности. Но родителям трудно освоить весь предлагаемый им функционал, что приводит к путанице и полному непониманию программного продукта. Учителям же разобраться в системе еще сложнее, достаточно малый процент преподающих в школах учителей – уверенные пользователи Интернета. Поэтому одной из задач при реализации электронного дневника было обеспечение максимальной простоты для работы в системе учителей.

Опрос, проведенный автором данной работы в среднестатистических российских школах показал, что только у 7% учителей есть домашний компьютер с подключением к Интернету. Это также явилось проблемой, которую необходимо было решить в процессе реализации проекта «Электронный дневник».

Большинство рассмотренных мной проектов имели бизнес-модель, предполагающую значительные траты со стороны школы. Например, проект 1dnevnik.ru [2] берет за одно лишь подключение школы к системе 50000 руб. МГУшка [1] предоставляет SMS-дневник с оплатой за каждую отправленную SMS в размере 10 руб. за сообщение и стоимостью подключения в размере 21000 руб. Некоторые проекты готовы предоставлять свои услуги с ежемесячной оплатой и бесплатным подключением. И ни один из рассмотренных мной проектов не предполагал отчислений в адрес школы. Данный факт также был учтен при разработке собственного электронного дневника, поэтому бизнес-модель проекта «Электронный дневник» предполагает выплаты школам процента от перечисленных средств. Это оказывает заметную



материальную поддержку школам, а так же мотивирует руководство школ к подключению к системе новых классов.

### 3.2. Обзор средств разработки

Учитывая задачи и функциональность проекта, было принято решение реализовывать веб-сервис «Электронный дневник» с использованием следующих технологий:

- HTML;
- CSS;
- JavaScript;
- PHP;
- MySQL.

Эти технологии взаимодействуют между собой, как это показано на рисунке (Рис. 1).

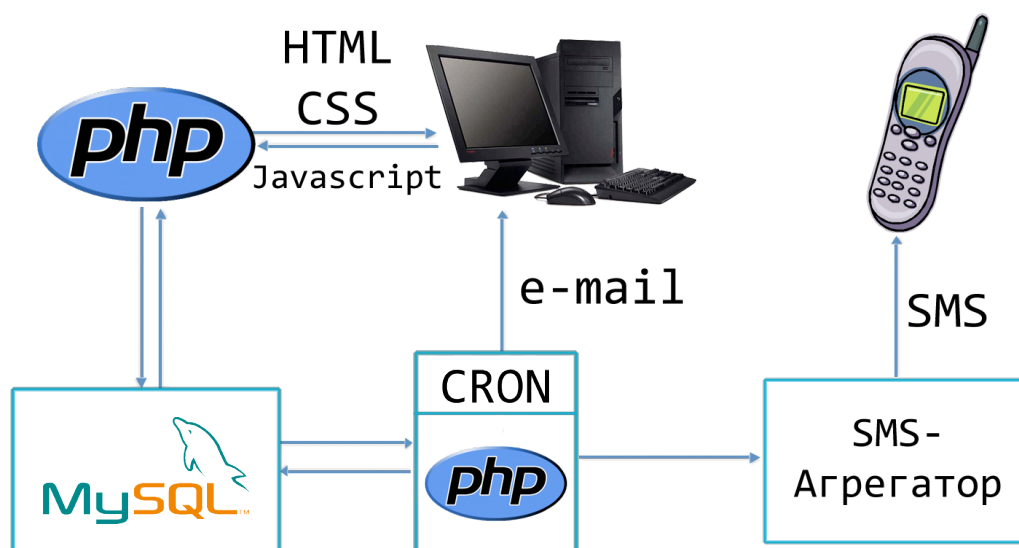


Рис. 1. Схема взаимодействия технологий

#### 3.2.1. Система Управления Базами Данных

В качестве СУБД выбрана MySQL версии 5 — свободная система управления базами данных, распространяемая под GNU General Public License и под собственной коммерческой лицензией, на выбор.

В MySQL 5 значительно расширена функциональность, которая ставит MySQL в один ряд с коммерческими СУБД. Если раньше СУБД MySQL обвиняли в недостаточной поддержке стандарта SQL, то с появлением пятой версии этой популярной базы данных, появилась практически полная поддержка стандарта SQL. [5]

После рассмотрения всех плюсов и минусов, был выбран тип таблиц InnoDB – одна из выбираемых подсистем низкого уровня в СУБД MySQL, которая входит во все стандартные сборки для различных операционных систем. Основным отличием InnoDB от других подсистем низкого уровня MySQL является наличие механизма транзакций.

В отличие от таблиц MyISAM, где для каждой таблицы создается один файл данных, данные InnoDB в настройках по умолчанию хранятся в больших совместно используемых файлах (изменить это можно с помощью настроек опции `innodb_file_per_table`), что позволяет использовать постраничный кэш страниц базы данных. [6]

Формат данных InnoDB обеспечивает надежное хранение данных за счет транзакционности и блокирование данных на уровне строки, а не всей таблицы. Это позволяет редактировать данные большому количеству человек без каких-либо проблем.

### **3.2.2. Работа с базами данных и генерация HTML**

Для работы с базами данных и дальнейшей генерации HTML используется язык PHP — один из популярнейших скриптовых языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей простоте, скорости

выполнения, богатой функциональности и распространению исходных кодов на основе лицензии PHP. PHP отличается наличием ядра и подключаемых модулей, «расширений»: для работы с базами данных, сокетами, динамической графикой, криптографическими библиотеками, документами формата PDF и т. п. Интерпретатор PHP подключается к веб-серверу Apache через специально разработанный для него модуль.

Наиболее стабильная версия на момент написания – 5.xx, была выпущена разработчиками 13 июля 2004 года. Изменения включают обновление ядра Zend (Zend Engine 2), что существенно увеличило эффективность интерпретатора. Введена поддержка языка разметки XML. Полностью переработаны функции ООП, которые стали во многом схожи с моделью, используемой в Java. В частности, введён деструктор, открытые, закрытые и защищённые члены и методы, окончательные члены и методы, интерфейсы и клонирование объектов. Нововведения, однако, были сделаны с расчётом сохранить наибольшую совместимость с кодом на предыдущих версиях языка. [7]

### **3.2.3. AJAX**

Для редактирования таблиц без перезагрузки страницы, использовалась концепция AJAX. В данном случае она реализована связкой Javascript и PHP.

С помощью концепции AJAX пользователь может работать с таблицами так, как если бы он работал в обычном приложении у себя на компьютере.

При написании веб-сервиса использовались свободные фреймворки, такие как Prototype, упрощающие работу с AJAX и некоторыми другими функциями. В Prototype присутствуют самые разные способы упрощения создания JavaScript приложений, от сокращённого вызова некоторых функций языка до сложных методов обращения к XMLHttpRequest.

Заявлено, что данный фреймворк поддерживается следующими браузерами: Internet Explorer (Windows) 6.0+, Mozilla Firefox 1.5+, Apple Safari 2.0+ и Opera 9.25+. Поддержка данных браузеров также подразумевает, что фреймворк поддерживается также Google Chrome, Camino, Konqueror, IceWeasel, Netscape 7+, SeaMonkey, и др., которые принадлежат этим же семействам. Таким образом охватываются все популярные браузеры, использующиеся в настоящее время.

### **3.3. Кроссбраузерность и стандарты веб-разработки**

Одной из проблем при создании публичного веб-проекта является обеспечение одинакового его отображения и работы в различных браузерах под различными системами, то есть обеспечение кроссбраузерности.

Поэтому в октябре 1994 года был основан Консорциум Всемирной Паутины (World Wide Web Consortium, W3C), разрабатывающий и внедряющий стандарты для всемирной паутины. W3C разрабатывает для Интернета единые принципы и стандарты, которые затем внедряются производителями программ и оборудования. Таким образом достигается совместимость между программными продуктами и аппаратурой различных компаний, что делает Всемирную сеть более совершенной, универсальной и удобной. [8]

Таким образом, соблюдение единых стандартов как со стороны разработчиков веб-сервисов, так и со стороны производителей программного обеспечения, обеспечивает полную совместимость и кроссбраузерность разрабатываемых веб-проектов.

Для реализации клиентской части веб-сервиса «Электронный дневник» был выбран строгий расширяемый язык разметки гипертекста версии 1.0 (XHTML 1.0 Strict), одобренный в качестве рекомендации Консорциума Всемирной Паутины. Стандарт XHTML построен не как самостоятельное описание языка, а как перечень различий между HTML 4.01 и XHTML. Такими различиями, например, являются следующие пункты:

- все элементы должны быть закрыты. Теги, которые не имеют закрывающего тега (например, `<img>` или `<br>`) должны иметь на конце `/` (например, `<br />`);

- булевы атрибуты записываются в развёрнутой форме. Например, следует писать `<option selected="selected">` или `<td nowrap="nowrap">` вместо `<option selected>` или `<td nowrap>` соответственно;

- имена тегов и атрибутов должны быть записаны строчными буквами (например, `<img alt="" />` вместо `<IMG ALT="" />`);

- XHTML гораздо строже относится к ошибкам в коде; `<` и `&` везде, даже в URL, должны замещаться `&lt;` и `&amp;` соответственно. По рекомендации W3C браузеры, встретив ошибку в XHTML, должны сообщить о ней и не обрабатывать документ. Для HTML браузеры должны были попытаться понять, что хотел сказать автор.

- кодировкой по умолчанию является UTF-8 (в отличие от HTML, где кодировкой по умолчанию является ISO 8859-1).

Код XHTML-страниц веб-сервиса «Электронный дневник» полностью соответствует рекомендуемым Консорциумом Всемирной Паутины стандартам. Таким образом, при отсутствии нарушений со стороны производителей программного обеспечения, веб-сервис «Электронный дневник» является полностью кроссбраузерным, кроссплатформенным продуктом.

## **4. Функциональность и архитектура**

### **4.1. Основные идеи**

Основной идеей веб-сервиса «Электронный дневник» является обеспечение оперативной и надежной коммуникации между родителями и школой в условиях нехватки времени и физической невозможности личной встречи родителя с представителем школы.

Родители должны иметь возможность получить необходимую им информацию, где бы они не находились. Также необходимо обеспечить родителям доступ к актуальной информации при невозможности использования компьютера с Интернетом.

Учителя же должны иметь доступ ко всей необходимой им информации и средствам коммуникации в удобной для них форме. При этом интерфейс взаимодействия учителя и системы должен быть максимально простым для освоения его даже очень далекими от информационных технологий лицами.

### **4.2. Описание функциональности**

Для начала пользования сервисом «Электронный дневник», пользователю необходимо представиться системе. От того, к какой группе принадлежит пользователь, зависит предоставляемая ему функциональность.

Доступ ко внутренней функциональностью системы для незарегистрированных пользователей закрыт. Регистрация производится по специальным приглашениям,



генерирующимся автоматически при добавлении учителем нового ученика.

Зарегистрированные пользователи разделяются на категории, в зависимости от которых им выдаются различные права. Существует 2 основные категории пользователей – это «Родитель» и «Учитель».

В веб-сервисе «Электронный дневник» реализована система автоматических уведомлений – SMS- и email-сообщения, отправляемые пользователям с заданной периодичностью.

### **4.3. Описание архитектуры**

#### **4.3.1. Классификация пользователей**

В рамках системы «Электронный дневник» различаются следующие классы пользователей:

- гость(guest);
- пользователь(member);
- администратор(admin);
- родитель(parent);
- учитель(teacher).

*Гость* – класс, имеющий доступ к минимальной функциональности. К классу «Гость» относится любой человек, не зарегистрированный в системе, либо не прошедший аутентификацию.

*Пользователь* – класс, не имеющий никаких привилегий и имеющий очень ограниченное количество возможностей. К классу «Пользователь» относится любой человек, прошедший регистрацию, но по какой-либо причине не прошедший идентификацию с дальнейшим причислением к классу «Родитель» или «Учитель». Этот класс является базовым для нижеприведенных классов.

*Администратор* – класс, имеющий максимальные привилегии. Имеет право редактирования любой информации, относящейся к проекту «Электронный дневник», а так же осуществляющий помощь при возникновении трудностей у других пользователей.

*Родитель* – класс, предоставляющий пользователю возможность просматривать информацию о собственном ребенке, вступать в переписку с учителями его класса и др. К классу «Родитель» относятся пользователи, прошедшие проверку на соответствие личности родителя, либо зарегистрированные по специальным приглашениям, выдаваемым учителями. Пользователю класса «Родитель» предоставляется следующая функциональность:

- просмотр текущих оценок:
  - на заданный период;
  - в виде классического дневника на неделю;
- просмотр статистики по оценкам;
- просмотр расписания класса;
- просмотр личных замечаний ребенка;

- просмотр объявлений уровня школы, класса, подгруппы;
- возможность общения с учителями через веб-интерфейс;
- подключение и отключение SMS- и email-уведомлений.

*Учитель* – класс, предоставляющий максимум возможных привилегий внутри класса, классным руководителем которого является пользователь. К этому классу относятся пользователи, прошедшие проверку на соответствие личности учителя/классного руководителя, либо зарегистрированные по специальным приглашениям от администрации проекта. Пользователь, причисленный к классу «Учитель» может редактировать оценки своего класса, расписание, объявления, замечания и прочую внутриклассную информацию, а также создавать приглашения для пользователей класса «Родитель», тем самым подключая желающих родителей к системе. Пользователи класса «Учитель», наравне с пользователями класса «Родитель», могут подключать последним SMS-уведомления. Эта функциональность используется, когда родитель не имеет свободного доступа к Интернету, но желает получать оперативную информацию о предоставляемых их ребенку оценках в виде коротких сообщений на телефон.

#### **4.3.2. Пользовательские модули**

Для разделения функционала перечисленных выше классов пользователей, было решено создать модули пользователей. Классы «Гость» и «Пользователь» используют модуль по умолчанию(default).

Для расширения функциональности до уровня классов «Администратор», «Учитель» и «Родитель», дополнительно

подключаются модули admin, teacher и parent соответственно. В каждом из этих модулей размещаются контроллеры и представления, обеспечивающие функциональность соответствующего класса.

### 4.3.3. Структура Базы Данных

В процессе проектирования веб-сервиса «Электронный дневник» было принято решение о хранении всего динамически изменяющегося содержимого в базе данных.

База данных используется для хранения информации о зарегистрированных пользователях, их привилегиях и настройках, и принадлежности к учебному классу. Также в базе данных хранятся внутришкольные объявления, домашние задания, оценки, личные сообщения, информация о предметах и привилегиях (Рис. 3).

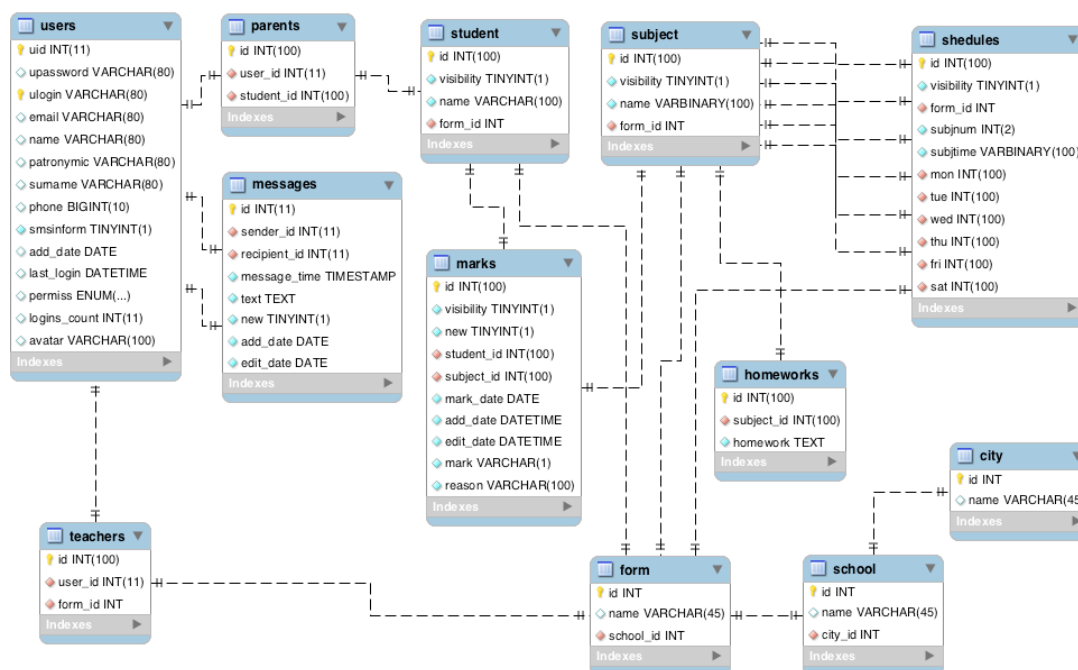


Рис. 3. Схема базы данных

У каждой таблицы задан первичный ключ id, благодаря которому можно однозначно идентифицировать любую запись таблицы.

#### **4.3.4. Модель-Представление-Контроллер(Model-View-Controller)**

В качестве архитектуры серверной части проекта было принято решение использовать архитектуру «Модель-Представление-Контроллер» (Рис. 4), т.е. архитектуру программного обеспечения, в которой модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента, так, что модификация одного из компонентов оказывает минимальное воздействие на другие компоненты.

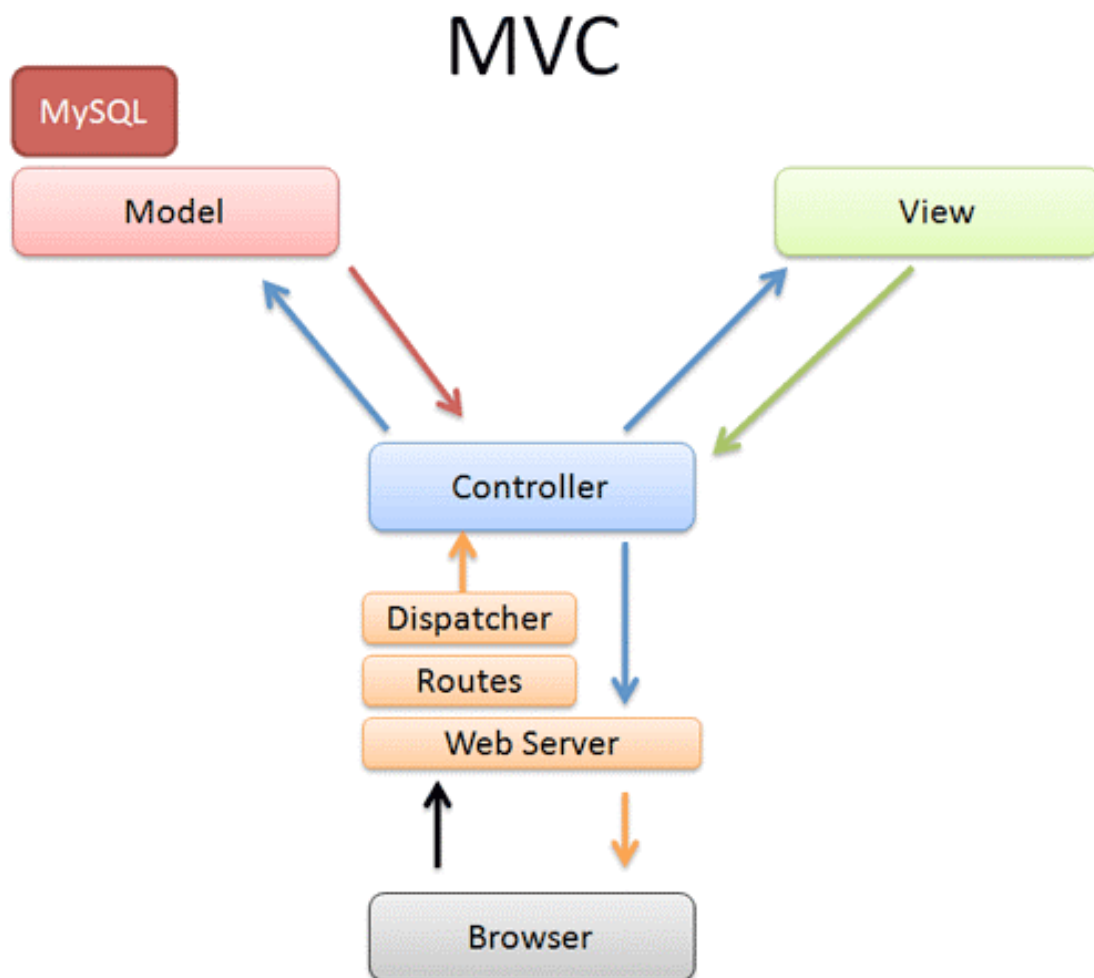


Рис. 4. Схема архитектуры «Модель-Представление-Контроллер»<sup>1</sup>

Шаблон MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

- модель (Model). Модель предоставляет данные (обычно для View), а также реагирует на запросы (обычно от контроллера), изменяя своё состояние;
- представление (View). Отвечает за отображение информации (пользовательский интерфейс);

---

<sup>1</sup> Режим доступа: <http://www.bhartisoftland.com/technologies-skill-sets/gifs/mvc-php.png>

- контроллер (Controller). Интерпретирует данные, введённые пользователем, и информирует модель и представление о необходимости соответствующей реакции.

Важно отметить, что как представление, так и поведение зависят от модели. Однако модель не зависит ни от представления, ни от поведения. Это одно из ключевых достоинств подобного разделения, позволяющее строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

#### **4.4. Особенности реализации**

##### **4.4.1. Использование «толстых» моделей вместо «тонких»**

«Тонкая» модель – модель, роль которой состоит лишь в доступе к данным, а бизнес-логику, чтение и запись данных и проверку входящей информации производит контроллер.

При использовании же «толстой» модели, она наделяется всем необходимым функционалом, позволяющим правильно распределить обязанности компонентов, разграничив логику приложения, и избавиться от зависимости модели и контроллера, что и является основной идеей архитектуры «Модель-Представление-Контроллер».

В проекте «Электронный дневник» используются только «толстые» модели, обеспечивая тем самым более легкое развитие и поддержку, чем если бы использовались «тонкие» модели. Контроллеры лишь являются связующим звеном, но не управляют данными напрямую, а вся работа с данными возложена на модели, таким образом переход на другую систему управления базами данных можно произвести быстро и легко, корректируя только модели, а не всё приложение целиком.

##### **4.4.2. Ajax-компоненты**

Для облегчения работы конечного пользователя с системой было принято решение о внедрении Ajax-компонентов в некоторые модули веб-сервиса.

Ajax – подход к построению интерактивных пользовательских интерфейсов веб-приложений,



закрывающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных, веб-страница не перезагружается полностью и веб-приложения становятся более быстрыми и удобными. [9]

Эта концепция оказалась очень удобной для использования её на страницах редактирования информации учителем: оценки, расписание, замечания редактируются всего лишь парой нажатий кнопки мыши. Это позволяет пользователю сосредоточиться на непосредственной работе с системой, а не на изучении ее особенностей, что крайне важно для учительского состава, имеющих массу трудностей при освоении современных информационных технологий.

Для редактирования этих данных, представленных в основном в табличном виде, была использована javascript-библиотека Tablekit, основанная на фреймворке Prototype – бесплатном javascript-фреймворке, значительно упрощающим работу с Ajax и некоторыми другими функциями.

Также Ajax-концепция позволяет снизить количество запросов к серверу, тем самым значительно понижает нагрузку веб-сервиса, что является значимым для проекта, поддерживающего большое количество одновременных посетителей.

#### **4.4.3. Влияние веб-сервера на быстродействие**

В качестве веб-сервера для проекта «Электронный дневник» была выбрана схема фронтенд-бэкенд в виде nginx и Apache соответственно.

Такая схема позволяет разгрузить тяжелый Apache от статических запросов, перенеся раздачу статического содержимого на nginx, а в случае обращений к динамическому содержимому nginx выступает в качестве посредника между пользователем и бэкендом в виде Apache. Такое посредничество дает значительный прирост в производительности за счет скорости отдачи содержимого от бэкенда к фронтенду в отличие от прямой отдачи от Apache к пользователю. Ведь у большинства пользователей Интернет-каналы не позволяют забирать содержимое со скоростью, равной генерации содержимого, поэтому происходит ситуация, когда в один момент запущено множество httpd-процессов, занимающих большое количество серверных ресурсов.

Введя в классическую схему фронтенд в виде nginx, мы получили выигрыш во времени работы httpd-процессов, ведь теперь Apache передает не ограниченному в Интернет-канале пользователю, а легкому веб-серверу nginx, связь с которым позволяет отдавать содержимое в сотни раз быстрее. Таким образом апач на каждый запрос занят не 10 секунд, а 0.1 секунду, и процессов httpd будет запущено не 1000, а 10. [13]

Ответ при этом буферизируется в nginx, но объем ответа обычно в несколько раз меньше, чем объем процесса на бэкенде (httpd+perl, PHP или даже Java), который этот ответ создал, поэтому память будет очень заметно экономиться.

#### **4.4.4. Противодействие SQL-инъекциям и XSS-атакам**

*Внедрение SQL-кода (SQL-инъекция)* – один из распространённых способов взлома сайтов и программ,

работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

Внедрение SQL-кода, в зависимости от типа используемой СУБД и условий внедрения, может дать возможность атакующему выполнить произвольный запрос к базе данных (например, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере. [16]

Поэтому важно избавить злоумышленника от такой возможности. Для обеспечения безопасности исполняемых SQL-запросов был использован объект `Zend_Db`, входящий в `Zend Framework` – свободный каркас для создания веб-приложений и веб-сервисов от разработчиков языка PHP, а также дополнительные функции приведения входных данных к корректному виду, такие как `addslashes`, производящая экранирование спецсимволов.

XSS – тип уязвимости интерактивных информационных систем в вебе. XSS возникает, когда в генерируемые сервером страницы по какой-то причине попадают пользовательские скрипты. Специфика подобных атак заключается в том, что вместо непосредственной атаки сервера они используют уязвимый сервер в качестве средства атаки на клиента. [17]

Для обеспечения невозможности проведения XSS-атак был проведен ряд предупреждающих мероприятий, таких как экранирование данных при вводе и выводе с помощью `addslashes` и удаление html-тегов при вводе с помощью функции `strip_tags`.

Для фильтрации и приведения данных к безопасному виду используются классы каркаса Zend Framework, такие как Zend\_Filter\_StringTrim, в частности, содержащаяся в нем функция filter():

```
$filter = new Zend_Filter_StringTrim();
```

```
$email = $filter->filter($email);
```

```
$password = $filter->filter($password);
```

и пр.

Эти и многие другие аспекты безопасности реализуются с помощью элементов каркаса Zend Framework, что позволяет обеспечить защищенность веб-проекта «Электронный дневник» на достаточном уровне.

#### **4.4.5. Периодические задания и нюансы настройки демона-планировщика Cron**

Для реализации периодических заданий, таких как рассылка SMS- и email-уведомлений было решено использовать связку демона-планировщика задач Cron и PHP-скриптов, отвечающих за рассылку уведомлений. Согласно проведенному опросу, такое решение удобней для родителей, чем получение отметок сразу при их выставлении, ведь отправка с помощью планировщика позволяет собрать в одно SMS-сообщение сразу все отметки за день и родитель не будет вынужден искать все сообщения за сутки чтобы понять как прошел учебный день у его ребенка. При этом SMS-сообщение выглядит следующим образом:

«Иванов Иван.

Новые оценки.

Математика: 5

Русский язык: 4»

В процессе реализации были выявлены неочевидные нюансы установки таблицы заданий в демоне-планировщике Cron.

Во-первых для выполнения PHP-скриптов необходимо указать полный путь к PHP-интерпретатору, иначе скрипт просто не выполняется, хотя при тех же условиях из командной строки он выполняется и с использованием алиаса «php».

Во-вторых рекомендуется выполнять PHP-интерпретатор с параметром “-q”, который указывает интерпретатору выполнять PHP-скрипт без ненужных в данном случае http-заголовков.

В-третьих после строки, задающей выполнение PHP-скрипта необходимо перевести курсор на следующую строку. Без этого Cron будет игнорировать существующее правило.

Таким образом, строка для выполнения SMS-уведомлений ежедневно в 20.00 будет выглядеть следующим образом:

```
0 20 * * * /usr/local/php5/bin/php-cli -q /home/username/domain.com/cron/sms_notify.php
```

#### **4.5. Ускорение проекта**

При обычном исполнении PHP скрипта выполняются такие шаги, как чтение PHP-файла, генерация из него байт-кода, выполнение кода и выдача результата.

Мы можем избавиться от первого пункта, чтения PHP-файла, путем кеширования байт-кода в памяти или на диске.

Для этих нужд существуют акселераторы PHP-кода, такие как eAccelerator или APC.

Данные из открытых источников [10] показывают, что eAccelerator и APC примерно равны в своем оптимизирующем эффекте, поэтому для ускорения веб-сервиса «Электронный дневник» был выбран eAccelerator как более привычный и удобный для автора инструмент.

Замеры, произведенные после подключения акселератора демонстрируют повышение производительности по количеству запросов в секунду в среднем в 2 раза, уменьшение потребляемой памяти в 3 раза и уменьшение времени обработки запроса в 2 раза.

## 5. Обзор веб-сервиса «Электронный дневник»

При использовании системы «Электронный дневник», его функциональность зависит от того, к какой группе принадлежит пользователь. Существует две привилегированных группы пользователей, доступных обычному человеку: группа «родители» и группа «учителя».

### 5.1. Интерфейс экранной формы «Родитель»

При входе в систему пользователь попадает на свою личную страничку, с которой осуществляется вся навигация по сайту (Рис. 5).

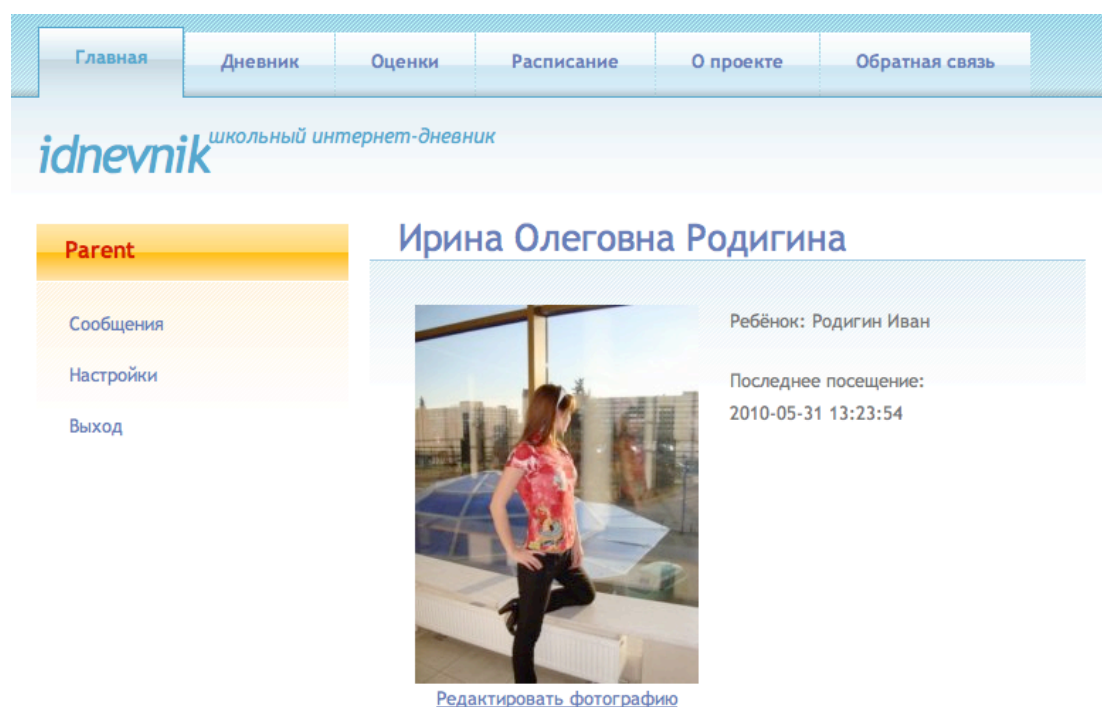


Рис. 5. Личная страница пользователя

Родитель имеет доступ к таким функциям, как просмотр оценок за определенный период в виде таблицы либо в форме

обычного классического дневника (Рис. 6).

Рис. 6. Оценки, представленные в виде классического дневника

Также ему доступно расписание, личные сообщения (Рис. 7), статистика по оценкам.

Рис. 7. Личные сообщения

## 5.2. Интерфейс экранной формы «Учитель»

Пользователи, принадлежащие группе «учителя» получают дополнительные возможности при пользовании системой «Электронный дневник».



Учитель имеет возможность просматривать оценки класса, классным руководителем которого он является и править их, используя удобный и простой интерфейс (Рис. 8).

**Классный журнал**

Редактировать: [оценки](#) [расписание](#) [учеников](#) [предметы](#)

№	Время	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
1	08:00-08:40	Математика	Химия <input type="text"/> <input type="button" value="OK"/> <input type="button" value="Отмена"/>	Химия	Астрономия	Астрономия	<input type="button" value="✗"/>
2	08:45-09:25	Астрономия	Химия	Химия	Астрономия	Астрономия	Математика <input type="button" value="✗"/>
3	09:35-10:15	География	Русский язык	Русский язык	География	Математика	Математика <input type="button" value="✗"/>
4	10:30-11:10	Химия	Русский язык	Русский язык			<input type="button" value="✗"/>
5	11:20-12:00		Математика				<input type="button" value="✗"/>

Рис. 8. Редактирование расписания

При нажатии на ячейку таблицы, ячейка превращается в выпадающий список (Рис. 9) с возможными вариантами выбора.

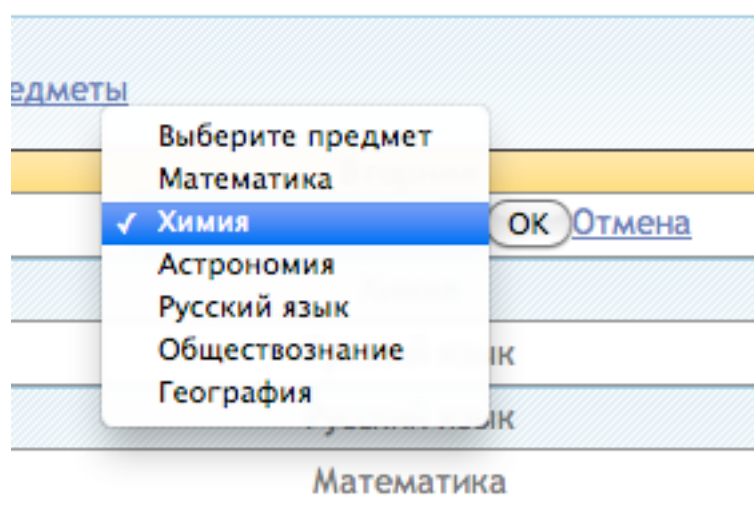


Рис. 9. Выпадающий список при редактировании

Таким образом исключается возможность ввода некорректных данных вследствие ошибки учителя – все данные, кроме основополагающих, выбираются учителем из выпадающих списков.

Данное решение является эффективным, поскольку оно удовлетворяет предпочтениям родителей по полноте и качеству предоставляемой информации и требованиям

учителей по простоте использования и интуитивно понятному интерфейсу. Школы же получают легко встраиваемый в их сайты веб-сервис, помогающий информировать родителей об успеваемости ребенка и школьных мероприятиях.

## 6. Заключение

В результате проделанной работы был реализован веб-сервис «Электронный дневник», предназначенный для формирования обратной связи «Школа-Родитель» и улучшения качества школьного образования.

Также была проведена работа со школами по выявлению уровня компьютерной грамотности среди школьных работников, показавшая очень низкий процент интернетизированных учителей – в среднем лишь 7% учителей имеют дома компьютер, подключенный к Интернету. Исходя из этих данных, при подключении школы к системе одним из условий является предоставление школой свободного доступа учителям к компьютерам, имеющим доступ к Интернету. Это не является проблемой для школ в связи с идущей программой по компьютеризации образовательных учреждений и поэтому все школы идут навстречу учителям, не подключенным к Интернету дома.

В процессе реализации проекта «Электронный дневник» были изучены способы атаки на Интернет-ресурсы и методы противодействия им, способы ускорения веб-проектов, нюансы работы UNIX-систем и настройки веб-сервера, работающего под их управлением.

Для обеспечения кроссплатформенности и эффективности веб-сервиса «Электронный дневник» были изучены стандарты веб-разработки и написания кода на следующих языках:

- HTML;

- CSS;
- PHP;
- Javascript.

Веб-сервис «Электронный дневник» в тестовом виде был опробован в реальном учебном заведении (МОУ СОШ №138, Челябинск), где способствовал улучшению успеваемости среди подключенных к системе учеников и показал высокую эффективность такого решения. Согласно собранной статистике, средняя успеваемость по подключенным к системе классам, выросла почти на 1 балл, увеличилось количество исправлений оценок на следующем после получения плохой оценки занятии.

С некоторыми школами заключены соглашения о дальнейшем сотрудничестве, ведется интеграция системы в Академической Гимназии при Санкт-Петербургском Государственном Университете.

## Список литературы

1. МГУшка.ру, образовательный портал, 2010. <http://mgushka.ru/>
2. 1dnevnik.ru, электронный дневник, 2010. <http://1dnevnik.ru/>
3. Дневник.ру, школьная социальная сеть, 2010. <http://dnevnik.ru/>
4. Баллов.нет, система электронных школьных журналов, 2010. <http://ballov.net/>
5. MySQL Reference Manual: History of MySQL, 2010. <http://dev.mysql.com/doc/refman/5.0/en/history.html>
6. InnoDB – Википедия, 2010. <http://ru.wikipedia.org/wiki/InnoDB>
7. PHP: History of PHP – Manual, 2010. <http://php.net/manual/en/history.php.php>
8. Консорциум Всемирной Паутины – Википедия, 2010. [http://ru.wikipedia.org/wiki/Консорциум\\_Всемирной\\_паутины](http://ru.wikipedia.org/wiki/Консорциум_Всемирной_паутины)
9. Учебник по AJAX и COMET, 2010. <http://javascript.ru/ajax>
10. Zend Framework и производительность – ZFConf, 2010. <http://www.zfconf.ru/2010/topics/performance-of-zend-framework-applications/>
11. Zend Framework и MVC, реализация модели – ZFConf, 2010. <http://www.zfconf.ru/2010/topics/zend-framework-and-mvc-fat-model/>
12. Настройка совместной работы Apache и nginx, 2010. <http://ssh3.livejournal.com/29252.html>
13. Зачем нужен nginx – Антон Южанинов, 2010. <http://ospf-ripe.livejournal.com/754.html>
14. Кристиан Дари, Богдан Бринзаре, Филип Черчерез-Тоза, Михай Бусика. AJAX и PHP. Разработка динамических веб-приложений. – СПб.: Символ-Плюс, 2007. – 336 стр.

15. Статистика по школьникам России, 2010.

<http://nsk.sibnovosti.ru/society/84065-kolichestvo-shkolnikov-v-rossii-za-10-let-sokratilos-na-10-mln>

16. Внедрение SQL-кода – Википедия, 2010.

[http://ru.wikipedia.org/wiki/Внедрение\\_SQL-кода](http://ru.wikipedia.org/wiki/Внедрение_SQL-кода)

17. Межсайтовый скриптинг – Википедия, 2010.

[http://ru.wikipedia.org/wiki/Межсайтовый\\_скриптинг](http://ru.wikipedia.org/wiki/Межсайтовый_скриптинг)