Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина: Модели данных и системы управления базами данных

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

ГЕНЕРАЦИЯ ИЗОБРАЖЕНИЙ СРЕДСТВАМИ KOHANA FRAMEWORK

Студент: гр. 052003

Веселов А.А.

Руководитель: ассистент кафедры информатики Степаньков Д.С.

Минск 2014

Содержание

[1. Введение. Kohana Framework 2](#_Toc385457707)

[2. Техническое задание 4](#_Toc385457708)

[3. База данных MySQL 6](#_Toc385457709)

[3.1. Технические возможности СУБД MySQL 7](#_Toc385457710)

[3.2. SQL 8](#_Toc385457711)

[3.3. Основные характеристики 9](#_Toc385457712)

[4. Реализация приложения 11](#_Toc385457713)

[4.1. Функциональная схема программы 11](#_Toc385457714)

[4.2. Структура таблиц базы данных 12](#_Toc385457715)

[4.3. Работа конструктора 13](#_Toc385457716)

[5. Дизайн приложения 17](#_Toc385457717)

[5.1. Frond-end 17](#_Toc385457718)

[5.2. Back-end 21](#_Toc385457719)

[6. Заключение 23](#_Toc385457720)

[Список литературы 24](#_Toc385457721)

# Введение. Kohana Framework

Движок сайта, это жаргонное выражение, правильное название – программный комплекс управления сайтом. Подобная система управления может состоять из множества небольших модулей (т.е. несколько маленьких программ), которые объединены в общую систему. Наиболее часто встречающаяся аббревиатура CMS (Content Management System) – Система Управления Контентом.

Движки, в подавляющем большинстве случаев, программируются на языке PHP, тем не менее, возможно применение других языков, всё зависит от программиста и возможностей хостинга. В любом случае, движок всегда работает в связке с базой данных MySQL (базы данных также должны поддерживаться хостером).

Принцип действия: Для начала, обратимся к работе обычного сайта. Как происходит открытие веб-страницы? В строку браузера вводится адрес и открывается соответствующая ему страница, из тех, которые лежат на сервере. Если адрес введён неправильно, то получим ошибку 404 (на сервере такая страница не обнаружена).

Как происходит открытие страницы на сайте, которой функционирует с помощью CMS? Страниц, как таковых, на сервере нет. Есть база данных, в которой находится текстовая информация и есть программа, которая работает с этой базой данный (тот самый движок). В строку браузера вводится адрес, в соответствии с адресом, программа «выдёргивает» из базы определённый кусок текста, вставляет его в заранее заданный дизайн и показывает посетителю. На все эти действия уходят доли секунды. Таким образом, страница будет генерироваться столько раз, сколько её будут запрашивать посетители. Итак, мы пришли к определениям статической и динамической страницы.

Статическая страница – находится на сервере в виде отдельного файла и отображается при каждом запросе.

Динамическая страница – фактически, её не существует, т.е. она собирается заново при каждом новом запросе посетителя сайта.

Kohana — это HMVC PHP5 веб фреймворк, предоставляющий средства для разработки web-приложений. Благодаря своей каскадной файловой системе, большому количеству встроенных утилит и последовательному API, он идеально подходит для решения большинства задач, а высокая скорость выполнения и обширные возможности профилирования позволяют ускорить разработку и использовать менее мощные ресурсы для содержания конечного продукта.

Почему именно Kohana?

* Высокая скорость выполнения
* Использование всех возможностей PHP5
* Большое количество встроенных инструментов, в том числе ORM и Database
* Малые системные требования
* Средства профилирования и отладки
* Поддержка UTF-8 и I18N

Что делает Kohana отличным фреймворком?

Расширяемость: с помощью удобной каскадной файловой системы вы можете полностью изменить функционал системы, простая установка: никаких инсталляторов и огромных конфигураций, а встроенный отладчик ошибок позволяет быстро отслеживать ошибки в приложении. Вы сможете легко профилировать, отлаживать код и находить узкие места в приложении используя стандартные утилиты.

А чтобы защитить приложение, Kohana предлагает такие утилиты как валидация введенных данных, защищенные от изменения cookie и защита от SQL-инъекций.

Официальная документация доступна только на английском языке: http://www.kohanaframework.org/guide/kohana/.

# Техническое задание

Основная задача веб-приложения – генерация изображений, созданных на клиенте, и загрузка их на сервер для полиграфической печати. Функциональность разрабатываемого программного комплекса должна включать в себя:

* Front-end (пользовательская часть):
  + Профиль пользователя (регистрация, авторизация, редактирование данных);
  + Генерация изображений, таких как:
    - Визитки;
    - Календари;
    - Приглашения (флаеры);
    - Керамика (тарелки и кружки);
  + Заказ сгенерированной продукции (корзина и workflow заказа);
  + Отзывы;
  + Статические страницы;
* Back-end (администраторская часть):
  + Управление разделами (добавление, редактирование, удаление);
  + Управление каталогами (добавление, редактирование, удаление);
  + Управление продукцией (добавление, редактирование, удаление);
  + Управления заказами (просмотр, управление, удаление);
  + Управление пользователями (создание, редактирование, удаление);
  + Управление скидками (создание, редактирование, удаление);
  + Управление способами оплаты (создание, редактирование, удаление);
  + Управление способами доставки (создание, редактирование, удаление);
  + Управление витриной (создание, редактирование, удаление);
  + Управление слайдером (создание, редактирование, удаление);
  + Управление новостями (создание, редактирование, удаление);
  + E-mail рассылка.
* Non-GUI функциональность:
  + Рассылка e-mail нотификаций при следующих действиях:
    - Регистрации пользователя;
    - Сброса пароля;
    - Изменения статуса заказа администратором/менеджером.

В приложении должны быть реализованы следующие роли пользователей:

1. Пользователь – доступ только к Front-end’у: редактирование и управление только своей информацией (учетной записью);
2. Менеджер – доступ к Back-end’у, кроме управления администраторами/менеджерами;
3. Администратор – полный доступ к Back-end’у.

Минимальные требования к серверному программному обеспечению:

* OS Linux (Debian/Gentoo/Ubuntu);
* Apache version 2.2.17 with OpenSSL;
* PHP VC6 version ThreadSafe 5.3.5;
* MySQL Community Server version 5.5.x.

Веб-приложение должно поддерживать следующий список браузеров:

* Internet Explorer 10;
* Google Chrome 34;
* Mozilla Firefox 28.

# База данных MySQL

База данных представляет собой структурированную совокупность данных. Эти данные могут быть любыми - от простого списка предстоящих покупок до перечня экспонатов картинной галереи или огромного количества информации в корпоративной сети. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, каковой и является ПО MySQL. Поскольку компьютеры замечательно справляются с обработкой больших объемов данных, управление базами данных играет центральную роль в вычислениях. Реализовано такое управление может быть по-разному - как в виде отдельных утилит, так и в виде кода, входящего в состав других приложений.

MySQL - это система управления реляционными базами данных. В реляционной базе данных данные хранятся не все скопом, а в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

MySQL - это ПО с открытым кодом. Применять его и модифицировать может любой желающий. Такое ПО можно получать по Internet и использовать бесплатно. При этом каждый пользователь может изучить исходный код и изменить его в соответствии со своими потребностями.

Почему веб-программисты отдают предпочтение СУБД MySQL? MySQL является очень быстрым, надежным и легким в использовании. Если вам требуются именно эти качества, попробуйте поработать с данным сервером. MySQL обладает также рядом удобных возможностей, разработанных в тесном контакте с пользователями. Первоначально сервер MySQL разрабатывался для управления большими базами данных с целью обеспечить более высокую скорость работы по сравнению с существующими на тот момент аналогами. И вот уже в течение нескольких лет данный сервер успешно используется в условиях промышленной эксплуатации с высокими требованиями. Несмотря на то что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Internet.

## Технические возможности СУБД MySQL

MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API). Мы также поставляем сервер MySQL в виде многопоточной библиотеки, которую можно подключить к пользовательскому приложению и получить компактный, более быстрый и легкий в управлении продукт. Доступно также большое количество программного обеспечения для MySQL, в большей части - бесплатного.

MySQL правильно произносится как ''Май Эс Кью Эль''.

## SQL

SQL состоит из двух частей: серверной и клиентской.

Сервер MySQL постоянно работает на компьютере. Клиентские программы (например, скрипты PHP) посылают серверу MySQL SQL-запросы через механизм сокетов (то есть при помощи сетевых средств), сервер их обрабатывает и запоминает результат. То есть скрипт (клиент) указывает, какую информацию он хочет получить от сервера баз данных. Затем сервер баз данных посылает ответ (результат) клиенту (скрипту).

Почему всегда передается не весь результат? Очень просто: дело в том, что размер результирующего набора данных может быть слишком большим, и на его передачу по сети уйдет чересчур много времени. Да и редко когда бывает нужно получать сразу весь вывод запроса (то есть все записи, удовлетворяющие выражению запроса). Например, нам может потребоваться лишь подсчитать, сколько записей удовлетворяет тому или иному условию, или же выбрать из данных только первые 10 записей. Механизм использования сокетов подразумевает технологию клиент-сервер, а это означает, что в системе должна быть запущена специальная программа — MySQL-сервер, которая принимает и обрабатывает запросы от программ. Так как вся работа происходит в действительности на одной машине, накладные расходы по работе с сетевыми средствами незначительны (установка и поддержание соединения с MySQL-сервером обходится довольно дешево).

Структура MySQL трехуровневая: базы данных — таблицы — записи. Базы данных и таблицы MySQL физически представляются файлами с расширениями frm, MYD, MYI. Логически - таблица представляет собой совокупность записей. А записи - это совокупность полей разного типа. Имя базы данных MySQL уникально в пределах системы, а таблицы - в пределах базы данных, поля - в пределах таблицы. Один сервер MySQL может поддерживать сразу несколько баз данных, доступ к которым может разграничиваться логином и паролем. Зная эти логин и пароль, можно работать с конкретной базой данных. Например, можно создать или удалить в ней таблицу, добавить записи и т. д. Обычно имя-идентификатор и пароль назначаются хостинг провайдерами, которые и обеспечивают поддержку MySQL для своих пользователей.

## Основные характеристики

* MySQL - это система управления реляционными базами данных.

В реляционной базе данные хранятся в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

* Программное обеспечение MySQL - это ПО с открытым кодом.

ПО с открытым кодом означает, что применять и модифицировать его может любой желающий. Такое ПО можно получать по Internet и использовать бесплатно. При этом каждый пользователь может изучить исходный код и изменить его в соответствии со своими потребностями.

* Технические возможности СУБД MySQL

ПО MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).

* Безопасность

Система безопасности основана на привилегиях и паролях с возможностью верификации с удаленного компьютера, за счет чего обеспечивается гибкость и безопасность. Пароли при передаче по сети при соединении с сервером шифруются. Клиенты могут соединяться с MySQL, используя сокеты TCP/IP, сокеты Unix или именованные каналы (named pipes, под NT)

* Вместимость данных

Начиная с MySQL версии 3.23, где используется новый тип таблиц, максимальный размер таблицы доведен до 8 миллионов терабайт (263 bytes). Однако следует заметить, что операционные системы имеют свои собственные ограничения по размерам файлов. Ниже приведено несколько примеров:

- 32-разрядная Linux-Intel – размер таблицы 4 Гб.

- Solaris 2.7 Intel - 4 Гб

- Solaris 2.7 UltraSPARC - 512 Гб

- WindowsXP - 4 Гб

Как можно видеть, размер таблицы в базе данных MySQL обычно лимитируется операционной системой. По умолчанию MySQL-таблицы имеют максимальный размер около 4 Гб. Для любой таблицы можно проверить/определить ее максимальный размер с помощью команд SHOW TABLE STATUS или myisamchk -dv table\_name. Если большая таблица предназначена только для чтения, можно воспользоваться myisampack, чтобы слить несколько таблиц в одну и сжать ее. Обычно myisampack ужимает таблицу по крайней мере на 50%, поэтому в результате можно получить очень большие таблицы.

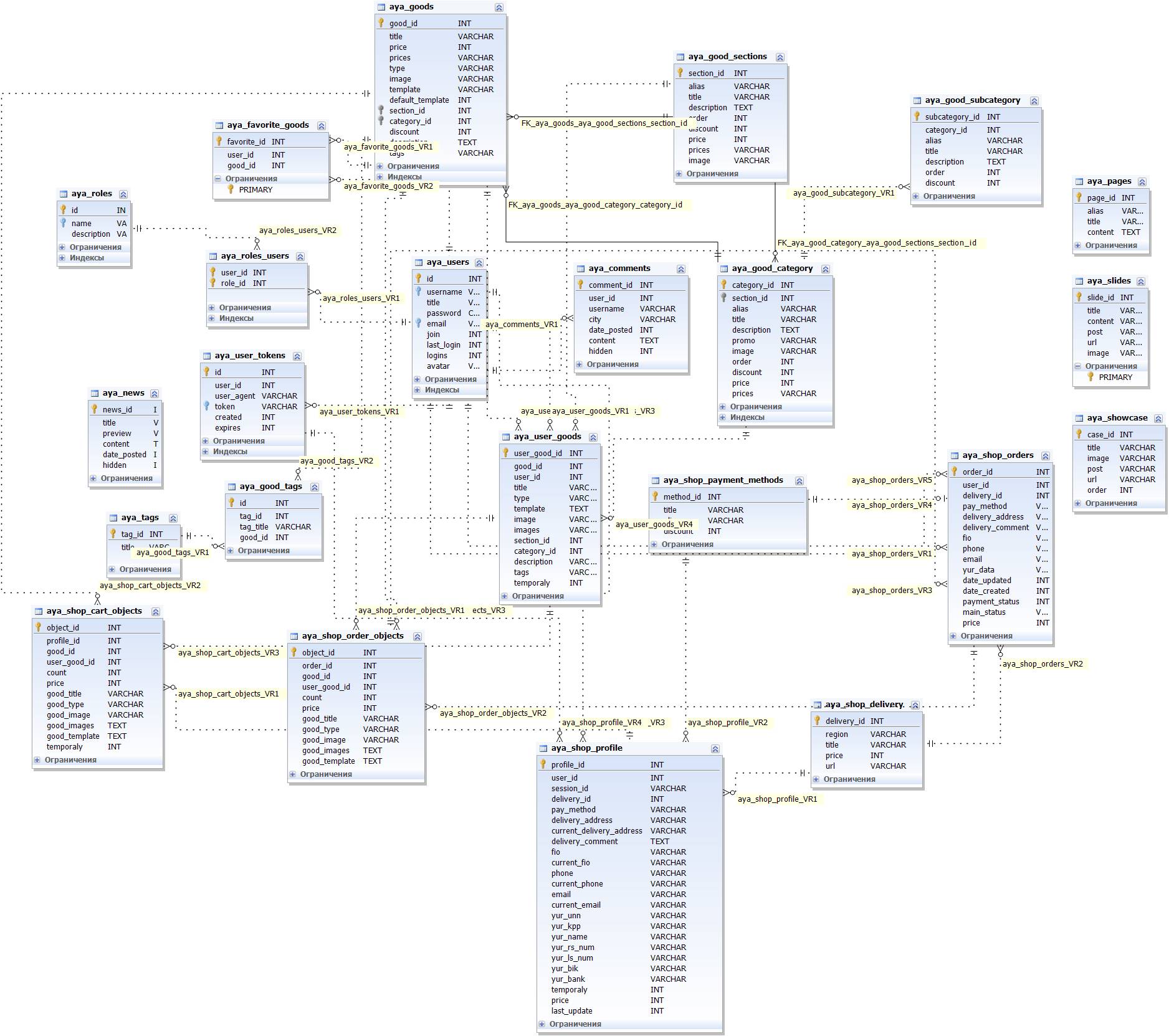
# Реализация приложения

## Функциональная схема программы

Приложение будет иметь следующую функциональную схему:

## Структура таблиц базы данных

Для реализации веб-приложения была разработана следующая структура базы данных (таблицы и набор полей):



## Работа конструктора

Для работы конструктора используются js библиотеки KineticJS, AngularJS, PreloadJS. AngularJS используется для управления формами и окнами конструктора, а KineticJS занимается отрисовкой всего, что происходит внутри главного окна конструктора. PreloadJS управляет загрузкой и хранением ресурсов для работы конструктора.

Конструктор визуально делится на два окна — менеджер слоя и окно холста. Информация о слоях содержится в массиве, который AngularJS использует для отображения через директиву ng-repeat. Напрямую объекты не берутся из KineticJS, а переносятся в отдельный контейнер в виде массива объектов. Это сделано для упрощения работы с информацией, которая отображается в менеджере слоя.

Работа конструктора начинается с инициализации рабочей схемы шаблона. Схема представляет из себя описание всех слоев шаблона в виде объекта, содержащего информацию о размерах слоя, названии и о предустановленных объектах, которые нужны для представления шаблона пользователю. Один из таких объектов может быть задний фон в виде изображения кружки или футболки.

После загрузки схемы осуществляется создание объектов на слоях. Предварительно загружаются картинки при помощи PreloadJS. Все ресурсы хранятся в отдельном массиве и доступ к ним осуществляется при помощи соответствующих методов. Работа с ресурсами через PreloadJS значительно оптимизирует работу системы. Таким образом система защищает себя от нежелательных дефектов на макете в виде незагруженных или наполовину загруженных изображений или шрифтов. Несвоевременная генерация готовых изображений конструктором по макету, когда не все ресурсы загружены, может вызвать краш браузера. Работа с PreloadJS эти моменты исключает.

После загрузки объектов на макете пользователь может начать работу в конструкторе с макетом. На макет можно нанести объекты в виде текста, изображения. Для каждого вида объектов используются свои характеристики, которые можно менять в модальном окне. Окно вызывается при нажатии на нужную кнопку в менеджере слоя. Каждый объект отрисовывается при помощи KineticJS на холсте, который представляет из себя canvas. Объекты можно растягивать по 8 направлениям за специальные уголки-контроллеры. Каждый контроллер обрабатывается своим методом, который высчитывает координаты объекта и его размеры. Координаты берутся исходя положения левого верхнего угла активного объекта. Активный объект кроме контроллеров имеет специальную подсветку для выделения и лучшего восприятия.

После каждого действия над слоем макета необходимо его отрисовать. Предыдущий вариант рендера слоя всегда хранится в кеше и к нему можно получить доступ. Кеш имеет не только слой, но и его объекты. Манипуляция кешем легко оптимизирует операция перемещения объекта по холсту в системе. Также задействована аналогичная система при рендере всего слоя. Это помогает избежать пролагов при смене активного слоя. К сожалению, система кеша может сильно понизить стабильность конструктора на мобильных устройствах. При очень большом объеме объектов на слоях и при большом количестве слоев уходит очень много оперативной памяти на хранение кеша. Из-за этого есть необходимость чистить кеш всего холста и слоев, чтобы сократить потребление памяти. Особенно это актуально на мобильных устройствах, когда используется макет типа «настольный календарь», где используется 13 слоев. Хранение в системе такого количества кеша в виде картинок 10000х10000 пикселей без сжатия становиться очень проблемным моментов. Поэтому необходимо чистить кеш неактивных слоев, но не объектов, которые находятся на этих слоях.

Генерация изображения слоя макета осуществляется методом, который наследуется методом объекта слоя из KineticJS. Фактически, все объекты слоя отрисовываются отдельно и накладываются на генерируемое изображения в порядке, который указывается в свойствах объекта. После этого генерируемое изображение усекается по координатам и размеру рабочего пространства, который является особым объектов слоя. Этот объект создается при загрузке схемы макета. Полученное изображение после генерации ужимается согласно формату, который установлен в настройках конструктора, и отправляется на сервер. Во время загрузки блокируется вся система конструктора, чтобы пользователь не смог навредить процессу какой-нибудь манипуляцией с макетом. После загрузки сервер возвращается объект, содержащий ссылку на изображение.

Превью генерируется при помощи следующего алгоритма:

1. Убирается вся ненужная дополнительная графика (уголки-контроллеры, подсветка);
2. Слой центрируется и скалируется до соотношения 1:1;
3. Генерируется изображение;
4. Отсекается ненужное согласно рабочей области;
5. Конвертируется в рабочий формат’
6. Изображение загружается на сервер с передачей параметра в url, который помечает изображение как превью;
7. Сервер отдает ссылку загруженного изображения, которое было уменьшено до размеров превью;
8. Ссылка изображения помещается в поле формы вне конструктора, которое хранит в себе ссылку на превью;
9. Слой скалируется до соотношения, которое было до генерации макета. Возвращается дополнительная графика активного объекта.

У генерации всего макета алгоритм куда сложнее:

1. Вызывается функция inner\_gen, в которой указывается номер слоя и количество слоев. Внутри функции выполняются действия с 2 по 9;
2. Убирается вся ненужная дополнительная графика (уголки-контроллеры, подсветка);
3. Скрываются объекты, которые не должны находится на конечном изображении (например, изображения футболки или кружки);
4. Генерируется изображение;
5. Отсекается ненужное согласно рабочей области;
6. Конвертируется в рабочий формат;
7. Изображение загружается на сервер с передачей параметра в url, который указывает id изображения;
8. Создается callback, который запускается после загрузки изображения. В нем хранится функция inner\_gen с идентификатором следующего слоя по списку. Если слой последний, то вызывается функция генерации превью;
9. Слой скалируется до соотношения, которое было до генерации макета. Возвращается дополнительная графика активного объекта;
10. Генерируется превью по алгоритму, описанному выше;
11. Генерируется объект, который содержит в себе массив всех изображений, названия слоев, к которому привязаны изображений.

Как можно понять из алгоритма генерации макета, изображения грузятся в асинхронном режиме последовательно, вызывая нужную функцию после завершения загрузки очередного изображения. Это нужно для исключения момента, когда одновременно грузятся все изображения. Далеко не все сервера могут позволить такую загрузку файлов. К тому же, браузер мобильных устройств может зависнуть при такой нагрузке. После генерации всего макета система в асинхронном режиме передает объект по нужного адресу и перенаправляет на страницу собранного товара.

Сохранение и загрузка самого макета осуществляется при помощи специального объекта, который грузится в поле скрытой формы вне конструктора. Объект представляет из себя массив отдельных объектов, который содержат идентификатор слоев и список объектов самого слоя. Объекты слоя хранятся в виде небольшого набора переменных типа координат, размеров, ссылок на изображения или названия и размера шрифта.

KineticJS поддерживает загрузку сериализированного холста. Однако хранить такие данные было бы нелогичным, так как большое количество «мусора» помешало бы быстрой загрузке макета и может вызвать проблемы с макетом при изменение конфигурации схемы. Поэтому для хранения макета используется свой вариант сериализации, который «подхватывает» только нужные атрибуты объектов.

Конструктор позволяет вносить изменения в макеты согласно определенному сценарию. Например, при работе с типом макета «настольный календарь» при загрузке конструктора вызывается диалоговое окно, которое позволяет выбрать месяц, с которого будет проходить отсчет календаря. После выбора месяца запускается сценарий, который перебирает слои, начиная со второго (первый слой содержит обложку календаря), и создает на них картинки с содержанием месяца и переименовывает слои, начиная с выбранного.

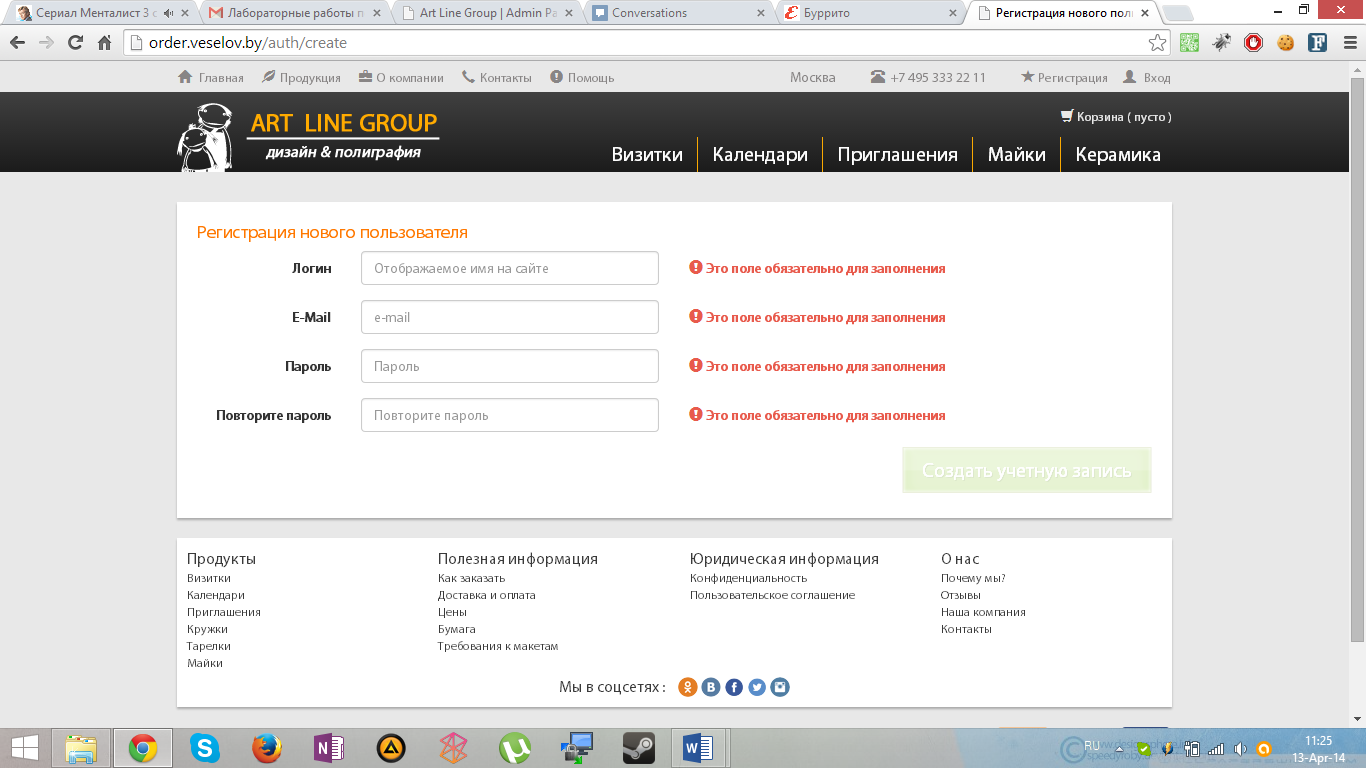
Генерация и все операции с макетом осуществляется на стороне клиента в виде JS. Такой подход помогает ускорить отзывчивость всех операций над макетом и куда проще в исполнении, чем серверный. Такая организация работы исключает возможные проблемы при изменении параметров объектов на макете, различные ошибки на уровне работы интерпретатора выбранного языка для работы проекта. К тому же, далеко не все хостинги смогут работать с конструктором, базирующемся на сервере.

# Дизайн приложения

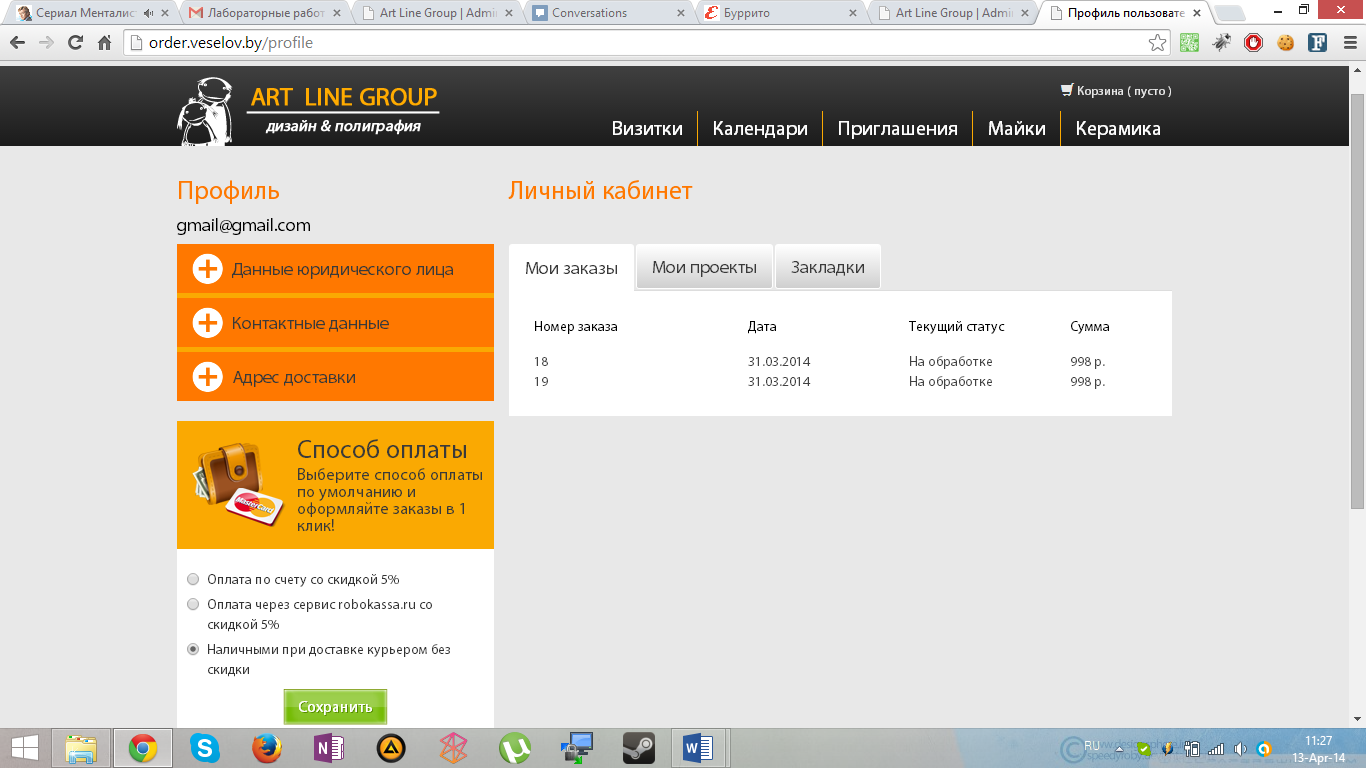
## Frond-end

Для клиентской части изначально был отрисоват макет в Photoshop’е с набором основных элементов, а недостающие элементы было решено взять из CSS-фреймворка Twitter Bootstrap 3. Ниже приведены скриншоты отдельно взятых страниц:

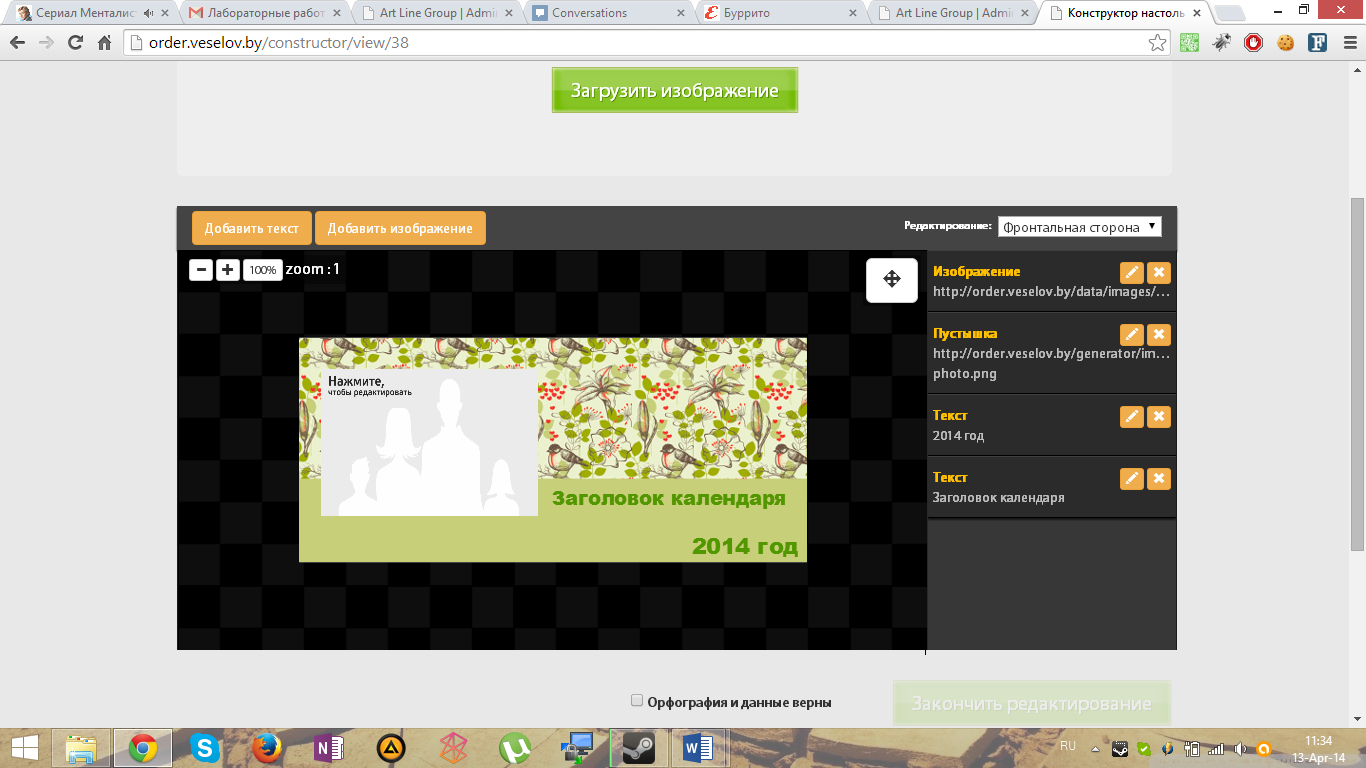
* Регистрация пользователя:



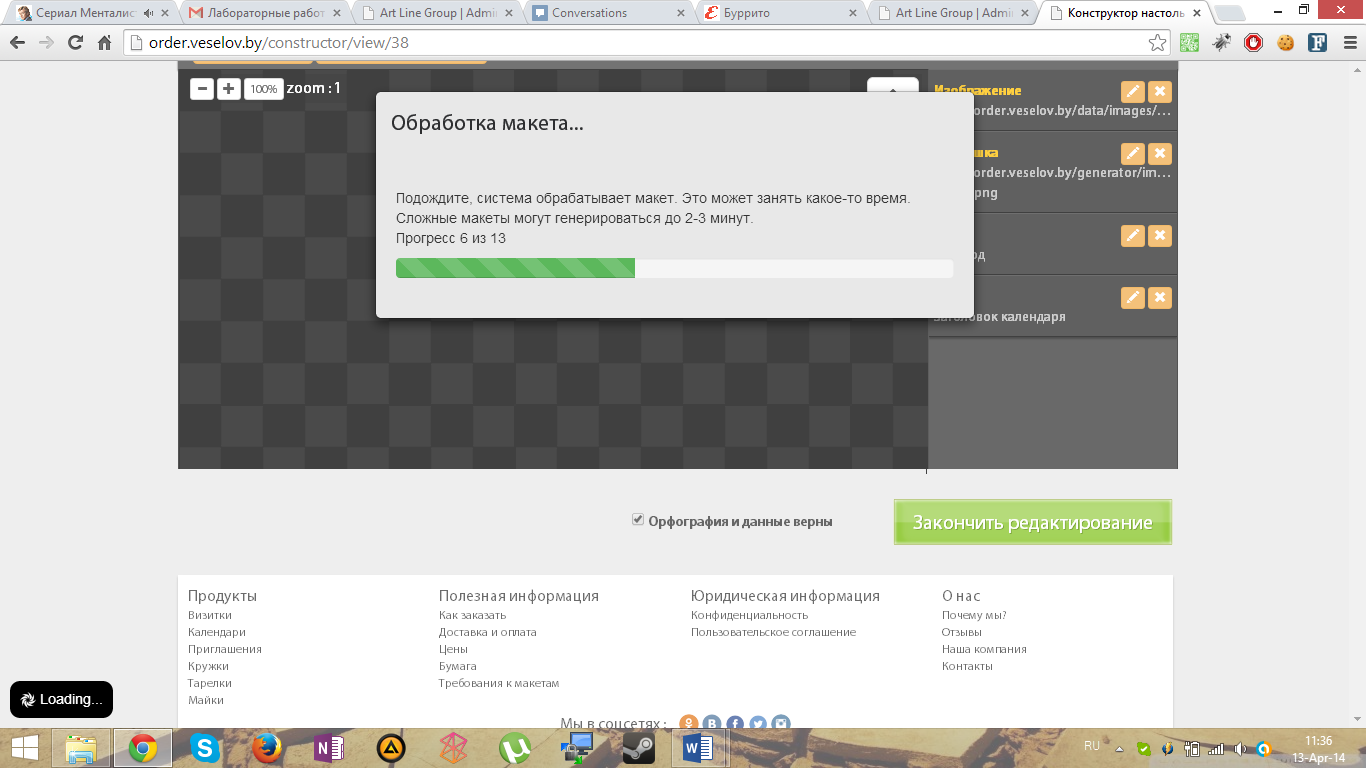
* Профиль пользователя:



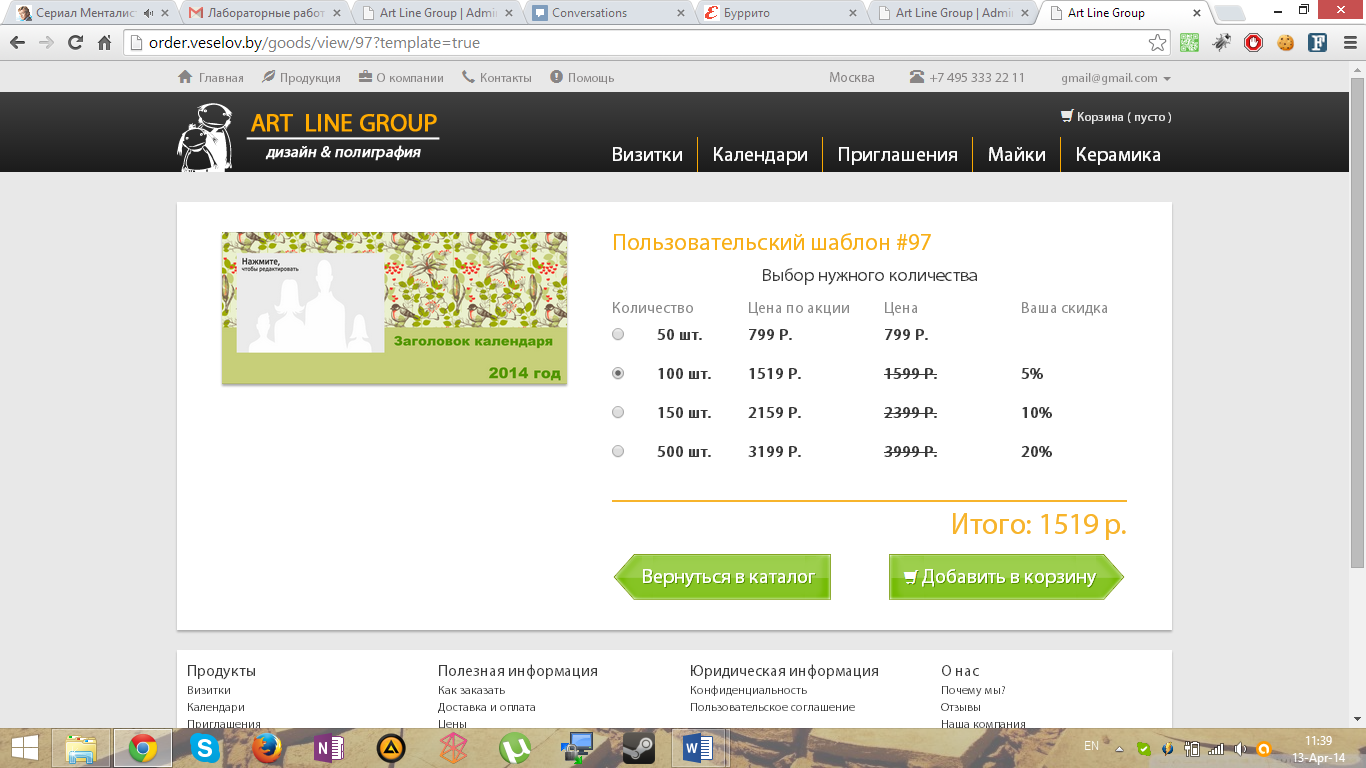
* Workflow заказа товара (настольного календаря)
  + Страница генерации календаря:



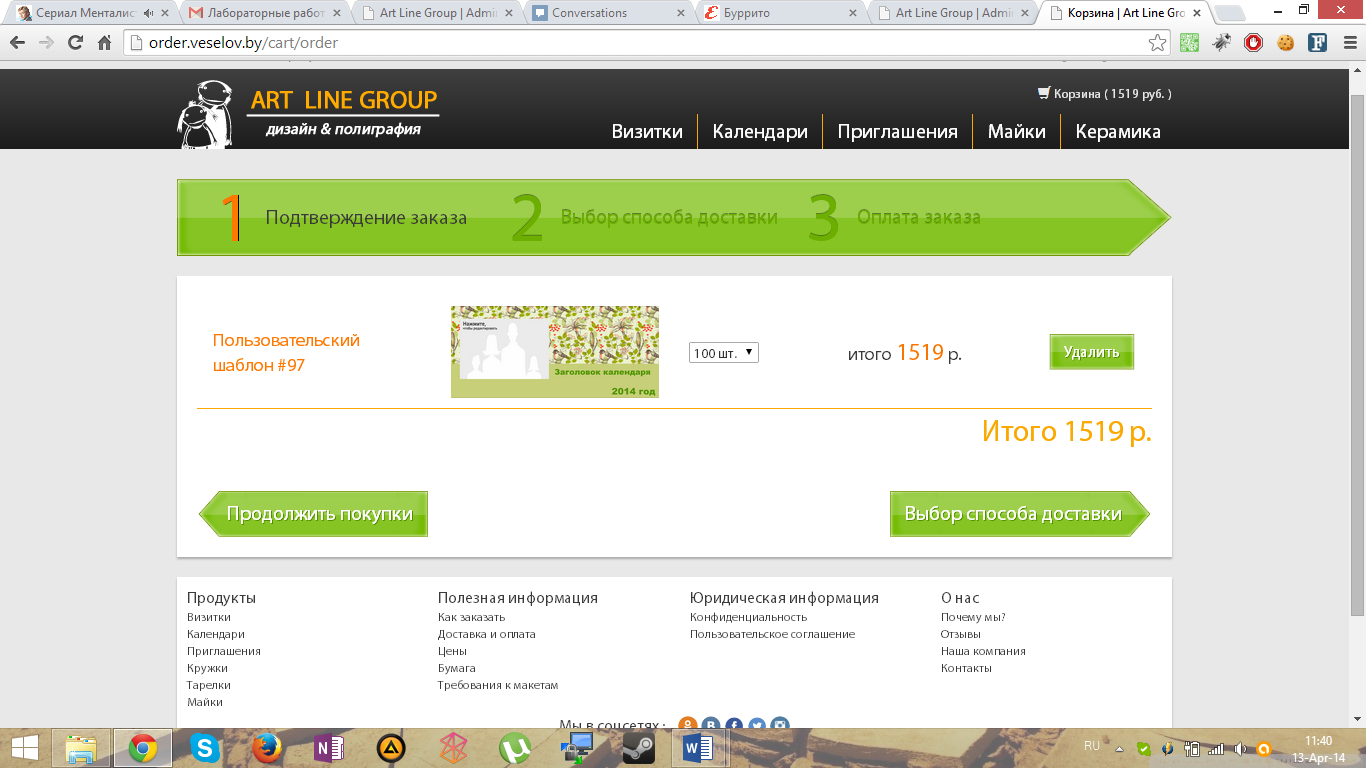
* + Процесс генерации макетов:



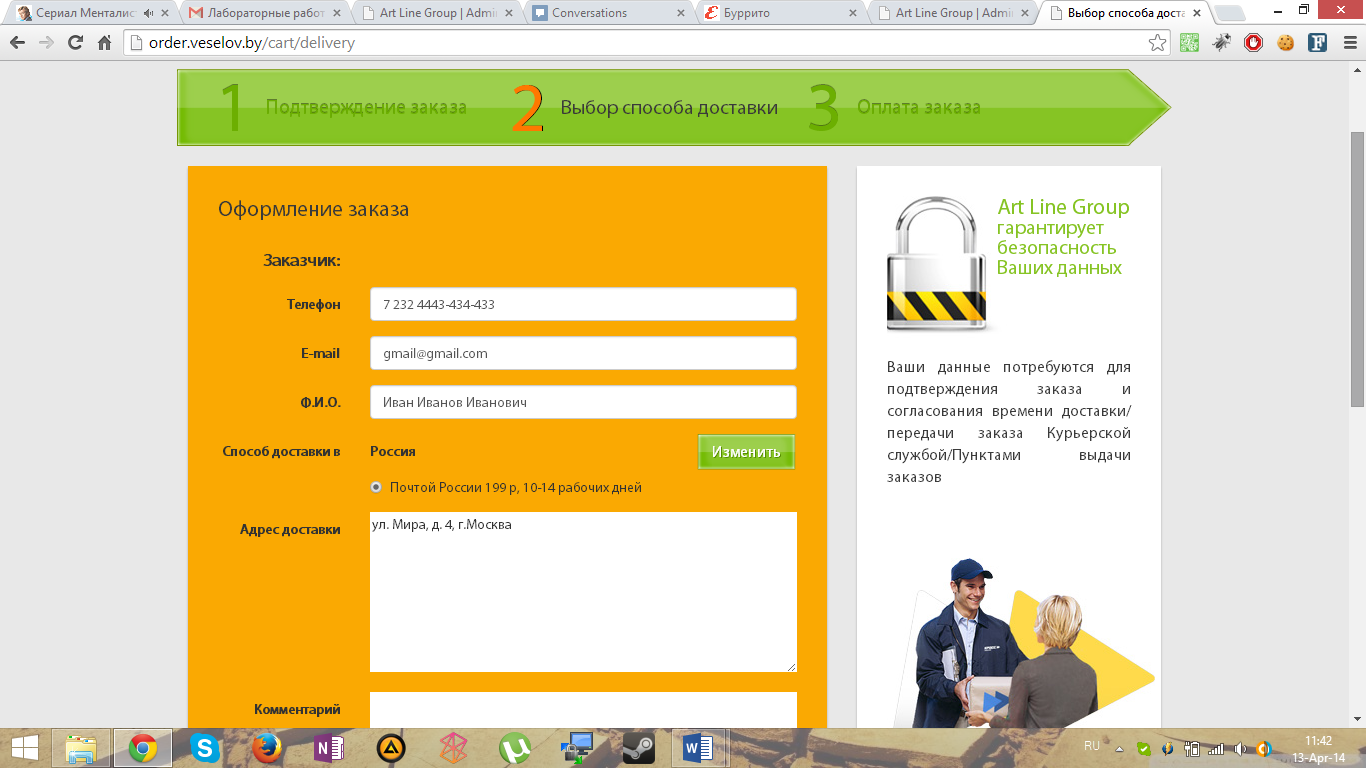
* + Выбор количества товаров:



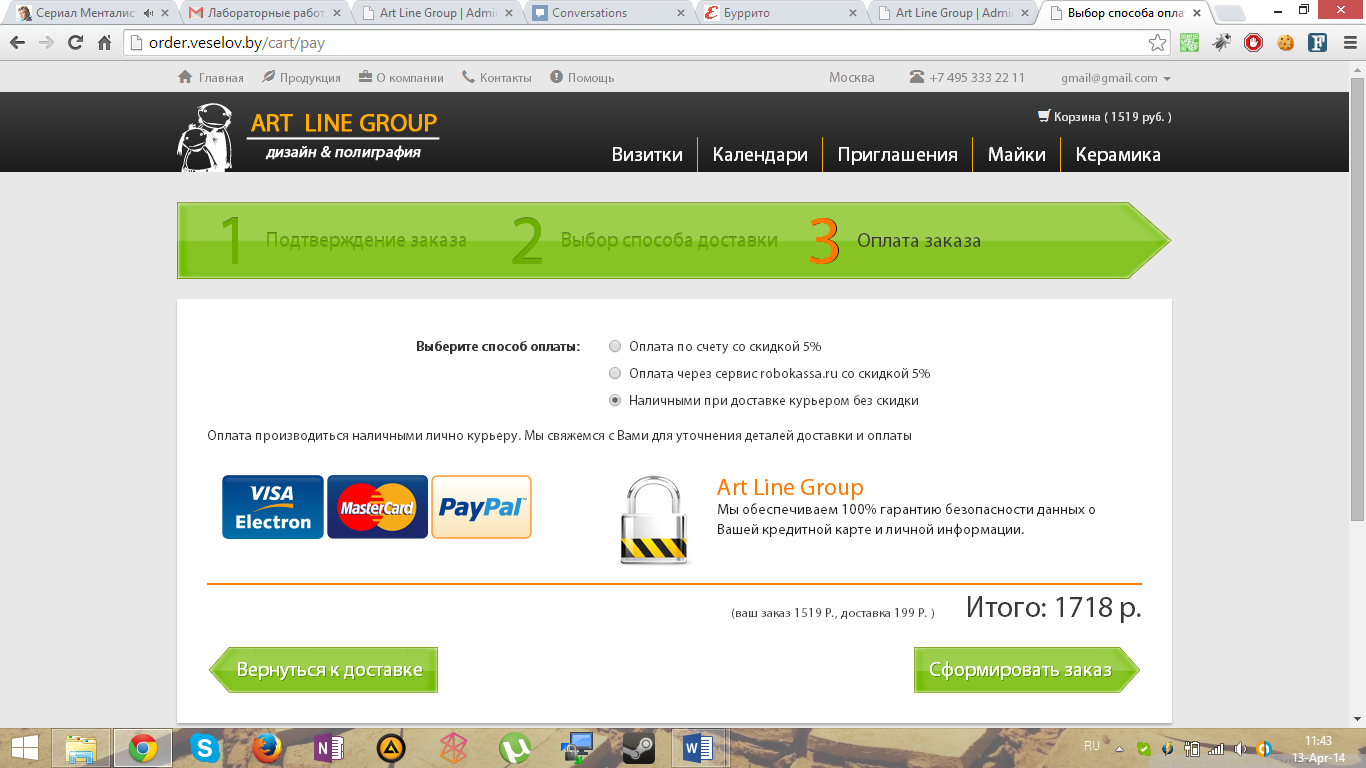
* + Оформление товара в корзине:



* + Выбор способа доставки:



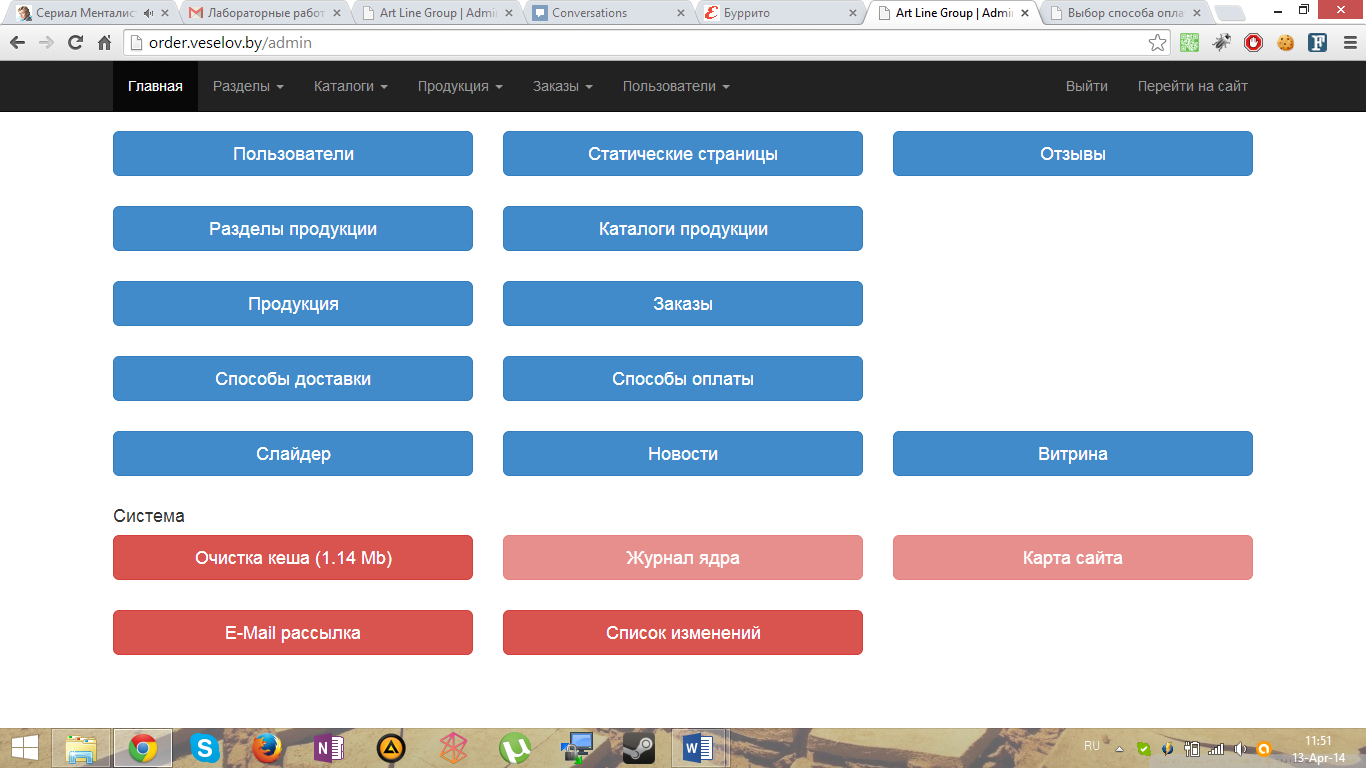
* + Оплата заказа:



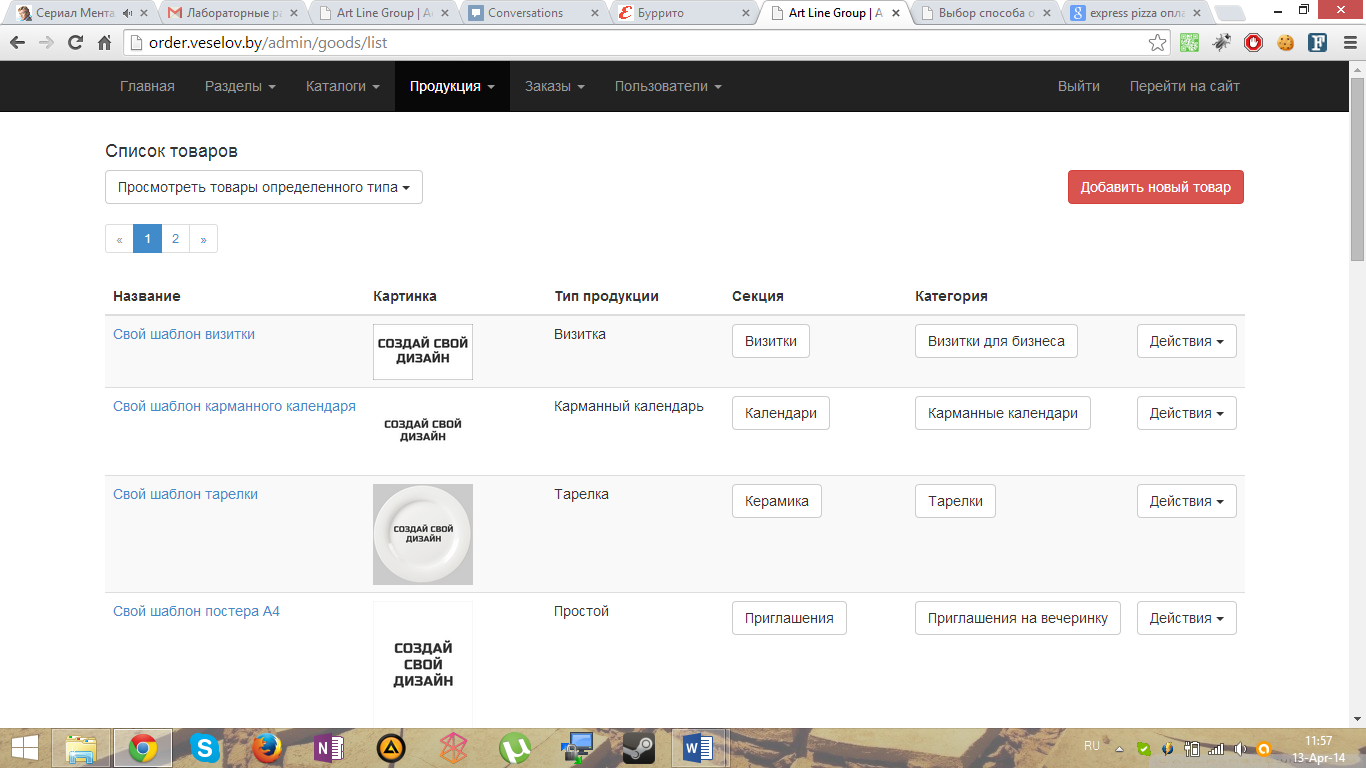
## Back-end

Для администраторской части используется CSS-фрейморк Twitter Bootstrap 3. Ниже представлены некоторые страницы:

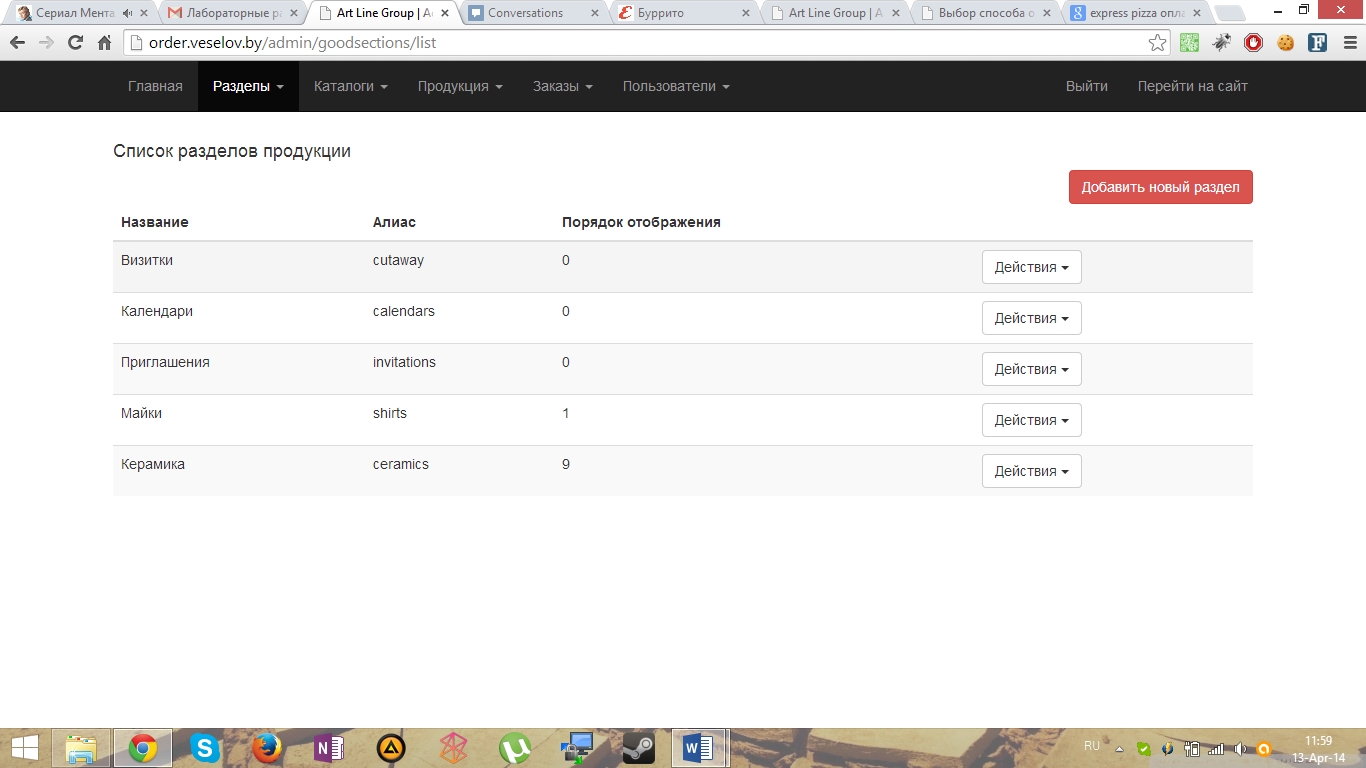
* Домашняя страница:



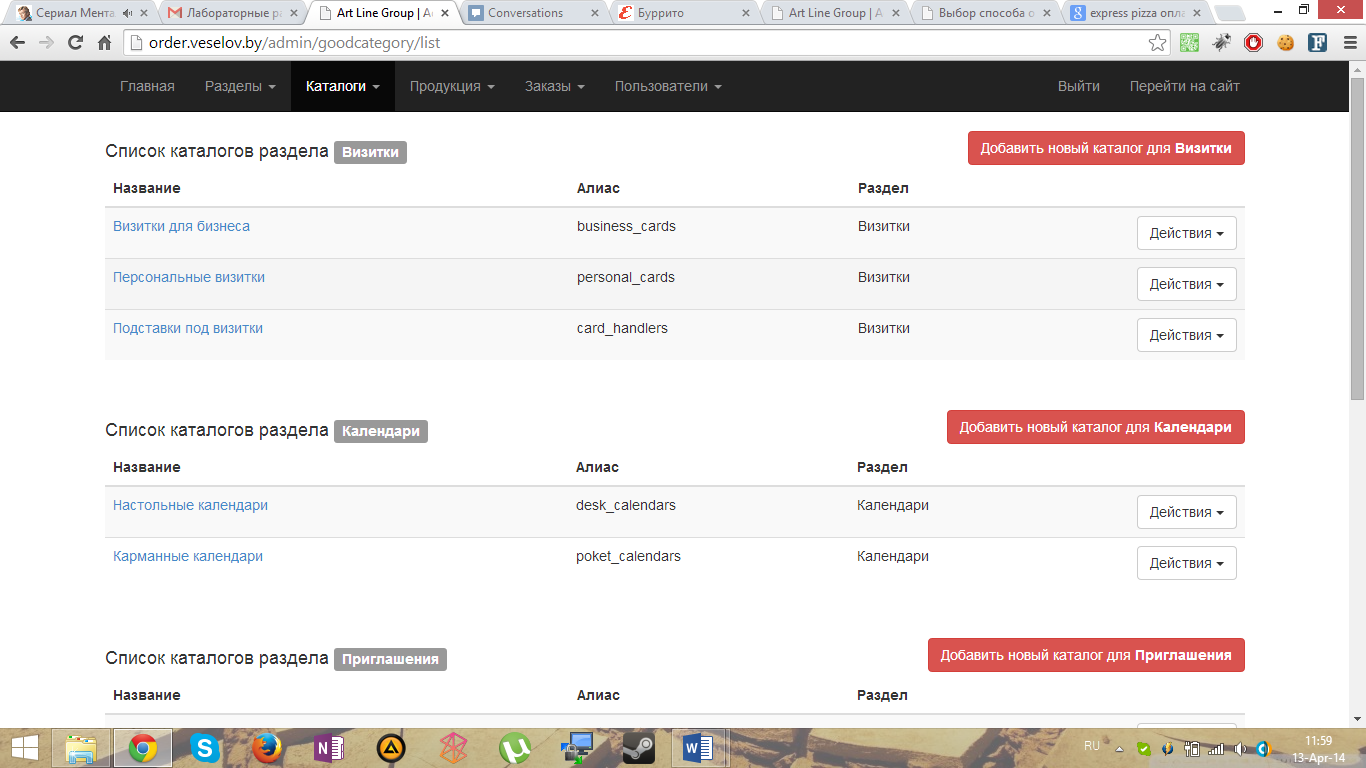
* Управление продукцией:



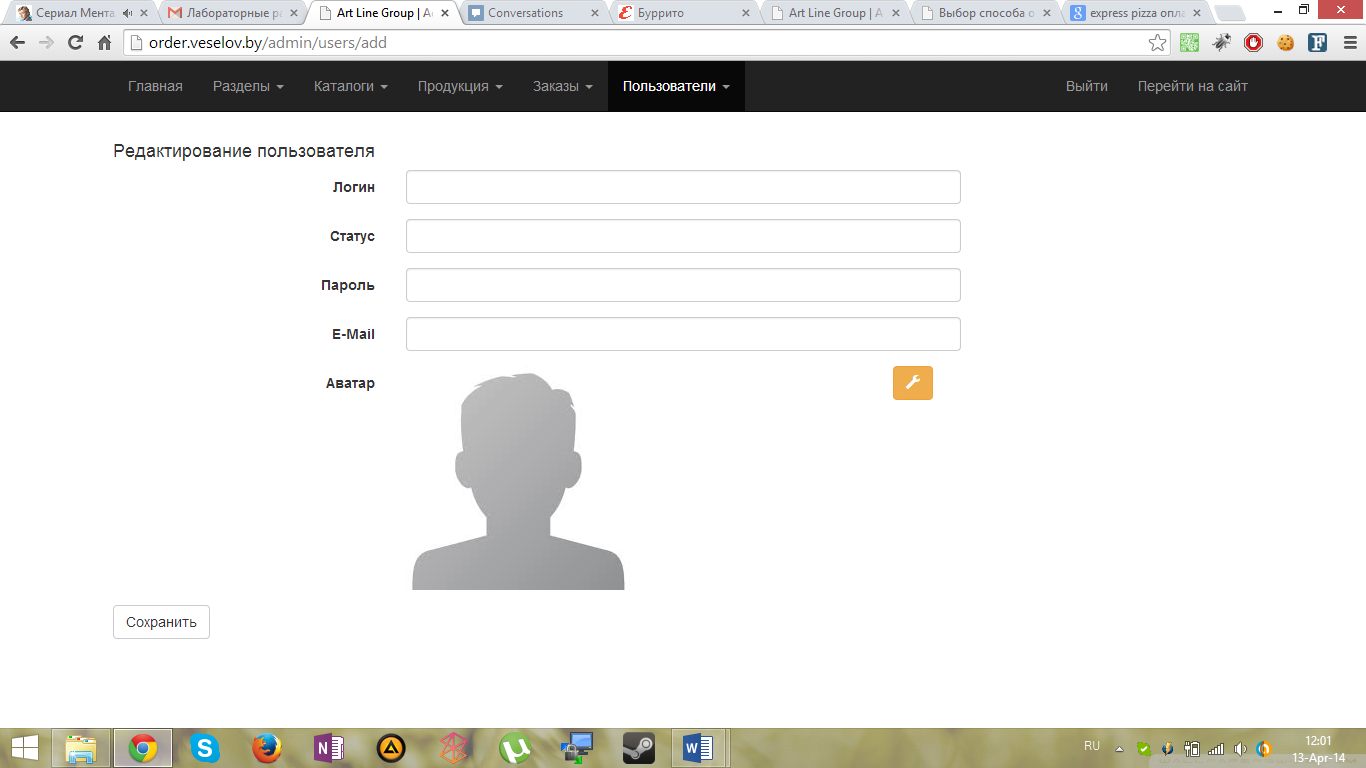
* Управление разделами:



* Управление каталогами:



* Добавление нового пользователя:



# Заключение

В результате выполнения курсового проекта были изучены такие технологии, как:

* СУБД MySQL;
* PHP-фреймворк Kohana;
* CSS-фрейморк Twitter Bootstrap 3;
* Веб-сервер Apache.

Результатом выполнения стало веб-приложения, в полной мере отвечающее техническому заданию (глава 2 – «Техническое задание»). Полученный программный продукт обладает возможностью генерации ряда приложений при графическом редактировании пользователем, после чего полученный изображения можно распечатать в полиграфии и получить готовы визитки, календари либо же при использовании специализированной печати – майки и кружки.

# Список литературы

1. Jason D. Straughan. Kohana 3.0. Beginner's Guide – 2011г.
2. <http://kohanaframework.org/>
3. <http://ru.wikipedia.org/wiki/Kohana>
4. <http://habrahabr.ru/hub/kohanaphp/>
5. <http://github.com>
6. <http://www.apache.org/>
7. <http://www.php.net/>
8. <http://www.mysql.com/>