Министерство образования Республики Беларусь

### Учреждение образования «[Институт информационных технологий](https://www.google.by/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&sqi=2&ved=0CDcQFjAC&url=http%3A%2F%2Fwww.iit-bsuir.by%2F&ei=unOvUpDmGemB4gSZ3IGQDQ&usg=AFQjCNHEr4the3QhkSSjmxbzcNJBZi5-Tg&sig2=GxV3hq7_8t34Csduk0HElg&bvm=bv.57967247,d.bGE&cad=rja) Белорусского государственного университета информатики и радиоэлектроники»

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Дисциплина: Современные платформы программирования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

«Ювелирная мастерская»

БГУИР КП 1-40 01 01 5810027 ПЗ

Студент: гр.581061 Фут Д.С.

Руководитель: Бакунова О.М.

Минск 2017

**ВВЕДЕНИЕ**

Основной задачей этого курсового проекта является разработка программного средства для ювелирной мастерской. Данное программное средство должно помогать администраторам и менеджерам ювелирной мастерской контролировать процесс работы в мастерской.

Мастерская работает с определенными материалами (платина, золото, серебро, различные драгоценные камни и т.д.). При обращении в мастерскую потенциального клиента, менеджер определяется с тем, какое именно изделие необходимо клиенту. Все изготавливаемые изделия принадлежат к некоторому типу (серьги, кольца, броши, браслеты), бывают выполнены из определенного материала и имеют цену (включающую стоимость материалов и работы).

**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

**1.1 Анализ существующих аналогов**

На данный момент времени существует множество аналогов программных средств по автоматизации рабочих мест сотрудников.

К основным недостаткам большинства таких программ можно отнести то, что для каждого рабочего места требуются свои функции, в зависимости от потребностей заказчика, соответственно программа, подходящая одному, не всегда может подойти другому, исходя из вышеизложенного причина создания программного средства обусловлена индивидуальными требованиями заказчика и невозможностью найти аналог под такие требования.

**1.2 Постановка задачи**

Анализ литературы и предметной области показали, что реализуемое программное средство имеет большое количество аналогов, но каждый из которых имеет свои недостатки.

Основной целью является – разработка программного средства для автоматизации рабочего места администратора ювелирной мастерской.

В данном программном средстве необходимо реализовать:

* Учет клиентов мастерской. Для каждого клиента необходимо хранить: Имя, Фамилию, Отчество, Телефон, Дату регистрации в системе.
* Учет услуг, оказанных клиентам.
* Каталог стрижек, хранящий в себе название стрижки, пол и цену.

**1.3 Входные данные**

Входной информацией является:

* Информация о клиенте;
* Информация о заказе;
* Каталог стрижек;

**1.4 Выходные данные**

* Список всех клиентов;
* Список оказанных услуг и их стоимость;

**2 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА**

**2.1 Разработка спецификаций требований**

Программа будет разрабатываться, и функционировать в операционной системе Windows 10.

Windows 10 – пользовательская операционная система семейства Windows NT, следующая по времени выхода за Windows 8.1 и являющаяся последней версией операционных систем данного семейства.

По данным веб-аналитики на сентябрь 2015 года Windows 10 была установлена более чем на 100 миллионов устройств по всему миру.

Для нормального функционирования проекта достаточно иметь персональный компьютер с операционной системой Windows 10.

Минимальные требования:

* + - процессор: Intel Pentium II 400 MHz или аналогичный;
    - оперативная память: 512 Mб;
    - видеокарта: объем 256 Mб;
    - операционная система: Windows 10;
    - устройства ввода: клавиатура, мышь;
    - устройство вывода: монитор;
    - Подключение к сети интернет.

Курсовой проект может работать на таких операционных системах как:

Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10.

**2.2 Разработка архитектуры программного средства**

Программное средство должно будет хранить все имеющуюся информацию в базе данных.

MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Веб-интерфейс — это совокупность средств, при помощи которых пользователь взаимодействует с веб-сайтом или любым другим приложением через браузер. Веб-интерфейсы получили широкое распространение в связи с ростом популярности всемирной паутины и соответственно — повсеместного распространения веб-браузеров.

Интерфейс курсового проекта будет представлен с помощью языка разметки XML.

**2.3 Обоснование выбора языка программирования**

NetBeans IDE — свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и ряда других.

Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведётся независимым сообществом разработчиков-энтузиастов (NetBeans Community) и компанией NetBeans Org. Последние версии NetBeans IDE поддерживают рефакторинг, профилирование, выделение синтаксических конструкций цветом, автодополнение набираемых конструкций на лету и множество предопределённых шаблонов кода.

Для разработки программ в среде NetBeans и для успешной инсталляции и работы самой среды NetBeans должен быть предварительно установлен Sun JDK или J2EE SDK подходящей версии. Среда разработки NetBeans по умолчанию поддерживала разработку для платформ J2SE и J2EE. Начиная с версии 6.0 NetBeans поддерживает разработку для мобильных платформ J2ME, C++ и PHP без установки дополнительных компонентов.

NetBeans IDE поддерживает плагины, позволяя разработчикам расширять возможности среды. Одним из самых популярных плагинов является мощный дизайнер отчётов iReport (основанный на библиотеке JasperReports).

На идеях, технологиях и в значительной части на исходном коде NetBeans IDE базируются предлагаемые фирмой Sun коммерческие интегрированные среды разработки для Java — Sun Java Studio Creator, Sun Java Studio Enterprise и Oracle Solaris Studio (для ведения разработки на C, C++ или Фортран).

NetBeans IDE доступна в виде готовых дистрибутивов (прекомпилированных бинарных файлов) для платформ Microsoft Windows, Linux, FreeBSD, Mac OS X, OpenSolaris и Solaris (как для SPARC, так и для x86 — Intel и AMD). Для всех остальных платформ доступна возможность скомпилировать NetBeans самостоятельно из исходных текстов.

В версии 7 производитель отказался от поддержки языка Ruby и Ruby on Rails, объясняя этот отказ большим объёмом работ по поддержке Java 7 и сравнительно невысокой потребностью в этом языке и технологии среди пользователей NetBeans. Таким образом, «из коробки» последние версии NetBeans поддерживают только Java (включая Java FX, Java ME, Java EE), C/C++, Groovy, PHP, HTML, JavaScript, CSS. Также для версии 7 не имеется ни собственных, ни бесплатных сторонних компонентов поддержки UML-моделирования.

2.4 Разработка схемы программы

Схема работы программного средства представлена на рисунке 2.1.

Рисунок 2.1 - Структура программного средства

2.5 Программная реализация программного средства

Программное средство предоставляет визуальный интерфейс для доступа к требуемым данным беря на себя работу по пересылке команд между сервером и клиентом и отображению данных.

Функции:

* отображения выполненных услуг;
* отображение информации о клиентах;
* отображение информации о работниках;
* отображение каталога стрижек;
* работа с данными базы.

3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Тесты существенно различаются по задачам, которые с их помощью решаются, и по используемой технике. Различие задач тестирования приводит, естественным образом, к необходимости использовать весьма разнообразные типы (виды) тестирования. Принято подразделять тестирование на виды по следующим категориям:

* по объектам (элементам) тестирования, часто разделение на виды тестов по данному критерию называют разделением тестирования на уровни;
* по глубине тестирования, то есть разделение тестовых испытаний на типы проводится в зависимости от количества времени и объема тестируемых компонент программного продукта.

1. Модульное тестирование (Автономное или Unit-тестирование).

На данном уровне тестируются по отдельности небольшие элементы системы, максимально отделенные от других элементов и, в то же время, пригодные для тестирования.

2. Комплексное тестирование (Сборочное тестирование, integration testing или interface testing).

На данном уровне тестируются объединенные элементы (компоненты или подсистемы) общей системы, чаще всего некоторая взаимодействующая между собой группа элементов

3. Системное тестирование (system testing).

После того, как система собрана из составляющих компонентов, она должна быть протестирована на соответствие “Системным спецификациям” – реализованы ли все функциональные и нефункциональные требования к разрабатываемой системе.

4. Приемочное тестирование (Приемо-сдаточное тестирование или acceptance testing).

На данном уровне завершенное приложение (система) тестируется Заказчиком, конечными пользователями или соответствующими уполномоченными с целью определения соответствия системы “Требованиям Заказчика” и готовности системы к внедрению

5. Операционное тестирование (Release Testing)

Тестирование в среде эксплуатации позволяет выявить и нефункциональные проблемы, такие как: конфликт с другими системами, смежными в области бизнеса или в программных и электронных окружениях; недостаточная производительность системы в среде эксплуатации и др.

Виды тестирования:

Инсталляционное тестирование (Installation testing). В процессе инсталляционного тестирования проверяется корректность установки и деинсталляции программного продукта в среде максимально приближенной к эксплуатационной. Проверка правильности установки программного продукта должна быть обязательным элементом проекта по тестированию любого продукта. Основная цель состоит в том, чтобы убедиться, что продукт может быть установлен/деинсталлирован при различных условиях – таких как: новая инсталляция, усовершенствование системы (upgrade), установка по умолчанию, полная установка, установка по выбору.

Дымное тестирование (проверка на дым, Smoke testing). Как правило, используется для определения, готова ли программа для проведения более обширного тестирования.

Функциональное тестирование (Functional testing). Проверка соответствия продукта функциональным требованиям и спецификациям.

Функциональные требования включают в себя: Функциональную пригодность, точность, способность к взаимодействию, соответствие стандартам и правилам,защищённость .

Нефункциональное тестирование описывает тесты, необходимые для определения характеристик программного обеспечения, которые могут быть измерены различными величинами. В целом, это тестирование того, "Как" система работает. Тестирование нефункциональных требований, то есть : тестирование производительности, защищенности, использования и восстановления.

Регрессионное тестирование (Regression testing). Повторное тестирование после внесения изменений в программное обеспечение или в его окружение (в новой версии приложения), чтобы убедиться в том, что функции, которые работали в предыдущей версии системы, по-прежнему работают так, как ожидалось, а найденные дефекты успешно исправлены (все протестированное ранее тестируется повторно)

Тестирование графического интерфейса пользователя (User Interface testing). Тестирование интерфейса – экранов, кнопок и т.д. Большая часть функциональности ПО реализуется, как правило, через пользовательский интерфейс.

Тестирование программы проводилось в два этапа: тестирование устойчивости и тестирование функциональности.

1. В тесте 1 – проверялась поведение клиента при попытке подключиться к отключенному серверу рисунок 3.1.

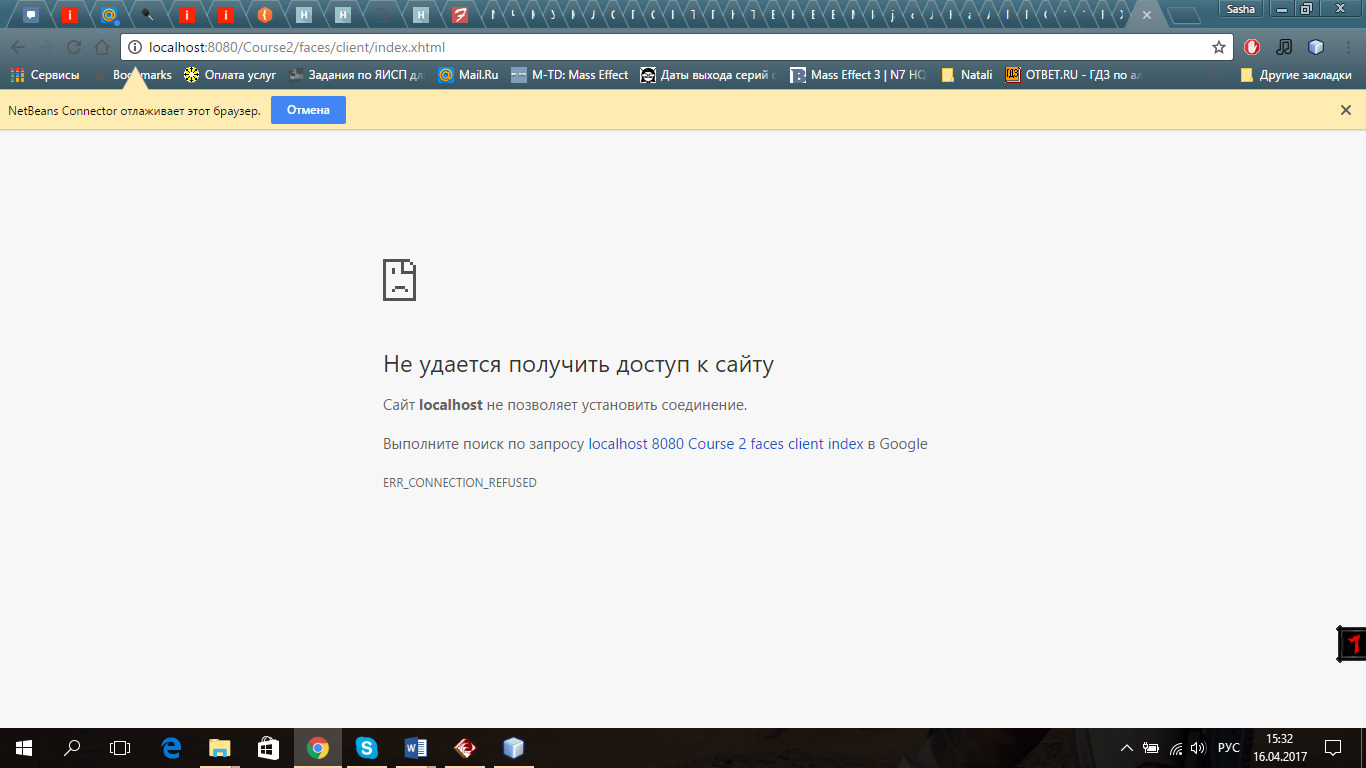


Рисунок 3.1 – попытка подключения к отключенному серверу

1. В тесте 2 – после того как клиент подключился, уведомления пользователю не показывается
2. В тесте 3 – при незапланированном отключении сервера, клиент некорректно завершал свою работу, рисунок 3.2.

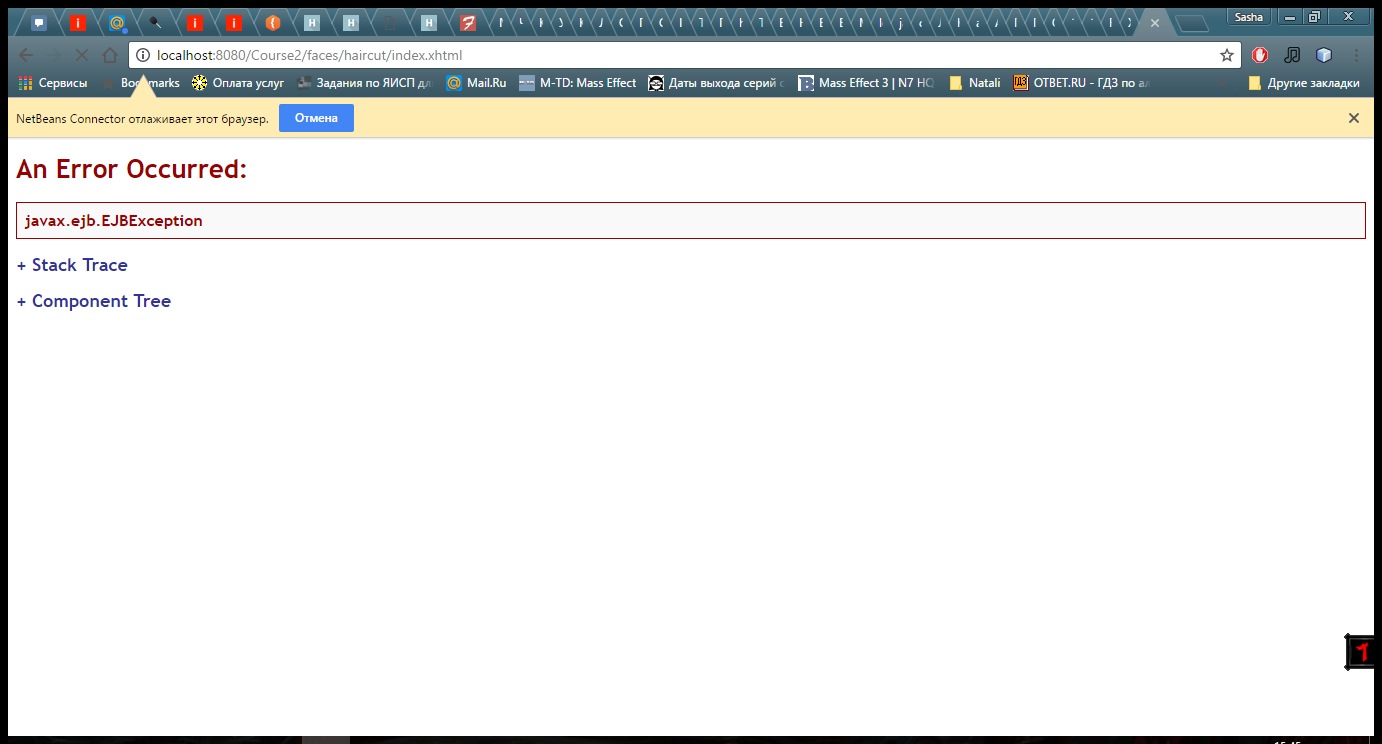


Рисунок 3.2 – ошибка при неожиданном отключении сервера

**4 МЕТОДИКА ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО СРЕДСТВА**

После запуска локального сервера в новой вкладке браузера откроется главная страница приложения, представленное на рисунке 4.1. В главном окне приложения находятся основные элементы навигации:

* Кнопка «Home»;
* Кнопка с выпадающим списком «Maintenance»;

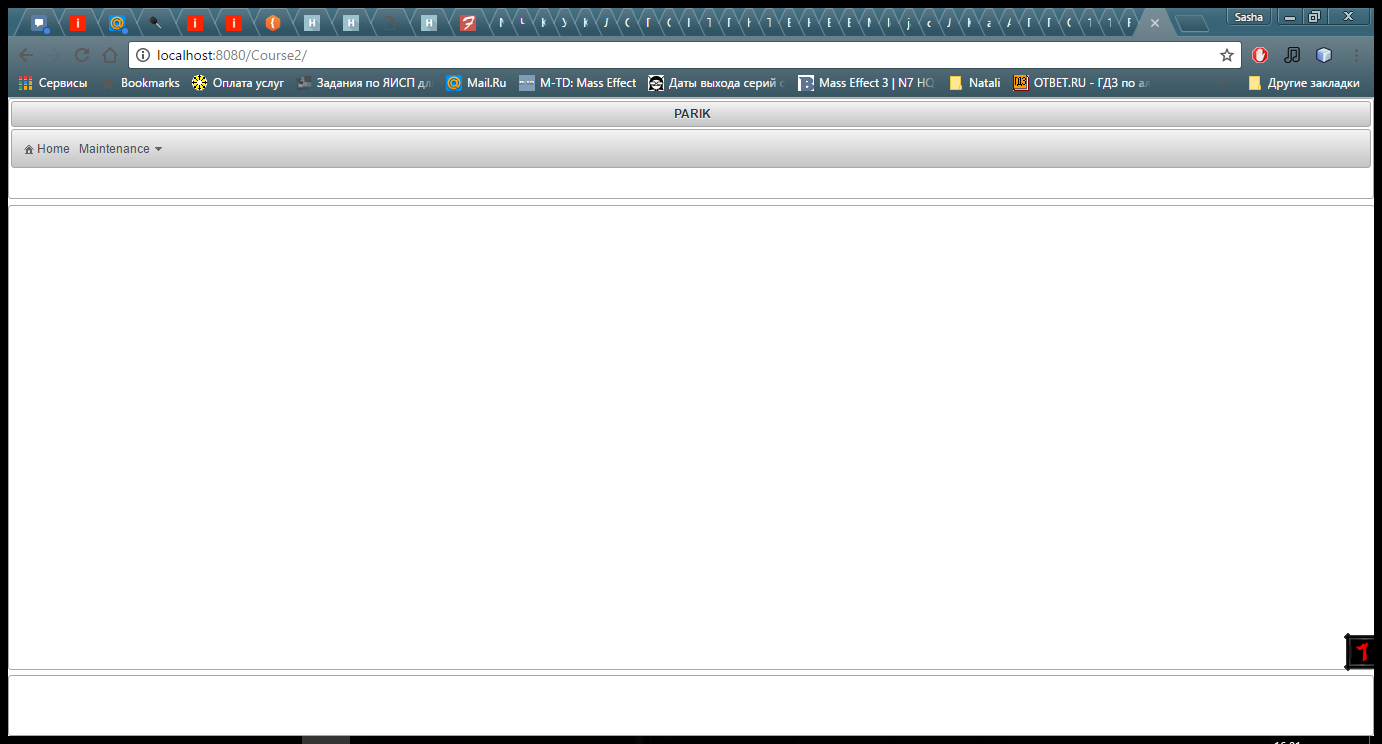


Рисунок 4.1 – Главная страница приложения

Для того, чтобы начать работу с данными необходимо по нажатию кнопки «Maintenance» выбрать из выпадающего списка таблицу необходимую для работы. Процесс выбора таблицы представлен на рисунке 4.2.

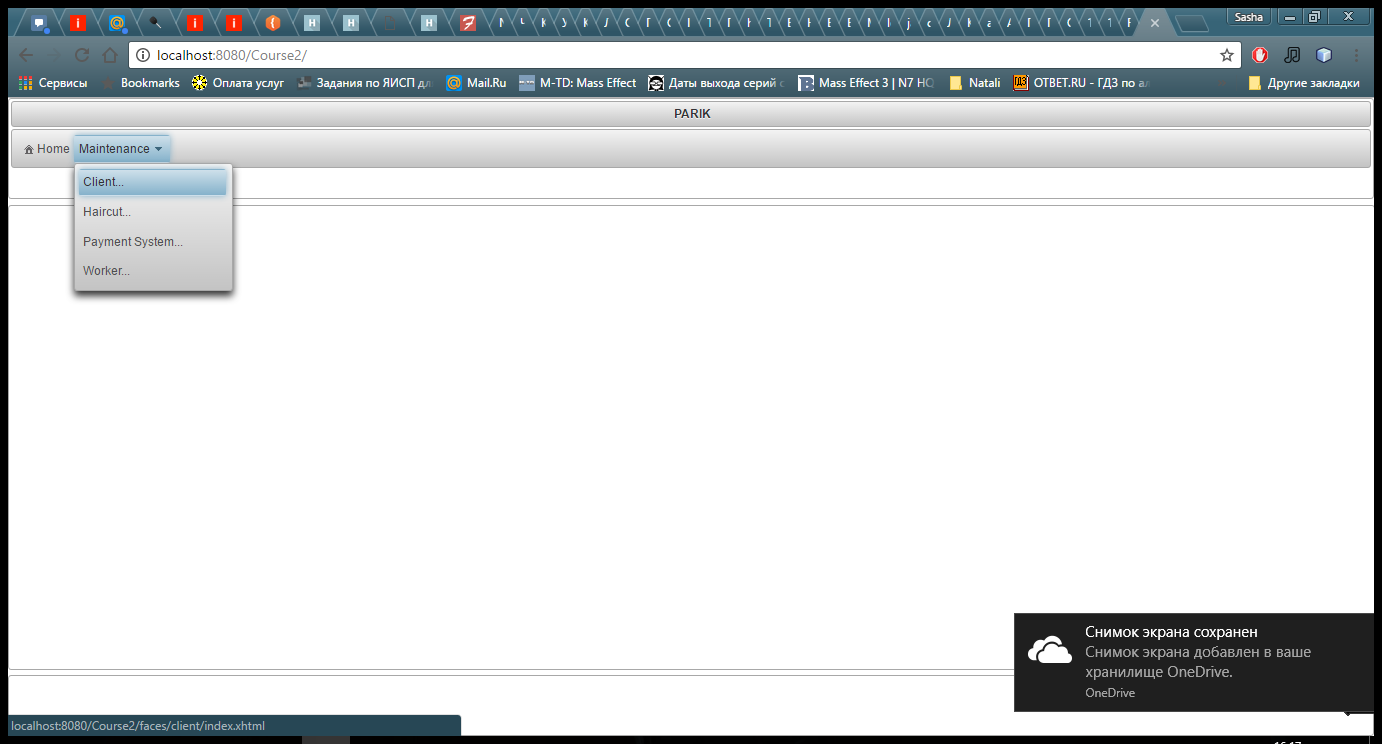


Рисунок 4.2 – Выбор таблицы для работы

При выборе пункта с именем одной из таблиц данных откроется страница с данными таблицы, рисунок 4.3. Для работы с данными необходимо нажать правой кнопкой мыши по таблице и откроется меню, там, где можно будет выбрать действие для работы с таблицей, рисунок 4.4. При выборе пункта меню «Create» (так же эта функция продублирована кнопкой с таким же именем в левой нижней части таблицы) появится форма для добавления новых данных в таблицу, рисунок 4.5. При выборе пункта меню «Edit» появится форма для редактирования данных таблицы, рисунок 4.6. При выборе пункта меню «View» появится форма для просмотра данных выбранной строки, рисунок 4.7. При выборе пункта меню «Delete» выбранная строка удалиться из таблицы. Работа с другими таблицами происходит аналогично, на рисунках 4.8 – 4.10 изображены другие таблицы.

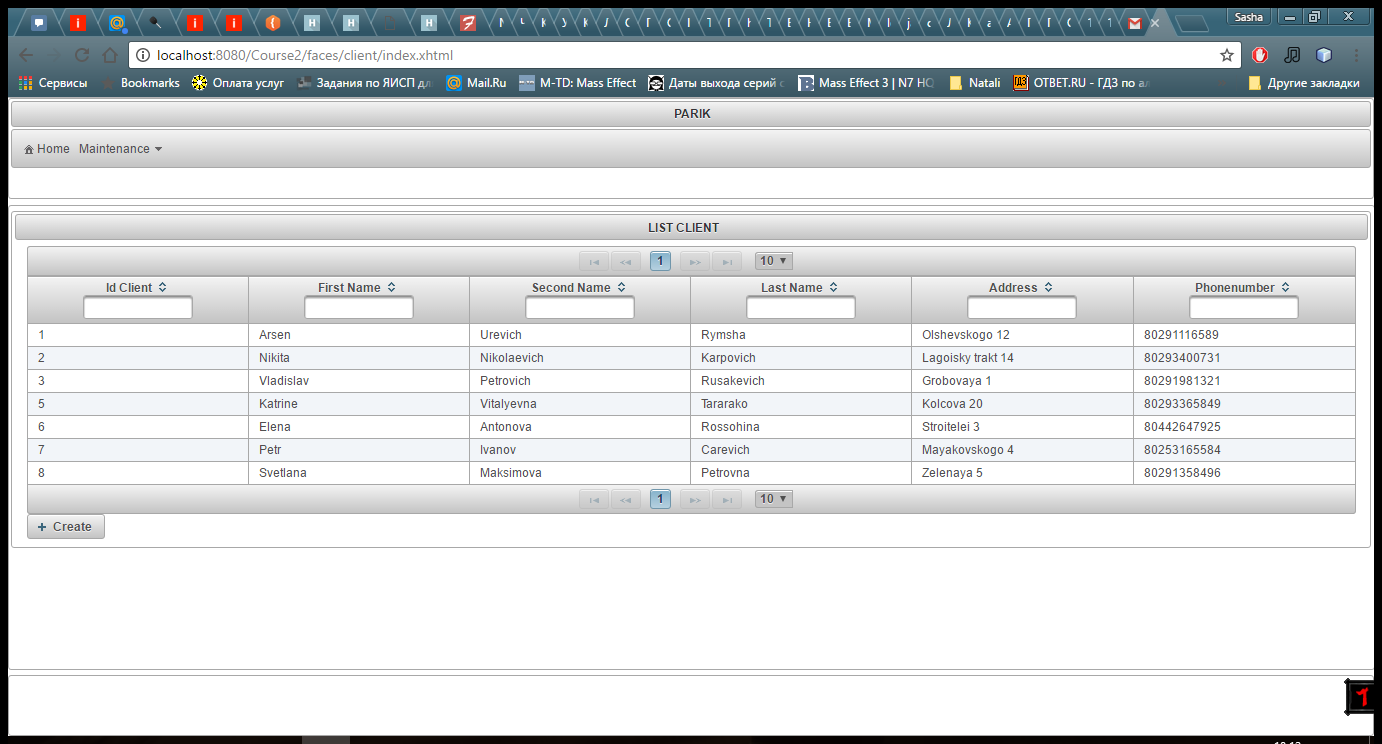


Рисунок 4.3 – Страница с данными таблицы «Client»

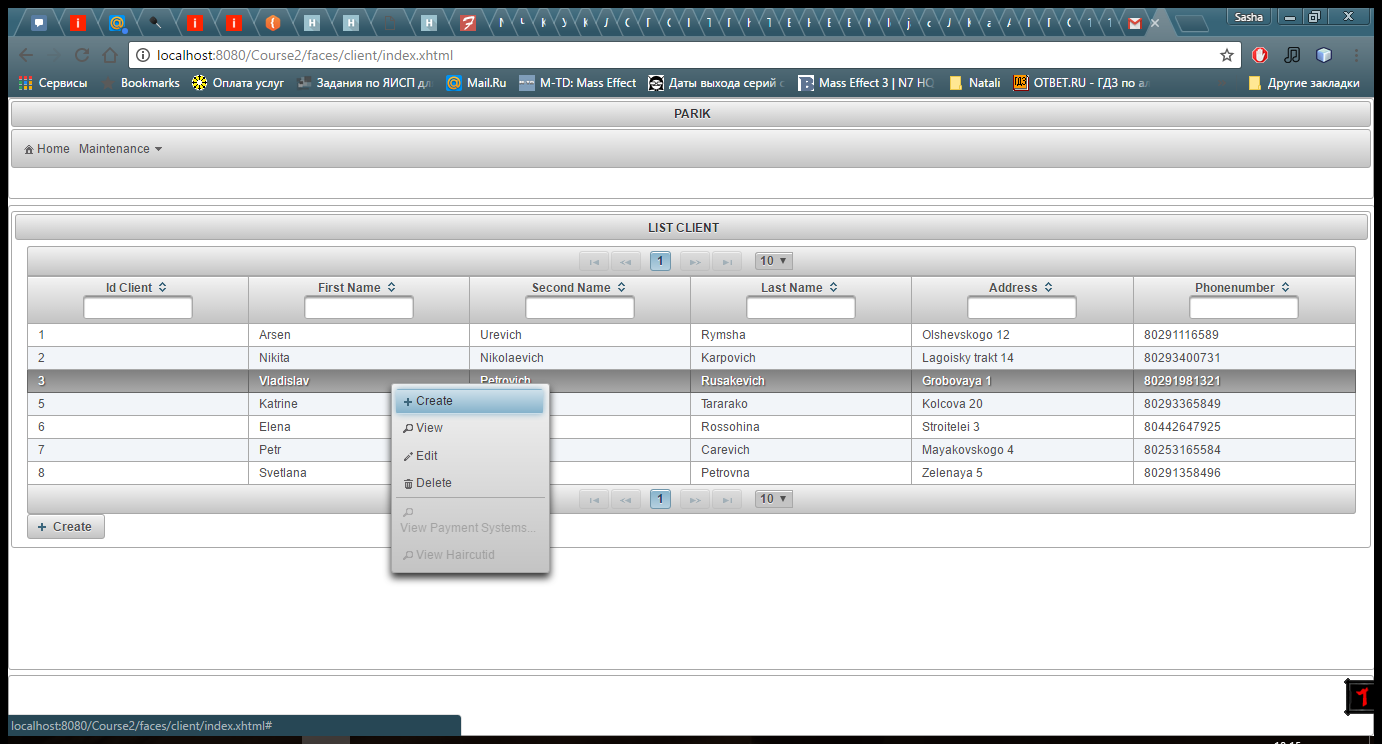


Рисунок 4.4 – Меню выбора действий над таблицей

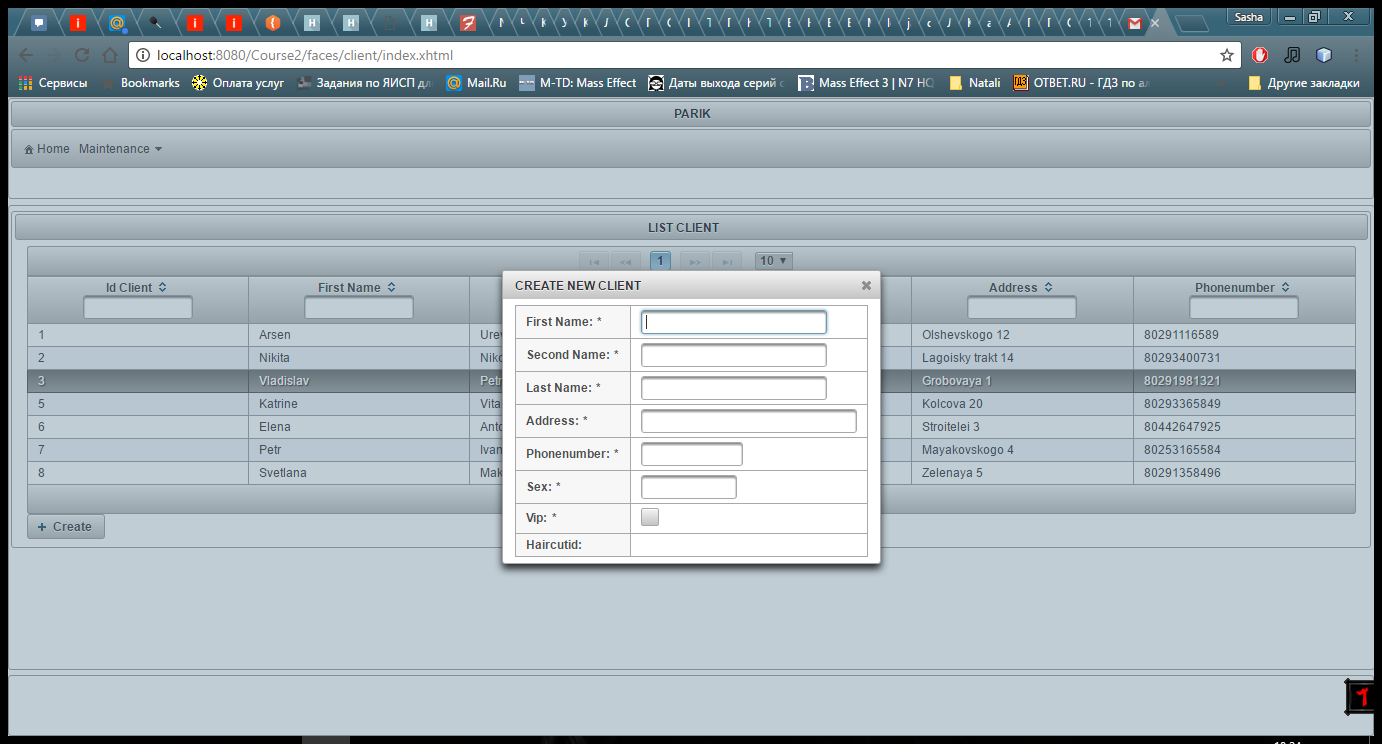


Рисунок 4.5 – Форма добавления данных

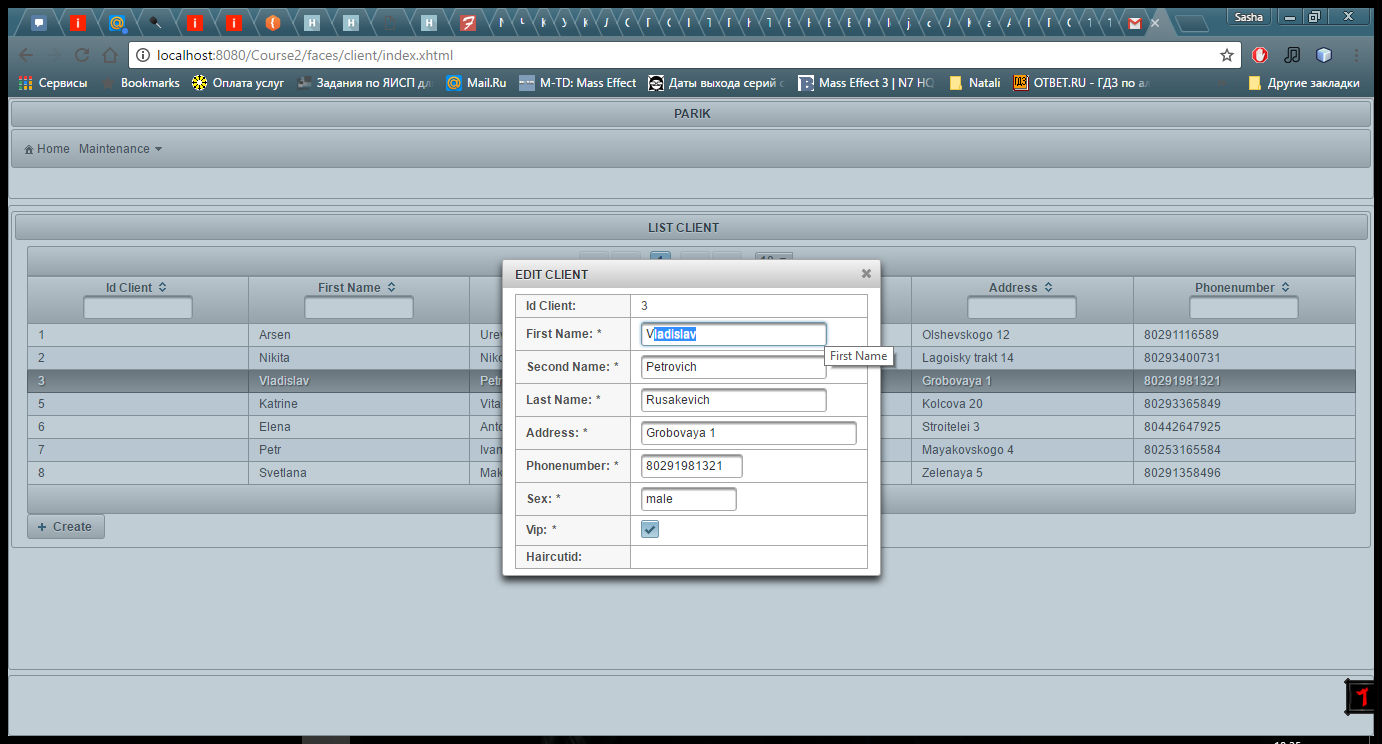


Рисунок 4.6 – Форма редактирования данных

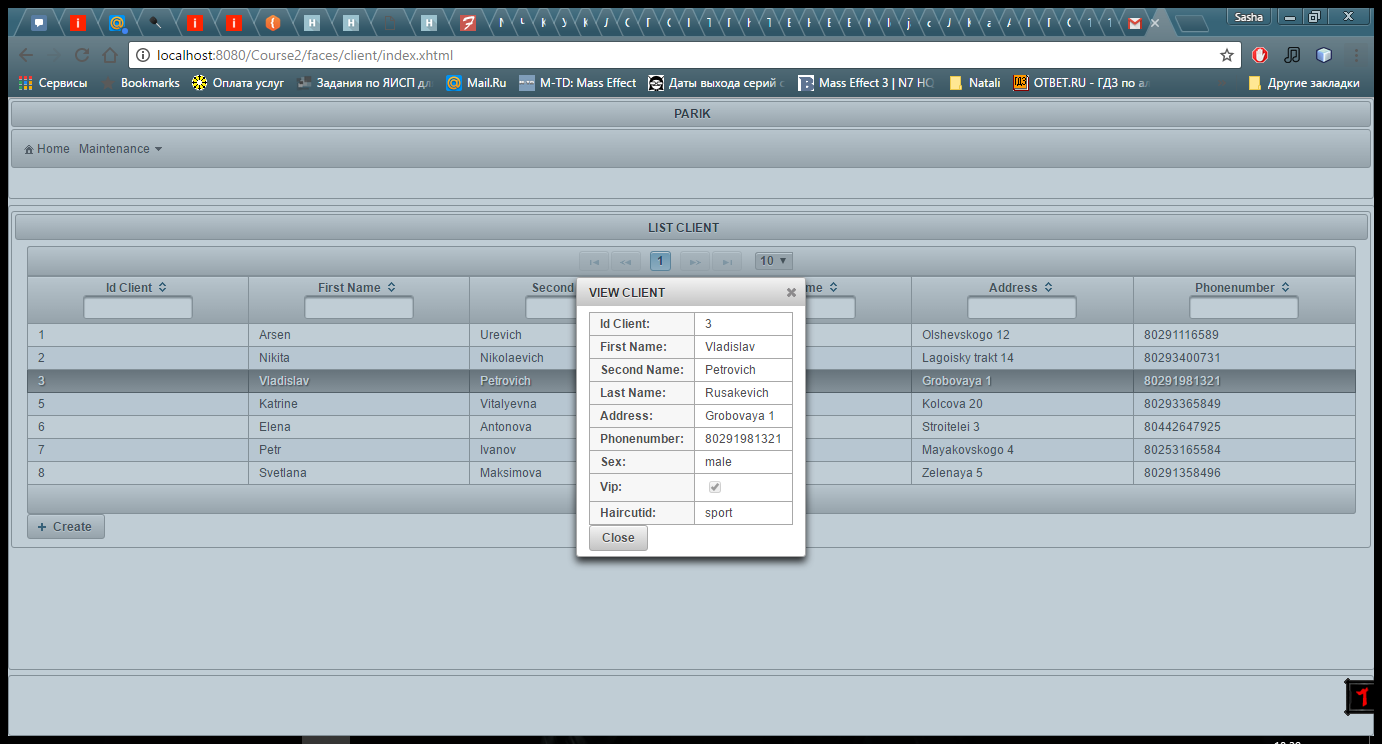


Рисунок 4.7 – Форма просмотра данных

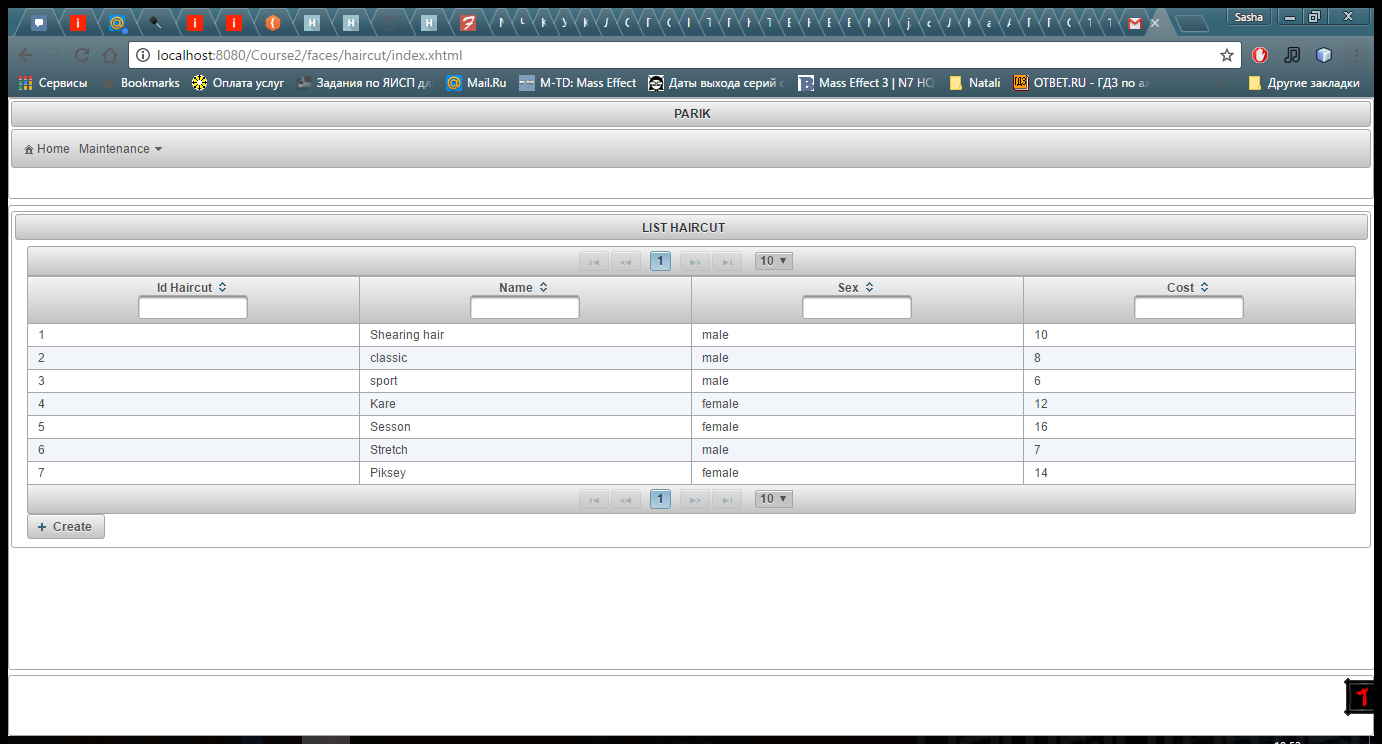


Рисунок 4.8 – Страница с данными таблицы «Haircut»

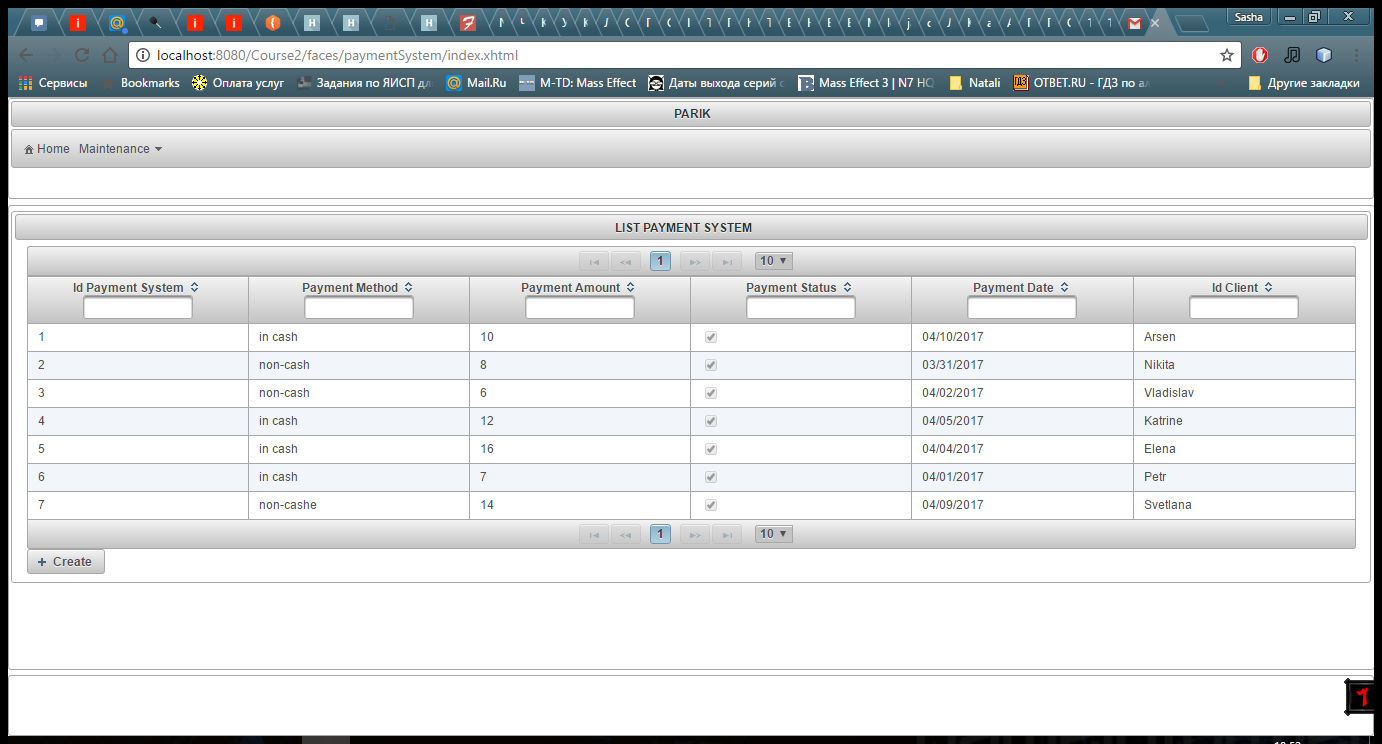


Рисунок 4.9 – Страница с данными таблицы «PaymentSystem»

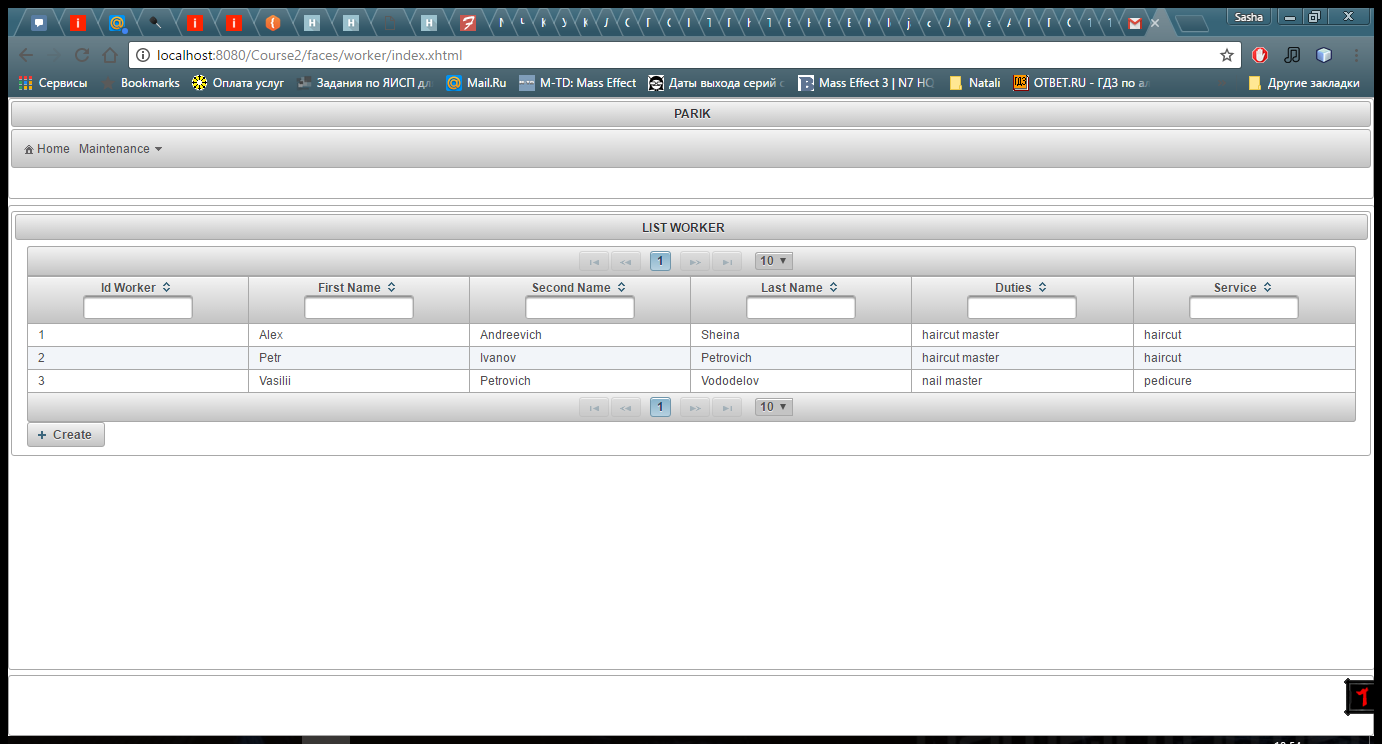


Рисунок 4.10 – Страница с данными таблицы «Worker»

**ЗАКЛЮЧЕНИЕ**

В данном курсовом проекте разработано Программное средство для автоматизации рабочего места в парикмахерской.

В данном программном средстве реализовано:

* Ведение списка клиентов, с последующим его расширением и редактированием.
* Ведение каталога стрижек, с последующим его расширением и редактированием.
* Учет услуг клиентов, который отражает информацию о услуге и ее стоимости.

Курсовая работа была реализована на высокоуровневом языке программирования Java с использованием среды разработки NetBeans 8.0.2 и Framework Faces.

Для проверки правильность работы программы были применены следующие виды тестов: модульное тестирование и интеграционное тестирование. Были проведены три теста, по результатам которых были исправлены ошибки и недочеты.

Все задачи, которые ставились в начале проектирования были реализованы и в некоторых местах дополнены.

В дальнейшем программное средство может быть модифицировано различными элементами управления и настройками, которые позволят сделать данное программное средство более универсальным, а также позволят решать другие более сложные задачи чем простое управление СУБД.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Свободная энциклопедия [Электронный ресурс] / центр информ. Режим доступа: http://ru.wikipedia.org/wiki/Windows\_10.
2. Программирование на языке Java: учеб. пособие / В.В. Фаронов; под ред. В.С. Кустов. – Питер, 2007. – 240с.
3. Язык Java: самоучитель / А.В. Фролов, Г.В. Фролов; под ред. О.А Фаронов. – Москва, 2003. – 560с.
4. Дьюсон Робин. для начинающих разработчиков. СПб.: БХВ-Питер, 2009. – 704 с.
5. Сайт разработчика [Электронный ресурс] центр информ. Режим доступа: http://www.protesting.ru/testing/. Дата доступа: 04.04.2017.
6. Энциклопедия [Электронный ресурс] центр информ. Режим доступа: http://habrahabr.ru/post/169381/ /ARM. Дата доступа: 06.04.2017.
7. Канер С., Фолк Дж., Енг Кек Нгуен. Тестирование программного обеспечения, 2004. – 73 с.