*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***КУРСОВАЯ РАБОТА***

*Прототип программной системы для*

*информационной системы «Служба такси»*

*Выполнили: студенты гр. ИСТ-111 Якунченков А.А.*

*Золотов В.И.*

*Принял: асс. Тимофеев А.*

*Владимир, 2014*

# Содержание

[Содержание 1](#_Toc376153914)

[1.ВВЕДЕНИЕ 4](#_Toc376153915)

[3.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc376153916)

[4.ФУНКЦИОНАЛЬНАЯ ДЕКОМПОЗИЦИЯ СИСТЕМЫ 6](#_Toc376153917)

[4.1.Сценарий взаимодействия пользователя с программной системой. 6](#_Toc376153918)

[4.2. Оформление онлайн-заказа для вызова такси. 6](#_Toc376153919)

[4.3. Просмотр информаци о существующих заказах. 7](#_Toc376153920)

[4.4. Просмотр информации о существуюших таксистах и их машинах. 7](#_Toc376153921)

[4.5 Авторизация. 7](#_Toc376153922)

[5.СТРУКТУРА БАЗЫ ДАННЫХ 8](#_Toc376153923)

[5.1. Построение структуры базы данных. 8](#_Toc376153924)

[5.2.Логическая схема базы данных. 11](#_Toc376153925)

[5.3. Физическая схема базы данных. 12](#_Toc376153926)

[5.5. Создание таблиц базы данных. 12](#_Toc376153927)

[6. СТРУКТУРА БИЗНЕС-ЛОГИКИ СИСТЕМЫ. 16](#_Toc376153928)

[6.1. Функции, необходимые программной системе. 16](#_Toc376153929)

[6.2. Диаграмма классов. 17](#_Toc376153930)

[6.3.Словесное описание сущностей, описание атрибутов, классов и функций. 18](#_Toc376153931)

[6.3.1.Package model. 18](#_Toc376153932)

[6.3.2. Package DAO. 21](#_Toc376153933)

[6.3.3. Package beans. 23](#_Toc376153934)

[6.4.Объяснение связей диаграммы классов. 27](#_Toc376153935)

[7. РЕАЛИЗАЦИЯ БИЗНЕС-ЛОГИКИ ПРОТОТИПА. 28](#_Toc376153936)

[7.1 Package Model 28](#_Toc376153937)

[7.1.1. Car.java 28](#_Toc376153938)

[7.1.2 Group.java 29](#_Toc376153939)

[7.1.3 Order.java 29](#_Toc376153940)

[7.1.4. Taxist.java 31](#_Toc376153941)

[7.1.5. User.java 34](#_Toc376153942)

[7.1.6. UserGroup.java 35](#_Toc376153943)

[7.2.Package DAO. 35](#_Toc376153944)

[*7.2.1. ConnectionClass.java* 35](#_Toc376153945)

[*7.2.2. DataAccesObject* 36](#_Toc376153946)

[*7.2.3. ConnectionGlassFish.java* 49](#_Toc376153947)

[7.3.Package beans. 49](#_Toc376153948)

[*7.3.1. AdminBean.java* 49](#_Toc376153949)

[*7.3.2. AuthBackingBean.java* 53](#_Toc376153950)

[*7.3.3. CabbieBean.java* 54](#_Toc376153951)

[7.3.4. OrderBean.java 57](#_Toc376153952)

[8.1. Директория resources. 60](#_Toc376153953)

[8.2. Директория styles. 60](#_Toc376153954)

[8.3. Директория templates. 60](#_Toc376153955)

[8.4. Основные страницы. 60](#_Toc376153956)

[9.РЕАЛИЗАЦИЯ ВЕБ-ИНТЕРФЕЙСА СИСТЕМЫ. 62](#_Toc376153957)

[9.1. Директория fonts. 62](#_Toc376153958)

[9.2. Директория swarming 62](#_Toc376153959)

[*9.2.1.swarming/default.css* 62](#_Toc376153960)

[9.3.Директории templates,admin,cabbie и основные страницы. 72](#_Toc376153961)

[*9.3.1.addNewCabbie/content.xhtml* 72](#_Toc376153962)

[*9.3.2. adminMainPage/content.xhtml* 73](#_Toc376153963)

[*9.3.3. adminMainPage/header.xhtml* 76](#_Toc376153964)

[*9.3.4. cabbieDeleted/content.xhtml* 77](#_Toc376153965)

[*9.3.5. CabbieMainPage/content.xhtml* 77](#_Toc376153966)

[*9.3.6. CabbieMainPage/header.xhtml* 80](#_Toc376153967)

[*9.3.7. common/copyrightxhtml* 81](#_Toc376153968)

[*9.3.8. common/header.xhtml* 81](#_Toc376153969)

[*9.3.9. common/logo.xhtml* 82](#_Toc376153970)

[*9.3.10. editCarData/content.xhtml* 83](#_Toc376153971)

[*9.3.11. editTaxistData/content.xhtml* 84](#_Toc376153972)

[*9.3.12. index/content.xhtml* 85](#_Toc376153973)

[*9.3.13.index/header.xhtml* 86](#_Toc376153974)

[*9.3.14. noneRegisteredMainPage/content.xhtml* 87](#_Toc376153975)

[*9.3.15. noneRegisteredMainPage/header.xhtml* 88](#_Toc376153976)

[*9.3.16. orderInProgressPage/content.xhtml* 89](#_Toc376153977)

[*9.3.17. SuccessAddOrder/content.xhtml* 91](#_Toc376153978)

[*9.3.18.SuccessFinishOrder/content.xhtml* 92](#_Toc376153979)

[*9.3.19 successTakeOrder/content.xhtml* 93](#_Toc376153980)

[*9.3.20. takingOrder/content.xhtml* 93](#_Toc376153981)

[*9.3.21. gmap.xhtml* 95](#_Toc376153982)

[*9.3.22. index.xhtml* 95](#_Toc376153983)

[*9.3.23. login.xhtml* 96](#_Toc376153984)

[*9.3.24 noneRegisteredMainPage.xhtml* 97](#_Toc376153985)

[*9.3.25. succesAddOrder.xhtml* 98](#_Toc376153986)

[*9.3.26. admin/addNewCabbie.xhtml* 99](#_Toc376153987)

[*9.3.27. admin/adminMainPage.xhtml* 100](#_Toc376153988)

[*9.3.28. admin/cabbieDeleted.xhtml* 100](#_Toc376153989)

[*9.3.29. admin/editCarDate.xhtml* 101](#_Toc376153990)

[*9.3.30. admin/editTaxistData.xhtml* 102](#_Toc376153991)

[*9.3.31. cabbie/cabbieMainPage.xhtml* 103](#_Toc376153992)

[*9.3.32. cabbie/orderInProgressPage.xhtml* 104](#_Toc376153993)

[*9.3.33.cabbie/successFinishOrder.xhtml* 105](#_Toc376153994)

[*9.3.34.cabbie/successTakeOrder.xhtml* 105](#_Toc376153995)

[9.4. Директория WEB-INF. 107](#_Toc376153996)

[*9.4.1. faces-config.xml* 107](#_Toc376153997)

[*9.4.2. sun-web.xml* 107](#_Toc376153998)

[*9.4.3. web.xml* 108](#_Toc376153999)

[10. ДЕМОНСТРАЦИЯ ПРОТОТИПА ПРОГРАММНОЙ СИСТЕМЫ. 109](#_Toc376154000)

[11. ЗАКЛЮЧЕНИЕ. 119](#_Toc376154001)

[12.СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ. 120](#_Toc376154002)

# 1.ВВЕДЕНИЕ

Курсовая работа является этапом дисциплины «Распределенные программные системы», и является частью подготовки бакалавров по специальности 230400. Выполнение курсовой работы способствует систематизации и обобщению знаний, выработке умений правильно ориентироваться в программном коде.

Цель курсовой работы – приобретение практических навыков создания прототипа информационной системы. Курсовая работа разрабатывается на основе задания, полученного у преподавателя. Для реализации данной цели студент должен:

* на основе анализа предметной области сформулировать перечень задач для своей информационной системы;
* построить реляционную структуру базы данных;
* построить структуру бизнес-логики системы;
* реализовать бизнес-логику прототипа;
* построить структуру веб-интерфейса системы;
* полностью реализовать прототип;

В курсовой работе должны использоваться следующие технологии:

* Sun JDK 1.7.0, сервер приложений Glassfish V4
* СУБД: MySQL 5.1
* Технологии: EJB, JPA 2.1, CDI, JSF 2.2

Защита курсовой работы должна выявить степень подготовленности студента к умениям:

* анализировать предметную область;
* строить модель предметной области;
* определять требования к разрабатываемой программной системе;
* выбирать и применять различные технологии для достижения конкретных целей;2.ПОСТАНОВКА ЗАДАЧИ

В процессе проектирования программной системы необходимо:

* составить исчерпывающее описание предметной области, которое

будет давать всю необходимую информацию для проектирования структуры бизнес-логики системы и структуры веб-интерфейса;

* выявить сущности предметной области, которые впоследствии станут основными частями базы данных и программной системы;
* устранить избыточность данных, и привести данные к удобному для обработки виду;
* выполнить функциональную декомпозицию системы, определив возможные действия пользователя;
* спроектировать структуру бизнес-логики системы, опираясь на функциональную декомпозицию системы;
* на основе функциональной декомпозиции системы, спроектировать веб-интерфейс системы таким образом, чтобы обеспечить максимальное удобство работы пользователя с программной системой;

В процессе реализации прототипа программной системы необходимо:

* реализовать бизнес-логику системы, на основе структуры бизнес-логики системы;
* реализовать веб-интерфейс системы, на основе структуры веб-интерфейса системы;
* соединить бизнес-логику системы, и веб-интерфейс системы;

# 3.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Совсем недавно мы и представить не могли, что такси можно заказать прямо со своего рабочего места или дома даже не снимая трубки телефона. Однако с приходом интернета в наш быт, заказ такси настолько упростился, что это сделать можно в считанные секунды из любой точки мира и все это совершенно бесплатно. Кроме того, заказ такси онлайн очень прост. Сегодня пользоваться интернетом умеет практически любой человек, поэтому зайти на сайт такси и оформить заказ такси занимает несколько десятков секунд. Естественно, что такая простота привлекает клиентов, к тому же многие из них проводят в интернете очень много времени.

# 4.ФУНКЦИОНАЛЬНАЯ ДЕКОМПОЗИЦИЯ СИСТЕМЫ

## 4.1.Сценарий взаимодействия пользователя с программной системой.

Пользователь взаимодействует с программной системой через веб-сайт, работая с элементами графического интерфейса, следуя инструкциям-указаниям, направляющим пользователя, а также названию разделов. Программная система построена таким образом, что неверное действие пользователя, не приведет к верному ответу системы, и уж тем более к поломке системы. В случае неверного действия пользователь всего лишь получит предупреждающее сообщение, в котором будет информация о том, что он неправильно сделал.

## 4.2. Оформление онлайн-заказа для вызова такси.

Пользователь находится на веб-сайте, и переходит в один из 3 главных разделов, под названием «Для клиентов», которые находятся в верхней части страницы, и попадает на страницу, на которой пользователю предлагается возможность осуществить заказ такси онлайн.

## 4.3. Просмотр информаци о существующих заказах.

Пользователь находится на веб-сайте, и переходит в один из 3 главных разделов, под названием «Для таксистов», находятся в верхней части страницы.При условии, что пользователь авторизовался, как таксист,он получает доступ к данному разделу.В разделе «Для таксистов» размещена информация о существующих заказах, а также реализована возможность выполнения одного из них.

## 4.4. Просмотр информации о существуюших таксистах и их машинах.

Пользователь находится на веб-сайте, и переходит в один из 3 главных разделов, под названием «Для администраторов», которые находятся в верхней части страницы. При условии, что пользователь авторизовался, как администратор,он получает доступ к данному разделу и попадает на страницу, на которой размещена информация о работающих таксистах, а также об их машинах, реализована возможность добавления нового таксиста, удаление уже существующего, а также редактирование информации о машине или таксисте.а

## 4.5 Авторизация.

Под главным меню, находящимся в верхней части страницы, находится панель авторизации. При вводе данных в поля «Логин» и «Пароль» и нажатии кнопки «Авторизоваться», происходит авторизация пользователя в системе. После авторизации устанавливается личность пользователя, на панели авторизации пишется приветствие авторизовавшемуся пользователю.

# 5.СТРУКТУРА БАЗЫ ДАННЫХ

## 5.1. Построение структуры базы данных.

Таблица «Группы»:

1. Название группы. Тип NVARCHAR – набор символов (букв), представляющих собой название группы. Not null – поле не может быть пустым, т.к. любая группа должна иметь название.
2. Описание группы. Тип NVARCHAR – набор символов (букв), представляющих собой краткое описание группы. Not null – поле не может быть пустым, т.к. любая группа должна иметь краткое описание. Связь «Группы-Пользователи групп» один-ко-многим.

Таблица «Пользователи групп»:

Является связывающей таблицей ,между таблицами: «Пользователи» и «Группы»

Таблица «Пользователи»:

1. Логин. Тип NVARCHAR – тип, Тип NVARCHAR – набор символов (букв), представляющих собой логин пользователя. Not null – поле не может быть пустым, т.к. любой пользователь должен иметь логин.
2. Имя. Тип NVARCHAR – тип, Тип NVARCHAR – набор символов (букв), представляющих собой имя пользователя. Not null – поле не может быть пустым, т.к. каждый пользователь должен иметь имя.
3. Фамилия. Тип NVARCHAR – тип, Тип NVARCHAR – набор символов (букв), представляющих собой фамилию пользователя. Not null – поле не может быть пустым, т.к. каждый пользователь должен иметь фамилию.
4. Отчество. Тип NVARCHAR – тип, Тип NVARCHAR – набор символов (букв), представляющих собой отчество пользователя. Not null – поле не может быть пустым, т.к. каждый пользователь должен иметь отчество.
5. Пароль. Тип NCHAR – тип, Тип NCHAR – набор символов , представляющих собой пароль пользователя. Not null – поле не может быть пустым, т.к. каждый пользователь должен иметь свой пароль. Связь: «Пользователи -Таксисты» один-к-одному.

Таблица «Таксисты»:

1. Содержит User\_id пользователя, который является таксистом. Связь: «Таксисты – Автомобили» один-к-одному.

Таблица «Заказы»:

1. Фамилия. Тип NVARCHAR – набор символов (букв), представляющих собой фамилию, имя, отчество клиента. Not null – поле не может быть пустым, т.к. каждый клиент, который заказал такси заносится в базу данных.
2. Номер телефона. Тип NVARCHAR – набор символов (букв), представляющих собой номер телефона заказчика. Not null – поле не может быть пустым, т.к. каждый заказчик должен иметь номер телефона.
3. Дата и время заказа. Тип DateTime – тип, представляющий собой совокупность числа, месяца, года, а также часов и минут заказа. Not null – поле не может быть пустым, т.к. любой заказ имеет дату и время.
4. Статус заказа. Тип Integer – тип, представляющий собой целочисленное значение статуса заказа, указанного на билете. Not null – поле не может быть пустым, т.к. на каждом купленном билете должен быть указан ряд.
5. Улица .Тип NVARCHAR – набор символов (букв), представляющих собой название улицы. Not null – поле не может быть пустым, т.к. для того чтобы выполнить заказ нужно знать название улицы .
6. Дом. Тип NVARCHAR – набор символов (букв), представляющих собой номер дома. Not null – поле не может быть пустым, т.к. для того чтобы выполнить заказ нужно знать номер дома .
7. Подьезд. Тип NVARCHAR – набор символов (букв), представляющих собой номер подьезда. Not null – поле не может быть пустым, т.к. для того чтобы выполнить заказ нужно знать номер подьезда .
8. Комментарий. Тип NVARCHAR – набор символов (букв), представляющих собой название комментарии к заказу. null – поле может быть пустым, т.к. комментарий писать не обязательно.
9. Путь. Тип NVARCHAR – набор символов (букв), представляющих собой название пути. Not null – поле не может быть пустым, т.к. для того чтобы выполнить заказ нужно знать название пути . Связь: «Таксисты – Заказы» один-ко-многим.

Таблица «Автомобили»:

1. Марка . Тип NVARCHAR – набор символов (букв), представляющих собой название марки автомобиля. Not null – поле не может быть пустым, т.к. любая автомобиль имеет марку .
2. Цвет . Тип INTEGER – тип, представляющий собой целочисленное значение заработной платы за месяц. Not null – поле не может быть пустым, т.к. любой автомобиль имеет цвет.
3. Гос.номер. Тип NVARCHAR – набор символов (букв), представляющих собой государственный номер автомобиля. Not null – поле не может быть пустым, т.к. на автомобиле должен быть указан государственный номер. Связь: « Автомобили – Таксисты» один-к-одному.

Таблица «Бухгалтерия» :

1. Наименование статьи расходов и доходов – Тип NVARCHAR. Поле описывает откуда пришли деньги, или куда они были потрачены.
2. Сумма дохода – Тип Integer, сумма полученных денег.
3. Сумма расхода – Тип Integer, сумма потраченных денег.
4. Дата - тип Date. Описывает дату произошедших доходов или расходов.

Таблица «Бухгалтеры»:

1. ID\_buhg – идентификатор бухгалтера
2. ID\_Пользователя – идентификатор пользователя
3. Id\_buh – идентификатор учета расходов и доходов.

Таблица «Фирмы» :

1) ID\_Fr – идентификатор фирмы.

2)ID\_Tax – идентификатор таксиста.

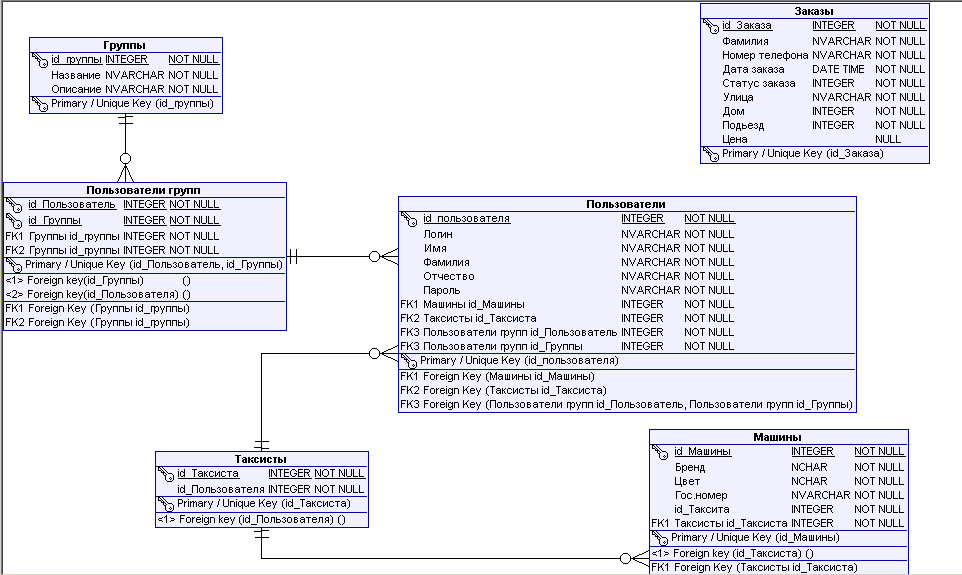
3)Наименование – название фирмы.

4)ID\_Car – идентификатор машины.

## 5.2.Логическая схема базы данных.

Логическая модель данных – то описание, диаграмма, изображение или модель предметной области, в которой показаны только объекты (сущности), которыми должна оперировать разрабатываемая система, и связи между ними.

В результате анализа предметной области была получена логическая модель данных (Рис.1).



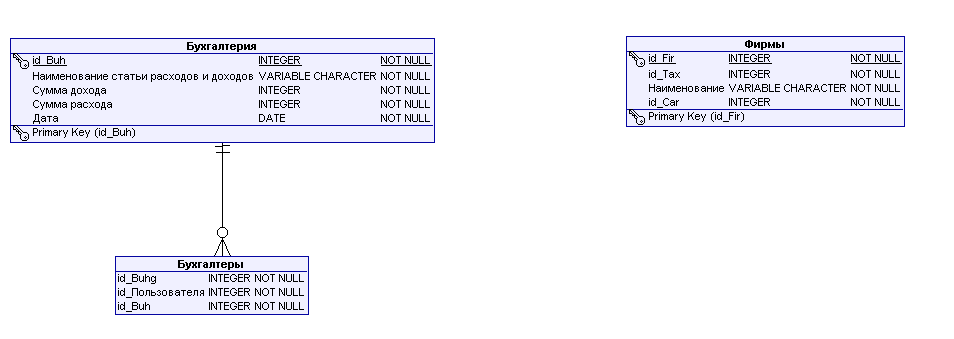
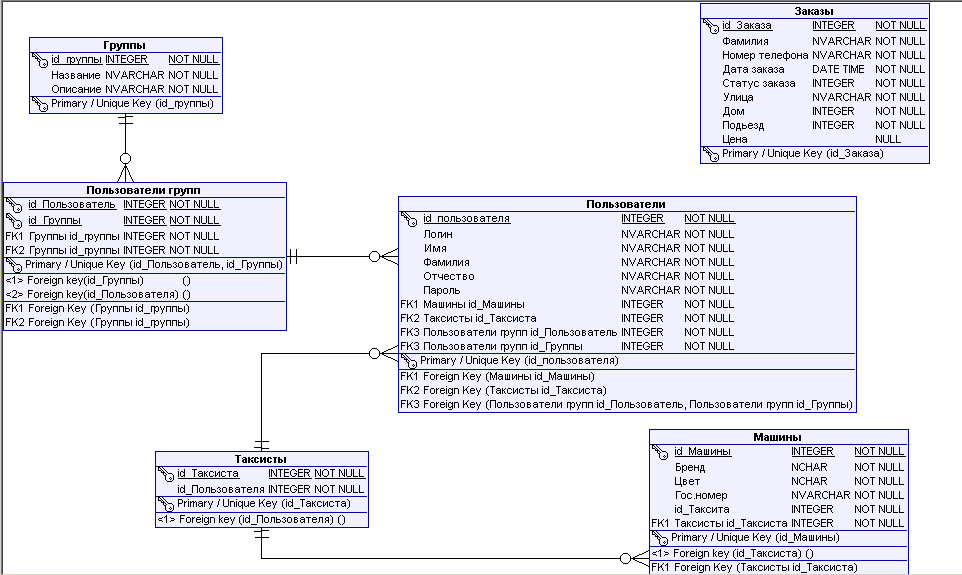


Рис.1.Логическая модель данных.

## 5.3. Физическая схема базы данных.

Физическая модель данных описывает данные средствами конкретной СУБД. Отношения, разработанные на стадии формирования логической модели данных, преобразуются в таблицы, атрибуты становятся столбцами таблиц, для ключевых атрибутов создаются уникальные индексы, связи реализуются с помощью механизма внешних ключей.

В результате анализа предметной области была получена физическая модель данных.



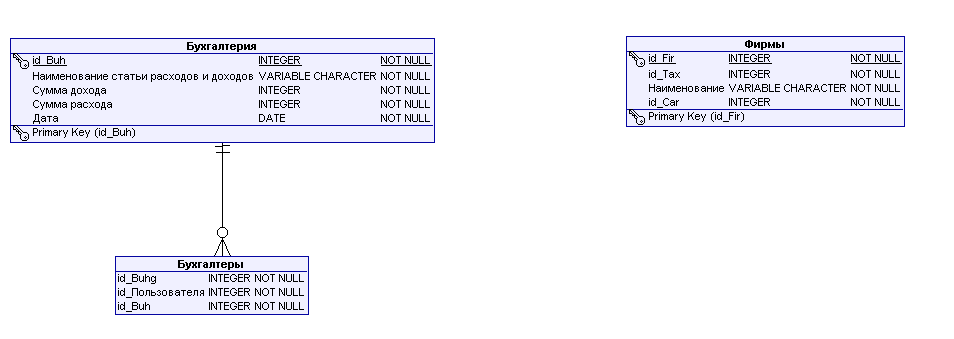


Рис.2. Физическая модель данных.

## 5.4. Создание таблиц базы данных.

CREATE TABLE `groups` (

`group\_id` int(10) NOT NULL,

`group\_name` varchar(20) NOT NULL,

`group\_desc` varchar(200) DEFAULT NULL,

PRIMARY KEY (`group\_id`)

);

CREATE TABLE `users` (

`user\_id` int(10) NOT NULL AUTO\_INCREMENT,

`username` varchar(10) NOT NULL,

`first\_name` varchar(20) DEFAULT NULL,

`middle\_name` varchar(20) DEFAULT NULL,

`last\_name` varchar(20) DEFAULT NULL,

`password` char(32) NOT NULL,

PRIMARY KEY (`user\_id`)

);

CREATE TABLE `user\_groups` (

`user\_id` int(10) NOT NULL,

`group\_id` int(10) NOT NULL,

PRIMARY KEY (`user\_id`,`group\_id`),

KEY `fk\_users\_has\_groups\_groups1` (`group\_id`),

KEY `fk\_users\_has\_groups\_users` (`user\_id`),

CONSTRAINT `fk\_groups` FOREIGN KEY (`group\_id`) REFERENCES `groups` (`group\_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_users` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`) ON DELETE NO ACTION ON UPDATE NO ACTION

);

CREATE VIEW `v\_user\_role` AS

SELECT u.username, u.password, g.group\_name

FROM `user\_groups` ug

INNER JOIN `users` u ON u.user\_id = ug.user\_id

INNER JOIN `groups` g ON g.group\_id = ug.group\_id;

INSERT INTO `groups`(`group\_id`,`group\_name`,`group\_desc`) VALUES

(1,'CABBIE','Cabbie users'),

(2,'ADMIN','Administration users');

INSERT INTO `users`(`user\_id`,`username`,`first\_name`,`middle\_name`,`last\_name`,`password`) VALUES

(1,'driver','Василий','Петрович','Иванов','6e0b7076126a29d5dfcbd54835387b7b'), /\*john123\*/

(2,'admin',NULL,NULL,NULL,'21232f297a57a5a743894a0e4a801fc3'); /\*admin\*/

INSERT INTO `user\_groups`(`user\_id`,`group\_id`) VALUES (1,1),(2,1),(2,2);

CREATE TABLE cars(

car\_id INTEGER NOT NULL PRIMARY KEY,

brand CHAR(50) NOT NULL,

color CHAR(50) NOT NULL,

number CHAR(8) NOT NULL,

taxist\_id INTEGER NOT NULL,

FOREIGN KEY (taxist\_id) REFERENCES taxists(taxist\_id));

CREATE TABLE taxists(

taxist\_id INTEGER NOT NULL PRIMARY KEY,

status\_taxi INTEGER NOT NULL,

user\_id INTEGER NOT NULL,

FOREIGN KEY (user\_id) REFERENCES users(user\_id));

CREATE TABLE orders(

order\_id INTEGER NOT NULL PRIMARY KEY,

surname CHAR(30) NOT NULL,

phonenumber CHAR(11) NOT NULL,

dateOfOrdering DATETIME NOT NULL,

status\_order INTEGER NOT NULL,

street CHAR(20) NOT NULL,

house INTEGER NOT NULL,

ponch INTEGER NOT NULL,

comment NVARCHAR(1000) NOT NULL);

CREATE TABLE firms (

fir\_id INTEGER NOT NULL PRIMARY KEY,

name\_of\_firm CHAR(50) NOT NULL,

tax\_id INTEGER NOT NULL,

car\_id INTEGER NOT NULL

FOREIGN KEY (tax\_id) REFERENCES taxists(taxist\_id));

FOREIGN KEY (car\_id) REFERENCES cars(car\_id));

CREATE TABLE buhgalteria (

buh\_id INTEGER NOT NULL PRIMARY KEY,

name\_of\_doh\_and\_ras CHAR(50) NOT NULL,

summ\_of\_doh INTEGER NOT NULL,

summ\_of\_ ras INTEGER NOT NULL,

date DATE INTEGER NOT NULL,

CREATE TABLE buhgalters (

buhg\_id INTEGER NOT NULL PRIMARY KEY,

buh\_id INEGER NOT NULL,

user\_id INTEGER NOT NULL,

FOREIGN KEY (buh\_id) REFERENCES buhgalteria(buh\_id));

FOREIGN KEY (user\_id) REFERENCES users(user\_id));

INSERT INTO `showcase`.`users` (`user\_id`, `username`, `first\_name`, `middle\_name`, `last\_name`, `password`) VALUES ('3', 'cabbie2', 'Василий', 'Андреевич', 'Бородин', 'e10adc3949ba59abbe56e057f20f883e');

INSERT INTO `showcase`.`users` (`user\_id`, `username`, `first\_name`, `middle\_name`, `last\_name`, `password`) VALUES ('4', 'cabbie3', 'Петр', 'Михайлович', 'Федосов', 'e10adc3949ba59abbe56e057f20f883e');

INSERT INTO `showcase`.`users` (`user\_id`, `username`, `first\_name`, `middle\_name`, `last\_name`, `password`) VALUES ('5', 'cabbie4', 'Андрей', 'Будимирович', 'Васильев', 'e10adc3949ba59abbe56e057f20f883e');

INSERT INTO `showcase`.`user\_groups` (`user\_id`, `group\_id`) VALUES ('3', '1');

INSERT INTO `showcase`.`user\_groups` (`user\_id`, `group\_id`) VALUES ('4', '1');

INSERT INTO `showcase`.`user\_groups` (`user\_id`, `group\_id`) VALUES ('5', '1');

INSERT INTO `showcase`.`taxists` (`taxist\_id`, `status\_taxi`, `user\_id`) VALUES ('1', '1', '1');

INSERT INTO `showcase`.`taxists` (`taxist\_id`, `status\_taxi`, `user\_id`) VALUES ('2', '1', '3');

INSERT INTO `showcase`.`taxists` (`taxist\_id`, `status\_taxi`, `user\_id`) VALUES ('3', '1', '4');

INSERT INTO `showcase`.`taxists` (`taxist\_id`, `status\_taxi`, `user\_id`) VALUES ('4', '1', '5');

INSERT INTO `showcase`.`cars` (`car\_id`, `brand`, `color`, `number`, `taxist\_id`) VALUES ('1', 'Mazda', 'Красный', 'о258уа', '1');

INSERT INTO `showcase`.`cars` (`car\_id`, `brand`, `color`, `number`, `taxist\_id`) VALUES ('2', 'Nissan', 'Зеленый', 'о040оо', '2');

INSERT INTO `showcase`.`cars` (`car\_id`, `brand`, `color`, `number`, `taxist\_id`) VALUES ('3', 'VW', 'Черный', 'п412еа', '3');

INSERT INTO `showcase`.`cars` (`car\_id`, `brand`, `color`, `number`, `taxist\_id`) VALUES ('4', 'Daewoo', 'Белый', 'а305хх', '4');

ALTER TABLE `showcase`.`orders`

CHANGE COLUMN `comment` `comment` VARCHAR(1000) NULL ,

ADD COLUMN `way` VARCHAR(100) NOT NULL AFTER `comment`;

ALTER TABLE `showcase`.`orders`

ADD COLUMN `taxist\_id` INT NULL AFTER `way`;

# 6. СТРУКТУРА БИЗНЕС-ЛОГИКИ СИСТЕМЫ.

## 6.1. Функции, необходимые программной системе.

Список функций, необходимых программной системе:

1)Вывести список всех заказов.

2)Вывести список всех таксистов.

3)Вывести список всех машин.

4)Редактирование информации о таксисте.

5)Редактирование информации о машинах.

6)Обновить статус заказа.

7)Удалить таксиста.

9)Зарегистрировать нового таксиста.

10)Проверка логина и пароля.

11) Расчет денег, заработанных фирмой.

## 

## 6.2. Диаграмма прецедентов.

Диаграмма прецедентов ИС “Такси”



**Актеры:**

**Клиент** – пользователь, который имеет возможность вызвать такси.

**Пользователь** – авторизованный пользователь в системе.

**Таксист** – авторизованный пользователь в системе, с ролью таксист. Работает с заказами.

**Гость** – неавторизованный пользователь в системе.

**Администратор** – авторизованный пользователь в системе, отвечающий за регистрацию таксистов, редактирование информации о таксистах и машинах, а также расчет доходов и расходов.

**Прецеденты :**

**Название прецедента**: Авторизоваться.

**Предусловие**: Пользователь имеет логин и пароль.

**Действующее лицо**: Пользователь.

**Основной поток**: Пользователь проходит процедуру аутентификации, указав свои данные: логин, пароль. Далее пользователю присваивается роль таксист или администратор.

**Альтернативный поток**:

Введены некорректные данные. Повторяется процедура аутентификации;

**Постусловие**:

Если клиент правильно ввёл данные, то выводится страница приветствия.

**Название прецедента**: Работа со справочником таксиста.

**Предусловие**: Пользователь авторизован и имеет роль администратор.

**Действующее лицо**: Администратор.

**Основной поток**: Пользователь авторизован в системе и имеет роль администратор. Администратор может редактировать информацию о таксистах, а также добавить нового.

**Альтернативный поток**:

Отказ от редактирования или создания таксиста.

**Постусловие**:

Если администратор правильно ввёл данные, то выводится редактируется информация о таксисте или создается новая учетная запись таксиста.

**Название прецедента:**Просмотр списка заказов.

**Предусловие:**Пользователь зашел в систему с ролью «Таксист».

**Действующее лицо:**Таксист.

**Основной поток:**Пользователь зашел в систему с ролью «Таксист». Он хочет получить данные о всех текущих заказах. Для этого он нажимает на кнопку «показать список заказов».

**Альтернативный поток:**В системе нет активных заказов

**Название прецедента:**Подать заказ.

**Предусловие:**Пользователь зашел в систему с ролью «Клиент».

**Действующее лицо:**Клиент.

**Основной поток:**Пользователь зашел в систему с ролью «Таксист». Он хочет сделать заказ такси, для этого он заполняет необходимые поля формы и нажимает на кнопку «Подать заказ».

**Альтернативный поток:**Клиент неверно заполнил поля формы, ему предлогается исправить неверные поля или отказаться от заявки.

## 6.3. Диаграмма состояний объекта «Заявление» .



Заявление имеет 3 возможных состояния : в очереди, то есть подано; подветрждено , то

есть таксист взял заказ на выполнение; заявление удалено. Клиент может удалить свое

заявление пока таксист не взял данное заявление на выполнение.

## 6.4. Диаграммы последовательностей .

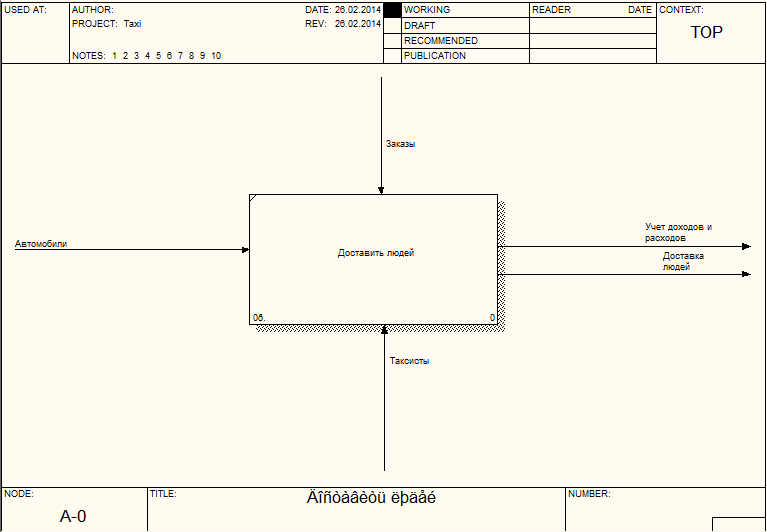
Диаграмма последовательностей для объекта «Подача заявления»



Диаграмма последовательностей для объекта «Получить конкретное заявление»



## 6.4. Диаграмма IDEF0.



## 6.5. Диаграмма деятельности .



## 6.6. Концептуальная модель диаграммы классов.

## 

## 6.2. Диаграмма классов.

Диаграмма классов представлена на Рис.3.

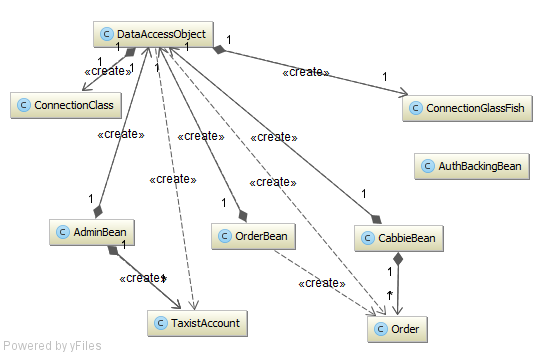
****

Рис.3. Диаграмма классов.

## 6.3.Словесное описание сущностей, описание атрибутов, классов и функций.

**6.3.1.Package model.**

1. ***Order***– класс, содержащий данные о заказе (Рис.4).

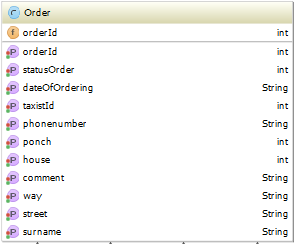


Рис.4.Class order

***Свойства****:*

-orderID : int – идентификационный номер заказа;

-statusOrder : int – статус заказа;

-dateOfOrdering : String – дата заказа;

-taxistID : String – идентификационный номер таксиста,выполняющего заказ;

-phonenumber : String – телефонный номер клиента;

-ponch : int – номер подъезда,откуда поедет клиент;

-house: int – номер дома,откуда поедет клиент;

-comment : String – комментарий к заказу;

-way: String – место,куда поедет клиент ;

-street: String – улица,откуда поедет клиент;

-surname: String – фамилия клиента;

1. ***Car*–** класс, содержащий данные о машинах таксистов (Рис.5).

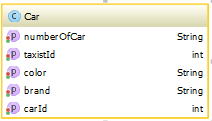


Рис.5.Class Car

***Свойства:***

-numberOfCar : String – номер машин;

-taxistID : int – идентификационный номер таксиста;

-carID : int – идентификационный номер машины;

-color: String – цвет машины;

- brand: String – марка машины;

**3) *User* –** класс, содержащий данные о пользователях (Рис.6).

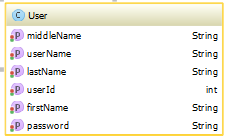


Рис.6.Class User.

***Свойства:***

-userId : int – идентификационный номер пользователя;

-firstName : String – имя пользователя;

-middleName : String – отчество пользователя;

-lastName : String – фамилия пользователя;

-userName : String – логин пользователя;

-password:String – пароль пользователя;

**4.TaxistAccount –** класс, обьединяющий все данные, необходимые таксисту(Рис.7).

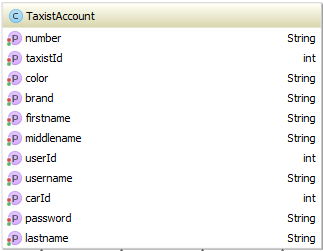


Рис.7.Class Group.

***Свойства:***

-taxistID : int – идентификационный номер таксиста;

-firstname:String- имя таксиста;

-middlename:String- отчество таксиста;

-lastname:String – фамилия таксиста;

-number:String- номер машины таксиста;

-color:String – цвет машины таксиста;

-brand:String – марка машины таксиста;

-userID:int – идентификационный номер пользователя;

-username:String – логин для входа в систему;

-password:String – пароль для входа в систему

-carId:int – идентификационный номер машины таксиста;

**6.3.2. Package DAO.**

1. **1) *ConnectionGlassFish* –** класс, соединения с базой данных. Соединение

берется из сервера приложений GlassFish .



Рис.8. Class ConnectionGlassFish.

***Атрибуты:***

- dataSource : DataSource – объект класса DataSource;

***Функции:***

+ createConnection() : DataSource – возвращает DataSource, содержащее данные

о соединении;

1. ***DataAccesObject* –** класс, содержащий методы бизнес-логики системы. (Рис.8).

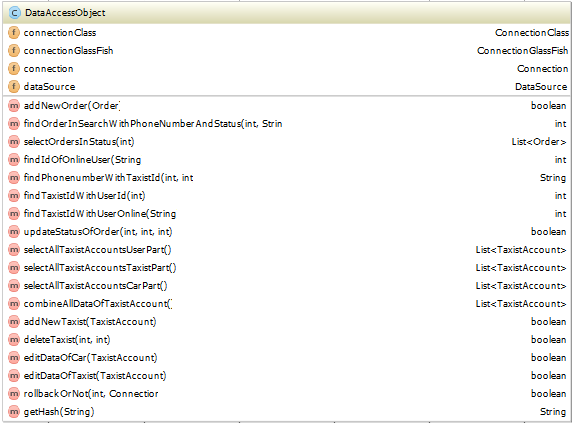


Рис.8. Class DataAccesObject.

***Атрибуты:***

**-** connection : Connection – объект соединения с базой данных.

**-**connectionGlassfish : ConnectionGlassFish – объект класса ConnectionGlassFish;

- dataSource : DataSource – объект класса DataSource;

***Функции:***

-addNewOrder(Order order) – добавляет новый заказ.

-findOrderInSearchWhithPhoneNumberandStatus(int status, String phonenumber)-

поиск заказа по номеру телефона и статусу.

-public List<Order> selectOrdersInStatus(int status)- выбор заказа с определенным статусом.

- public int findIdOfOnlineUser(String username) – Поиск идентификационного номера пользователей,которые в данный момент находятся в системе.

- public String findPhonenumberWithTaxistId(int taxistId,int statusOrder)-

- public int findTaxistIdWithUserId(int userId)- поиск таксиста по идентификационному номеру пользователя.

-public int findTaxistIdWithUserOnline(String username) – поиск таксиста в списке пользователей,которые онлайн.

- public boolean updateStatusOfOrder(int orderId,int status,int taxistId) – обновить статус заказа.

- public List<TaxistAccount> selectAllTaxistAccountsUserPart() – возвращает список всех пользователей.

- public List<TaxistAccount> selectAllTaxistAccountsTaxistPart()-возвращает список таксистов.

- public List<TaxistAccount> selectAllTaxistAccountsCarPart()- возвращает список машин.

- public List<TaxistAccount> combineAllDataOfTaxistAccount() –возвращает данные из таблицы машин,пользователей и таксистов.

- public boolean addNewTaxist(TaxistAccount taxistAccount) – добавить нового таксиста.

- public boolean deleteTaxist(int userID,int taxistID) – удалить таксиста.

- public boolean editDataOfCar(TaxistAccount taxistAccount) – редактировать информацию о машине.

- public boolean editDataOfTaxist(TaxistAccount taxistAccount ) – редактировать информацию о таксисте.

- public String getHash(String password) – шифрование пароля.

**6.3.3. Package beans.**

1. ***CabbieBean*–** бин контекста сессии, использующийся для работы с данными о таксистах и их заказах (Рис.9).

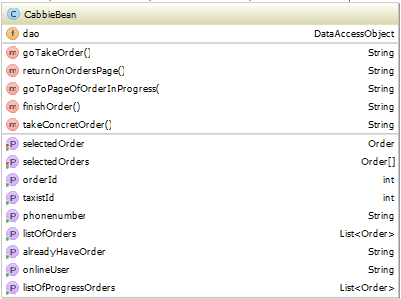


Рис.9. Class CabbieBean.

***Атрибуты:***

dao- объект класса DataAccesObject;

***Функции:***

-goTakeOrder() – просмотреть какие существуют возможные заказы в данный момент.

-returnOnOrderPage() – вернуться на страницу заказов.

-goToPageOfOrderInProgress() –вернуться на страницу выполнения заказов.

-finishOrder() – выполнить заказ.

-takeConcretOrder() – взять заказ на выполнение.

***Свойства:***

-selectedOrder:Order – определенная запись заказа;

-selectedOrders:Order[] – определенные записи заказа;

-orderid:int –идентификационный номер заказа;

-taxistid:int – идентификационный номер таксиста;

-phonenumber – телефонный номер клиента;

-listofOrders:List<Order> - список записей заказов;

-alreadyHaveOrder:String – уже имеет заказ;

-onlineUser:String – пользователь,который в данный момент Online.

-listOfProgressOrders:List<Order> - список,выполняющихся заказов.

1. ***OrderBean –*** бин, позволяющий добавить заказ (Рис.10).

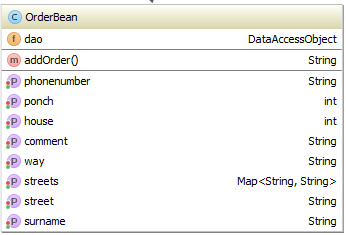


Рис.10.Class ConcretSessionBean.

***Атрибуты:***

dao- объект класса DataAccesObject;

***Функции:***

-addOrder() – добавляет новый заказ.

***Свойства:***

-phonenumber:string – телефонный номер клиента;

-ponch:int – номер подъезда клиента.

-house:int – номер дома клиента.

-comment:string – комментарий к заказу.

-way:string – путь,куда необходимо ехать.

-street: string – улица,на которой находится клиент.

-streets:Map<String,String> - список улиц.

-surname: фамилия клиента.

1. ***AdminBean* –** бин,содержащий методы,используемые администратором (Рис.11).

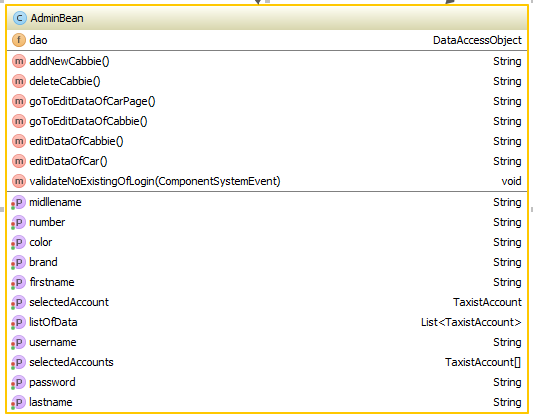


Рис.11. Class AdminBean.

***Атрибуты:***

dao- объект класса DataAccesObject;

***Функции:***

-addnewCabbie() – создавать нового пользователя таксиста.

-deleteCabbie() – удалить таксиста.

- goToEditDataOfCarPage() – перейти на страницу редактирование данных о машине таксиста.

- goToEditDataOfCabbie() – перейти на страницу редактирования данных о таксисте.

-editDataOfCabbie() – редактирование данных о таксисте.

-editDataOfCar() – редактирование данных о машине.

-validateNoExistingOfLogin(ComponentSystemEvent event) – валидация на создание пользователя с уже существущим логином.

***Свойства:***

-firstname:String- имя таксиста;

-middlename:String- отчество таксиста;

-lastname:String – фамилия таксиста;

-number:String- номер машины таксиста;

-color:String – цвет машины таксиста;

-brand:String – марка машины таксиста;

-username:String – логин для входа в систему;

-password:String – пароль для входа в систему;

-listofdata:List<TaxistAccount>- список таксистов;

-selectedAccount:TaxistAccount – определенная запись таксиста;

-selectedAccounts:TaxistAccount[] – определенные записи таксиста;

1. ***AuthBackingBean –*** бин, выполняющий выход из системы определенного пользователя(Рис. 12).

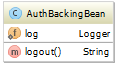
****

Рис.12. Class AutoBackingBean.

***Функции:***

-logout() – позволяет выйти из системы.

## 6.4.Объяснение связей диаграммы классов.

- зависимость

Один класс выступает в качестве параметра или локальной переменной другого класса.

 - композиция

Более сильный вариант агрегации (один класс содержит объекты другого класса). Если уничтожается композитор, то его объекты классов на которые он ссылается также перестают существовать.

# 7. РЕАЛИЗАЦИЯ БИЗНЕС-ЛОГИКИ ПРОТОТИПА.

# 7.1 Package Model

# 7.1.1. Car.java

Car.java

public class Car

{

private int carId;

private String brand;

private String color;

private String numberOfCar;

private int taxistId;

public int getCarId() {

return carId;

}

public void setCarId(int carId) {

this.carId = carId;

}

public String getBrand() {

return brand;

}

public void setBrand(String brand) {

this.brand = brand;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public String getNumberOfCar() {

return numberOfCar;

}

public void setNumberOfCar(String numberOfCar) {

this.numberOfCar = numberOfCar;

}

public int getTaxistId() {

return taxistId;

}

public void setTaxistId(int taxistId) {

this.taxistId = taxistId;

}

}

## 7.1.2 Group.java

public class Group

{

private int id;

private int name;

private String description;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getName() {

return name;

}

public void setName(int name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

}

## 7.1.3 Order.java

public class Order

{

private int orderId;

private String surname;

private String phonenumber;

private String dateOfOrdering;

private int statusOrder;

private String street;

private int house;

private int ponch;

private String comment;

private String way;

private int taxistId;

public Order(){}

public Order(int orderId, String surname, String phonenumber, String dateOfOrdering, int statusOrder, String street, int house, int porch, String comment, String way)

{

this.orderId = orderId;

this.surname = surname;

this.phonenumber = phonenumber;

this.dateOfOrdering = dateOfOrdering;

this.statusOrder = statusOrder;

this.street = street;

this.house = house;

this.ponch = porch;

this.comment = comment;

this.way = way;

}

public int getTaxistId() {

return taxistId;

}

public void setTaxistId(int taxistId) {

this.taxistId = taxistId;

}

public String getDateOfOrdering() {

return dateOfOrdering;

}

public void setDateOfOrdering(String dateOfOrdering) {

this.dateOfOrdering = dateOfOrdering;

}

public int getOrderId() {

return orderId;

}

public void setOrderId(int orderId) {

orderId = orderId;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public String getPhonenumber() {

return phonenumber;

}

public void setPhonenumber(String phonenumber) {

this.phonenumber = phonenumber;

}

public int getStatusOrder() {

return statusOrder;

}

public void setStatusOrder(int statusOrder) {

this.statusOrder = statusOrder;

}

public String getStreet() {

return street;

}

public void setStreet(String street) {

this.street = street;

}

public int getHouse() {

return house;

}

public void setHouse(int house) {

this.house = house;

}

public int getPonch() {

return ponch;

}

public void setPonch(int ponch) {

this.ponch = ponch;

}

public String getComment() {

return comment;

}

public void setComment(String comment) {

this.comment = comment;

}

public String getWay() {

return way;

}

public void setWay(String way) {

this.way = way;

}

}

# 7.1.4. Taxist.java

public class Taxist

{

private int taxistId;

private int userId;

public int getTaxistId() {

return taxistId;

}

public void setTaxistId(int taxistId) {

this.taxistId = taxistId;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

}

TaxistAccount.java

public class TaxistAccount

{

private int userId;

private String username;

private String lastname;

private String firstname;

private String middlename;

private String password;

private int carId;

private int taxistId;

private String brand;

private String color;

private String number;

public TaxistAccount(){}

public TaxistAccount(String username,String lastname,String firstname,String middlename,

String password,String brand,String color,String number)

{

this.username = username;

this.lastname = lastname;

this.firstname = firstname;

this.middlename = middlename;

this.password = password;

this.brand = brand;

this.color = color;

this.number = number;

}

public int getCarId() {

return carId;

}

public void setCarId(int carId) {

this.carId = carId;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getLastname() {

return lastname;

}

public void setLastname(String lastname) {

this.lastname = lastname;

}

public String getFirstname() {

return firstname;

}

public void setFirstname(String firstname) {

this.firstname = firstname;

}

public String getMiddlename() {

return middlename;

}

public void setMiddlename(String middlename) {

this.middlename = middlename;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public int getTaxistId() {

return taxistId;

}

public void setTaxistId(int taxistId) {

this.taxistId = taxistId;

}

public String getBrand() {

return brand;

}

public void setBrand(String brand) {

this.brand = brand;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

}

## 7.1.5. User.java

public class User

{

private int userId;

private String userName;

private String firstName;

private String middleName;

private String lastName;

private String password;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getMiddleName() {

return middleName;

}

public void setMiddleName(String middleName) {

this.middleName = middleName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

## 7.1.6. UserGroup.java

public class UserGroup

{

private int userId;

private int groupId;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public int getGroupId() {

return groupId;

}

public void setGroupId(int groupId) {

this.groupId = groupId;

}

}

## 7.2.Package DAO.

***7.2.1. ConnectionClass.java***

package DAO;

import org.apache.commons.dbcp.BasicDataSource;

import java.sql.Connection;

public class ConnectionClass

{

Connection connection = null;

BasicDataSource bdSource = new BasicDataSource();

public ConnectionClass()

{

bdSource.setDriverClassName("com.mysql.jdbc.Driver");

bdSource.setUrl("jdbc:mysql://localhost:3306/showcase");

bdSource.setUsername("artyom");

bdSource.setPassword("08041993");

}

public Connection createConnection()

{

Connection con = null;

try

{

if (connection != null)

{

System.out.println("Cant create a New Connection");

}

else

{

con = bdSource.getConnection();

System.out.println("Connection Done successfully");

}

} catch (Exception e) {

System.out.println("Error Occured " + e.toString());

}

return con;

}

}

***7.2.2. DataAccesObject***

package DAO;

import model.Order;

import model.TaxistAccount;

import javax.naming.NamingException;

import javax.sql.DataSource;

import java.io.UnsupportedEncodingException;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import java.util.logging.Level;

import java.util.logging.Logger;

public class DataAccessObject

{

ConnectionClass connectionClass = new ConnectionClass();

ConnectionGlassFish connectionGlassFish;

Connection connection = null;

DataSource dataSource = null;

public DataAccessObject() throws NamingException

{

connectionGlassFish = new ConnectionGlassFish();

}

public boolean addNewOrder(Order order) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

int orderID;

Statement stmt = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet result = stmt.executeQuery("SELECT \* FROM orders");

result.last();

orderID = result.getInt("order\_id");

result.moveToInsertRow();

result.updateInt("order\_id",orderID+1);

result.updateString("surname",order.getSurname());

result.updateString("phonenumber",order.getPhonenumber());

java.sql.Timestamp timestamp = new java.sql.Timestamp(System.currentTimeMillis());

result.updateTimestamp("dateOfOrdering",timestamp);

result.updateInt("status\_order",2);

result.updateString("street",order.getStreet());

result.updateInt("house",order.getHouse());

result.updateInt("ponch",order.getPonch());

result.updateString("comment",order.getComment());

result.updateString("way",order.getWay());

result.insertRow();

stmt.close();

return true;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return false;

}

finally

{

connection.close();

}

}

public int findOrderInSearchWithPhoneNumberAndStatus(int status, String phonenumber) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT order\_id FROM orders WHERE status\_order = ? AND phonenumber = ?");

stmt.setInt(1,status);

stmt.setString(2,phonenumber);

ResultSet result = stmt.executeQuery();

int orderId = 0;

while (result.next())

{

orderId = result.getInt(1);

}

stmt.close();

return orderId;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return 0;

}

finally

{

connection.close();

}

}

/\*

ID становятся нулями, хрен знает почему!!!

\*/

public List<Order> selectOrdersInStatus(int status) throws SQLException

{

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT \* FROM orders WHERE status\_order = ?");

stmt.setInt(1,status);

ResultSet result = stmt.executeQuery();

List<Order> listOfOrders = new ArrayList<Order>();

while (result.next())

{

Order order = new Order();

order.setOrderId(result.getInt(1));

order.setSurname(result.getString(2));

order.setPhonenumber(result.getString(3));

String dateOfOrdering = result.getTimestamp(4).toString();

order.setDateOfOrdering(dateOfOrdering.substring(0,dateOfOrdering.length()-5));

order.setStatusOrder(result.getInt(5));

order.setStreet(result.getString(6));

order.setHouse(result.getInt(7));

order.setPonch(result.getInt(8));

order.setComment(result.getString(9));

order.setWay(result.getString(10));

order.setTaxistId(result.getInt(11));

listOfOrders.add(order);

}

stmt.close();

return listOfOrders;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return null;

}

finally

{

connection.close();

}

}

public int findIdOfOnlineUser(String username) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT \* FROM users WHERE username = ?");

stmt.setString(1,username);

int userId = 0;

ResultSet result = stmt.executeQuery();

while (result.next())

{

userId = result.getInt(1);

}

stmt.close();

return userId;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return 0;

}

finally

{

connection.close();

}

}

public String findPhonenumberWithTaxistId(int taxistId,int statusOrder) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT phonenumber FROM orders WHERE taxist\_id = ? AND status\_order = ?");

stmt.setInt(1,taxistId);

stmt.setInt(2,statusOrder);

String phonenumber = null;

ResultSet result = stmt.executeQuery();

while (result.next())

{

phonenumber = result.getString(1);

}

return phonenumber;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return null;

}

finally

{

connection.close();

}

}

public int findTaxistIdWithUserId(int userId) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT \* FROM taxists WHERE user\_id = ?");

stmt.setInt(1, userId);

int taxistId;

ResultSet result = stmt.executeQuery();

while (result.next())

{

taxistId = result.getInt(1);

}

stmt.close();

return userId;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return 0;

}

finally

{

connection.close();

}

}

public int findTaxistIdWithUserOnline(String username) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

int taxistId = 0;

int userId = findIdOfOnlineUser(username);

taxistId = findTaxistIdWithUserId(userId);

return taxistId;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return 0;

}

finally

{

connection.close();

}

}

public boolean updateStatusOfOrder(int orderId,int status,int taxistId) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("UPDATE orders SET status\_order = ?, " +

"taxist\_id = ? WHERE order\_id = ?");

stmt.setInt(1, status);

stmt.setInt(2, taxistId);

stmt.setInt(3, orderId);

int result = stmt.executeUpdate();

if (result>=0) return true;

return false;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return false;

}

finally

{

connection.close();

}

}

public List<TaxistAccount> selectAllTaxistAccountsUserPart() throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT \* FROM users");

ResultSet result = stmt.executeQuery();

List<TaxistAccount> listOfData = new ArrayList<TaxistAccount>();

while (result.next())

{

TaxistAccount taxAc = new TaxistAccount();

taxAc.setUserId(result.getInt(1));

taxAc.setUsername(result.getString(2));

taxAc.setFirstname(result.getString(3));

taxAc.setMiddlename(result.getString(4));

taxAc.setLastname(result.getString(5));

taxAc.setPassword(result.getString(6));

listOfData.add(taxAc);

}

return listOfData;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return null;

}

finally

{

connection.close();

}

}

public List<TaxistAccount> selectAllTaxistAccountsTaxistPart() throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT \* FROM taxists");

ResultSet result = stmt.executeQuery();

List<TaxistAccount> listOfData = new ArrayList<TaxistAccount>();

while (result.next())

{

TaxistAccount taxAc = new TaxistAccount();

taxAc.setTaxistId(result.getInt(1));

taxAc.setUserId(result.getInt(2));

listOfData.add(taxAc);

}

return listOfData;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return null;

}

finally

{

connection.close();

}

}

public List<TaxistAccount> selectAllTaxistAccountsCarPart() throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("SELECT \* FROM cars");

ResultSet result = stmt.executeQuery();

List<TaxistAccount> listOfData = new ArrayList<TaxistAccount>();

while (result.next())

{

TaxistAccount taxAc = new TaxistAccount();

taxAc.setCarId(result.getInt(1));

taxAc.setBrand(result.getString(2));

taxAc.setColor(result.getString(3));

taxAc.setNumber(result.getString(4));

taxAc.setTaxistId(result.getInt(5));

listOfData.add(taxAc);

}

return listOfData;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return null;

}

finally

{

connection.close();

}

}

public List<TaxistAccount> combineAllDataOfTaxistAccount() throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

List<TaxistAccount> userPart = selectAllTaxistAccountsUserPart();

List<TaxistAccount> taxistPart = selectAllTaxistAccountsTaxistPart();

List<TaxistAccount> carPart = selectAllTaxistAccountsCarPart();

List<TaxistAccount> resultList = new ArrayList<TaxistAccount>();

resultList = userPart;

for (int i = 0; i < resultList.size(); i++)

{

if (resultList.get(i).getUsername().equals("admin"))

resultList.remove(i);

}

for (int i = 0; i < resultList.size(); i++)

{

for (int j = 0; j < taxistPart.size(); j++)

{

if (resultList.get(i).getUserId()==taxistPart.get(i).getUserId())

{

resultList.get(i).setTaxistId(taxistPart.get(i).getTaxistId());

}

}

}

for (int i = 0; i < resultList.size(); i++)

{

for (int j = 0; j < carPart.size(); j++)

{

if (resultList.get(i).getTaxistId() == carPart.get(i).getTaxistId())

{

resultList.get(i).setCarId(carPart.get(i).getCarId());

resultList.get(i).setBrand(carPart.get(i).getBrand());

resultList.get(i).setColor(carPart.get(i).getColor());

resultList.get(i).setNumber(carPart.get(i).getNumber());

}

}

}

return resultList;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return null;

}

finally

{

connection.close();

}

}

public boolean addNewTaxist(TaxistAccount taxistAccount) throws SQLException {

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

int userID;

Statement stmt = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet result = stmt.executeQuery("SELECT \* FROM users");

result.last();

userID = result.getInt("user\_id");

result.moveToInsertRow();

result.updateInt("user\_id",userID + 1);

result.updateString("username",taxistAccount.getUsername());

result.updateString("first\_name",taxistAccount.getFirstname());

result.updateString("middle\_name",taxistAccount.getMiddlename());

result.updateString("last\_name",taxistAccount.getLastname());

result.updateString("password",getHash(taxistAccount.getPassword()));

result.insertRow();

Statement stmt2 = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet result2 = stmt2.executeQuery("SELECT \* FROM user\_groups");

result2.last();

result2.moveToInsertRow();

result2.updateInt("user\_id",userID + 1);

result2.updateInt("group\_id",1);

result2.insertRow();

int taxistID;

Statement stmt3 = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet result3 = stmt3.executeQuery("SELECT \* FROM taxists");

result3.last();

taxistID = result3.getInt("taxist\_id");

result3.moveToInsertRow();

result3.updateInt("taxist\_id",taxistID+1);

result3.updateInt("user\_id",userID+1);

result3.insertRow();

int carId;

Statement stmt4 = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet result4 = stmt4.executeQuery("SELECT \* FROM cars");

result4.last();

carId = result4.getInt("car\_id");

result4.moveToInsertRow();

result4.updateInt("car\_id",carId+1);

result4.updateString("brand",taxistAccount.getBrand());

result4.updateString("color",taxistAccount.getColor());

result4.updateString("number",taxistAccount.getNumber());

result4.updateInt("taxist\_id",taxistID+1);

result4.insertRow();

stmt.close();

return true;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return false;

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

return false;

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

return false;

}

finally

{

connection.close();

}

}

public boolean deleteTaxist(int userID,int taxistID) throws SQLException

{

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt = connection.prepareStatement("DELETE FROM user\_groups WHERE user\_id = ?");

stmt.setInt(1,userID);

stmt.executeUpdate();

stmt.close();

PreparedStatement stmt2 = connection.prepareStatement("DELETE FROM cars WHERE taxist\_id = ?");

stmt2.setInt(1,taxistID);

stmt2.executeUpdate();

stmt2.close();

PreparedStatement stmt3 = connection.prepareStatement("DELETE FROM taxists WHERE user\_id = ?");

stmt3.setInt(1,userID);

stmt3.executeUpdate();

stmt3.close();

PreparedStatement stmt4 = connection.prepareStatement("DELETE FROM users WHERE user\_id = ?");

stmt4.setInt(1,userID);

stmt4.executeUpdate();

stmt4.close();

return true;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return false;

}

finally

{

connection.close();

}

}

public boolean editDataOfCar(TaxistAccount taxistAccount) throws SQLException

{

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("UPDATE cars SET brand = ?, color = ?, " +

"number = ? WHERE taxist\_id = ?");

stmt.setString(1,taxistAccount.getBrand());

stmt.setString(2,taxistAccount.getColor());

stmt.setString(3,taxistAccount.getNumber());

stmt.setInt(4,taxistAccount.getTaxistId());

int result = stmt.executeUpdate();

if (result!=0) return true;

return false;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return false;

}

finally

{

connection.close();

}

}

public boolean editDataOfTaxist(TaxistAccount taxistAccount) throws SQLException

{

//connection = connectionClass.createConnection();

dataSource = connectionGlassFish.createConnection();

connection = dataSource.getConnection();

try

{

PreparedStatement stmt;

stmt = connection.prepareStatement("UPDATE users SET username = ?, first\_name = ?, " +

"middle\_name = ?, last\_name = ?, password = ? WHERE user\_id = ?");

stmt.setString(1,taxistAccount.getUsername());

stmt.setString(2,taxistAccount.getFirstname());

stmt.setString(3,taxistAccount.getMiddlename());

stmt.setString(4,taxistAccount.getLastname());

stmt.setString(5,getHash(taxistAccount.getPassword()));

stmt.setInt(6,taxistAccount.getUserId());

int result = stmt.executeUpdate();

if (result!=0) return true;

return false;

}

catch (SQLException ex)

{

Logger.getLogger(DataAccessObject.class.getName()).log(Level.SEVERE, null, ex);

return false;

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

return false;

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

return false;

} finally

{

connection.close();

}

}

public boolean rollbackOrNot(int num, Connection connection) throws SQLException

{

if (num == 0)

{

connection.rollback();

return false;

}

return true;

}

public String getHash(String password) throws NoSuchAlgorithmException, UnsupportedEncodingException

{

MessageDigest md = MessageDigest.getInstance("MD5");

byte[] hash = md.digest(password.getBytes("UTF-8")); //converting byte array to Hexadecimal String

StringBuilder sb = new StringBuilder(2\*hash.length);

for(byte b : hash)

{

sb.append(String.format("%02x", b&0xff));

}

return sb.toString();

}

}

***7.2.3. ConnectionGlassFish.java***

package DAO;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.sql.DataSource;

public class ConnectionGlassFish

{

private DataSource dataSource;

public ConnectionGlassFish() throws NamingException

{

Context ctx = new InitialContext();

dataSource = (DataSource)ctx.lookup("java:comp/env/jdbc/showcase");

}

public DataSource createConnection()

{

try

{

return dataSource;

}

catch (Exception e)

{

e.printStackTrace();

return null;

}

}

}

## 7.3.Package beans.

***7.3.1. AdminBean.java***

package beans;

import DAO.DataAccessObject;

import model.TaxistAccount;

import javax.faces.application.FacesMessage;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.SessionScoped;

import javax.faces.component.UIComponent;

import javax.faces.component.UIInput;

import javax.faces.context.FacesContext;

import javax.faces.event.ComponentSystemEvent;

import javax.naming.NamingException;

import javax.validation.constraints.Size;

import java.sql.SQLException;

import java.util.List;

@ManagedBean

@SessionScoped

public class AdminBean

{

private List<TaxistAccount> listOfData;

private DataAccessObject dao;

@Size(min = 3,max = 10,message = "Логин должен быть от 3 до 10 символов")

private String username;

@Size(min = 3,max = 20,message = "Фамилия должна быть от 3 до 20 символов")

private String lastname;

@Size(min = 3,max = 20,message = "Имя должно быть от 3 до 20 символов")

private String firstname;

@Size(min = 3,max = 20,message = "Отчество должно быть от 3 до 20 символов")

private String midllename;

@Size(min = 3,max = 40,message = "Пароль должен быть от 3 до 40 символов")

private String password;

@Size(min = 3,max = 50,message = "Марка автомобиля должна быть от 3 до 50 символов")

private String brand;

@Size(min = 3,max = 50,message = "Цвет автомобиля должен быть от 3 до 50 символов")

private String color;

@Size(min = 6,max = 6,message = "Номер автомобиля должен быть 6 символов")

private String number;

TaxistAccount[] selectedAccounts;

TaxistAccount selectedAccount;

public AdminBean() throws SQLException, NamingException {

dao = new DataAccessObject();

listOfData = dao.combineAllDataOfTaxistAccount();

}

public String addNewCabbie() throws SQLException

{

TaxistAccount taxistAccount = new TaxistAccount(username,lastname,firstname,midllename,password,brand,color,number);

if (dao.addNewTaxist(taxistAccount))

{

listOfData = dao.combineAllDataOfTaxistAccount();

password = null;

return "adminMainPage";

}

else return null;

}

public String deleteCabbie() throws SQLException

{

if (dao.deleteTaxist(selectedAccount.getUserId(),selectedAccount.getTaxistId()))

{

listOfData = dao.combineAllDataOfTaxistAccount();

return "cabbieDeleted";

}

else return null;

}

public String goToEditDataOfCarPage()

{

return "editCarData";

}

public String goToEditDataOfCabbie()

{

return "editTaxistData";

}

public String editDataOfCabbie() throws SQLException {

selectedAccount.setPassword(password);

if (dao.editDataOfTaxist(selectedAccount))

{

listOfData = dao.combineAllDataOfTaxistAccount();

return "adminMainPage";

}

else return null;

}

public String editDataOfCar() throws SQLException {

if (dao.editDataOfCar(selectedAccount))

{

listOfData = dao.combineAllDataOfTaxistAccount();

return "adminMainPage";

}

else return null;

}

public TaxistAccount getSelectedAccount() {

return selectedAccount;

}

public void setSelectedAccount(TaxistAccount selectedAccount) {

this.selectedAccount = selectedAccount;

}

public List<TaxistAccount> getListOfData() {

return listOfData;

}

public void setListOfData(List<TaxistAccount> listOfData) {

this.listOfData = listOfData;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getLastname() {

return lastname;

}

public void setLastname(String lastname) {

this.lastname = lastname;

}

public String getFirstname() {

return firstname;

}

public void setFirstname(String firstname) {

this.firstname = firstname;

}

public String getMidllename() {

return midllename;

}

public void setMidllename(String midllename) {

this.midllename = midllename;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getBrand() {

return brand;

}

public void setBrand(String brand) {

this.brand = brand;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

public TaxistAccount[] getSelectedAccounts() {

return selectedAccounts;

}

public void setSelectedAccounts(TaxistAccount[] selectedAccounts) {

this.selectedAccounts = selectedAccounts;

}

public void validateNoExistingOfLogin(ComponentSystemEvent event) throws SQLException {

FacesContext fc = FacesContext.getCurrentInstance();

UIComponent components = event.getComponent();

UIInput uiInputLogin = (UIInput) components.findComponent("login");

String login = uiInputLogin.getLocalValue() == null ? ""

: uiInputLogin.getLocalValue().toString();

String loginId = uiInputLogin.getClientId();

if (dao.findTaxistIdWithUserOnline(login)!=0)

{

FacesMessage msg = new FacesMessage("Пользователь с таким логином уже зарегистрирован");

msg.setSeverity(FacesMessage.SEVERITY\_ERROR);

fc.addMessage(loginId, msg);

fc.renderResponse();

}

}

}

***7.3.2. AuthBackingBean.java***

package beans;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.RequestScoped;

import javax.faces.context.FacesContext;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import java.util.logging.Level;

import java.util.logging.Logger;

@ManagedBean

@RequestScoped

public class AuthBackingBean {

private static Logger log = Logger.getLogger(AuthBackingBean.class.getName());

public String logout() {

String result="/index?faces-redirect=true";

FacesContext context = FacesContext.getCurrentInstance();

HttpServletRequest request = (HttpServletRequest)context.getExternalContext().getRequest();

try {

request.logout();

} catch (ServletException e) {

log.log(Level.SEVERE, "Failed to logout user!", e);

result = "/loginError?faces-redirect=true";

}

return result;

}

}

***7.3.3. CabbieBean.java***

package beans;

import DAO.DataAccessObject;

import model.Order;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.SessionScoped;

import javax.faces.context.FacesContext;

import javax.naming.NamingException;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

@ManagedBean

@SessionScoped

public class CabbieBean

{

private List<Order> listOfOrders;

private List<Order> listOfProgressOrders;

DataAccessObject dao;

private Order[] selectedOrders;

private Order selectedOrder;

private String onlineUser;

private int orderId;

private String phonenumber;

private int taxistId;

private String alreadyHaveOrder = null;

public List<Order> getListOfProgressOrders() {

return listOfProgressOrders;

}

public String getAlreadyHaveOrder() {

return alreadyHaveOrder;

}

public String getPhonenumber() {

return phonenumber;

}

public int getTaxistId() {

return taxistId;

}

public int getOrderId() {

return orderId;

}

public CabbieBean() throws SQLException, NamingException {

dao = new DataAccessObject();

listOfOrders = dao.selectOrdersInStatus(2);

}

public List<Order> getListOfOrders() {

return listOfOrders;

}

public Order[] getSelectedOrders() {

return selectedOrders;

}

public void setSelectedOrders(Order[] selectedOrders) {

this.selectedOrders = selectedOrders;

}

public Order getSelectedOrder() {

return selectedOrder;

}

public void setSelectedOrder(Order selectedOrder) {

this.selectedOrder = selectedOrder;

}

public String getOnlineUser() {

return onlineUser;

}

public String goTakeOrder() throws SQLException

{

FacesContext fc = FacesContext.getCurrentInstance();

onlineUser = fc.getExternalContext().getRemoteUser();//работает

taxistId = dao.findTaxistIdWithUserOnline(onlineUser);//работает

phonenumber = selectedOrder.getPhonenumber();

orderId = dao.findOrderInSearchWithPhoneNumberAndStatus(2,phonenumber);

alreadyHaveOrder = dao.findPhonenumberWithTaxistId(taxistId,1);

return "takingOrder";

}

public String returnOnOrdersPage() throws SQLException

{

listOfOrders = dao.selectOrdersInStatus(2);

return "cabbieMainPage";

}

public String goToPageOfOrderInProgress() throws SQLException

{

listOfProgressOrders = dao.selectOrdersInStatus(1);

for (int i = 0; i < listOfProgressOrders.size(); i++)

{

if (listOfProgressOrders.get(i).getTaxistId()!=taxistId)

{

listOfProgressOrders.remove(i);

}

}

alreadyHaveOrder = dao.findPhonenumberWithTaxistId(taxistId,1);

return "orderInProgressPage";

}

public String finishOrder() throws SQLException {

phonenumber = selectedOrder.getPhonenumber();

orderId = dao.findOrderInSearchWithPhoneNumberAndStatus(1,phonenumber);

if (dao.updateStatusOfOrder(orderId,0,taxistId))

{

listOfOrders = dao.selectOrdersInStatus(2);

alreadyHaveOrder = null;

return "successFinishOrder";

}

else return null;

}

public String takeConcretOrder() throws SQLException

{

if (dao.updateStatusOfOrder(orderId,1,taxistId))

{

listOfOrders = dao.selectOrdersInStatus(2);

//alreadyHaveOrder = null;

return "successTakeOrder";

}

else return null;

}

}

# 7.3.4. OrderBean.java

package beans;

import DAO.DataAccessObject;

import model.Order;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.RequestScoped;

import javax.naming.NamingException;

import javax.validation.constraints.Size;

import java.sql.SQLException;

import java.util.HashMap;

import java.util.Map;

@ManagedBean

@RequestScoped

public class OrderBean

{

public OrderBean() throws NamingException {

streets.put("Василисина","Василисина");

streets.put("Диктора Левитана","Диктора Левитана");

streets.put("Хирурга Орлова","Хирурга Орлова");

streets.put("Завадского","Завадского");

streets.put("Мира","Мира");

dao = new DataAccessObject();

}

DataAccessObject dao;

private Map<String,String> streets = new HashMap<String, String>();

@Size(min = 2,max = 30,message = "Фамилия должна быть от 2 до 30 символов")

private String surname;

@Size(min = 6,max = 11,message = "Номер телефона должен быть от 6 до 11 символов")

private String phonenumber;

@Size(min = 3,max = 40,message = "Название улицы должно быть от 3 до 40 символов")

private String street;

private int house;

private int ponch;

@Size(min = 3,max = 50,message = "Путь должен быть от 3 до 50 символов")

private String way;

private String comment;

public String addOrder() throws SQLException {

Order order = new Order(0,surname,phonenumber,null,2,street,house,ponch,comment,way);

boolean temp = dao.addNewOrder(order);

if (temp)

return "/successAddOrder.xhtml";

return null;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public String getPhonenumber() {

return phonenumber;

}

public void setPhonenumber(String phonenumber) {

this.phonenumber = phonenumber;

}

public String getStreet() {

return street;

}

public void setStreet(String street) {

this.street = street;

}

public int getHouse() {

return house;

}

public void setHouse(int house) {

this.house = house;

}

public int getPonch() {

return ponch;

}

public void setPonch(int ponch) {

this.ponch = ponch;

}

public String getWay() {

return way;

}

public void setWay(String way) {

this.way = way;

}

public String getComment() {

return comment;

}

public void setComment(String comment) {

this.comment = comment;

}

public Map<String, String> getStreets() {

return streets;

}

public void setStreets(Map<String, String> streets) {

this.streets = streets;

}

}8. СТРУКТУРА ВЕБ-ИНТЕРФЕЙСА СИСТЕМЫ.

.

## 8.1. Директория resources.

Папка resources(рис.21) состоит из следующих частей:

* папка images – картинки, которые используются на веб-сайте;
* папка interfaces – user interfaces templates, использующиеся на веб-сайте;

## 8.2. Директория styles.

Папка styles (Рис.22) состоит из следующих частей:

* css/fonts – шрифты, использующиеся на веб-сайте;
* css/default.css – css-стиль, использующийся на веб-сайте;
* css/fonts.css – css-стиль, ответственный за шрифты;
* css/images – картинки, применяющиеся в css-стиле;

## 8.3. Директория templates.

Каждая страница веб-сайта состоит из четырёх частей:

* header – верхняя часть страницы, содержащая в себе основное меню навигации между страницами;
* copyright – нижняя часть страница, содержащая в себе панель copyright;
* content – содержимое страницы;
* logo – логотип;

## 8.4. Основные страницы.

1) index.xhtml – страница, которая содержит в себе приветствие, и ознакомительную информацию о нашей фирме такси, а также информация о предоставляемых услугах.

2) noneRegisterMainPage.xhtml – страница, которая предоставляет пользователю возможность заказать такси,при условии ввода верных данных.

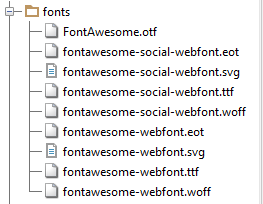
3) cabbieMainPage.xhtml– страница для авторизованных пользователей, являющихся таксистами. На данной странице они могут просмотреть все свободные заказы, а также начать или закончить выполнять определенный заказ.

4) adminMainPage.xhtml – страница, для авторизованных пользователей,являющихся администраторами. На данной странице они могут просмотреть всю информацию,касающуюся таксистов и их автомобилей, а также ей редактировать,создавать новые аккаунты для таксистов, а также удалять таксистов из базы данных.

5) gmap.xhtml – страница, на которой расположена карта.

# 9.РЕАЛИЗАЦИЯ ВЕБ-ИНТЕРФЕЙСА СИСТЕМЫ.

## 9.1. Директория fonts.



## 9.2. Директория swarming

***9.2.1.swarming/default.css***

html, body

{

height: 100%;

}

body

{

margin: 0px;

padding: 0px;

background: #E6E6FA;

font-family: 'Source Sans Pro', sans-serif;

font-size: 12pt;

font-weight: 400;

color: #4E4D4D;

}

h1, h2, h3

{

margin: 0;

padding: 0;

}

p, ol, ul

{

margin-top: 0;

}

ol, ul

{

padding: 0;

list-style: none;

}

p

{

line-height: 180%;

}

strong

{

}

a

{

color: #383838;

}

a:hover

{

text-decoration: none;

}

.container

{

width: 1200px;

margin: 0px auto;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Image Style \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.image

{

display: inline-block;

}

.image img

{

display: block;

width: 100%;

}

.image-full

{

display: block;

width: 100%;

margin: 0 0 0 0;

}

.image-left

{

float: left;

margin: 0 2em 2em 0;

}

.image-centered

{

display: block;

margin: 0 0 2em 0;

}

.image-centered img

{

margin: 0 auto;

width: auto;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* List Styles \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ul.style1

{

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Social Icon Styles \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ul.contact

{

overflow: hidden;

margin: 0em 0 3em 0;

padding-top: 1em;

text-align: center;

}

ul.contact li

{

display: inline-block;

padding: 0em 0.30em;

font-size: 1em;

}

ul.contact li span

{

display: none;

margin: 0;

padding: 0;

}

ul.contact li a

{

color: #FFF;

}

ul.contact li a:before

{

display: inline-block;

width: 40px;

height: 40px;

line-height: 40px;

text-align: center;

font-size: 2em;

color: rgba(255,255,255,.3);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Major Styles \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.major

{

overflow: hidden;

margin-bottom: 4em;

text-align: center;

}

.major h2

{

letter-spacing: 0.10em;

text-transform: uppercase;

font-size: 3em;

font-weight: 700;

color: #3f3f3f;

}

.major .byline

{

display: block;

padding-top: 0.2em;

font-weight: 300;

font-size: 2em;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Title Styles \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.title

{

text-align: center;

}

.title h2

{

letter-spacing: 0.10em;

text-transform: uppercase;

font-weight: 700;

color: #3f3f3f;

}

.title .byline

{

display: block;

padding-top: 1em;

font-weight: 300;

font-size: 1.1em;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Button Style \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.button

{

display: inline-block;

margin: 1em 0em;

padding: 1em 3em 1em 3em;

background: #2A70E8;

border-radius: 8px;

letter-spacing: 0.20em;

text-decoration: none;

text-align: center;

text-transform: uppercase;

font-weight: 400;

font-size: 0.90em;

color: #FFF;

}

.button:before

{

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Header \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#header

{

background: #2A70E8;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Logo \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#logo

{

padding: 4em 0em;

background: #FFF;

text-transform: uppercase;

}

#logo h1

{

text-align: center;

font-weight: 900;

font-size: 2.5em;

}

#logo h1 a

{

}

#logo span

{

padding: 0em 0em 0em 0.5em;

}

#logo h1:before

{

}

#logo a

{

color: #2A70E8;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Menu \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#menu ul

{

margin: 0;

padding: 20px 0px 20px 0px;

list-style: none;

line-height: normal;

text-align: center;

}

#menu li

{

display: inline-block;

}

#menu a

{

display: block;

padding: 1em 2em;

letter-spacing: 1px;

text-decoration: none;

text-transform: uppercase;

font-size: 1em;

font-weight: 700;

color: #FFF;

}

#menu .current\_page\_item a

{

border-radius: 30px;

border: 1px solid;

border-color: rgba(255,255,255,.2);

}

#menu a:hover

{

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Banner \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#banner

{

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Page \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#page-wrapper

{

background: #EEEEEE;

}

#page

{

overflow: hidden;

padding: 6em 0em;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Content \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#content

{

float: left;

width: 1300px;

text-align: center;

}

#content .title

{

padding: 1em 2em 2em 2em;

text-align: center;

}

#content h2

{

font-size: 2em;

}

#content .byline

{

padding-top: 0;

font-size: 1.5em;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Sidebar \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#sidebar

{

float: right;

width: 588px;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Footer \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#footer

{

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#copyright

{

overflow: hidden;

padding: 5em 0em;

}

#copyright p

{

letter-spacing: 0.20em;

text-align: center;

text-transform: uppercase;

font-size: 0.80em;

color: #6F6F6F;

}

#copyright a

{

text-decoration: none;

color: #8C8C8C;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Featured \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#featured-wrapper

{

overflow: hidden;

padding: 5em 0em;

background: #FFF;

}

#featured h2

{

text-align: center;

}

#featured .icon

{

position: relative;

display: block;

background: #2A70E8;

margin: 0px auto 20px auto;

line-height: 2.5em;

font-size: 4em;

text-align: center;

color: #FFF;

}

.column1,

.column2,

.column3,

.column4

{

width: 282px;

}

.column1,

.column2

{

float: left;

margin-right: 24px;

}

.column3

{

float: left;

}

.column4

{

float: right;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Portfolio \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#portfolio-wrapper

{

padding: 8em 0em 6em 0em;

}

#portfolio

{

overflow: hidden;

text-align: center;

}

#portfolio .major h2,

#portfolio .major .byline

{

color: rgba(255,255,255,0.9);

}

#portfolio img

{

margin: 0;

padding: 0;

}

#portfolio .box

{

padding: 30px;

background: #FFF;

border-radius: 0px 0px 8px 8px;

border-right: 1px solid #DADADA;

border-left: 1px solid #DADADA;

}

#contact .major h2,

#contact .major .byline

{

color: rgba(255,255,255,0.9);

}

## 9.3.Директории templates,admin,cabbie и основные страницы.

***9.3.1.addNewCabbie/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 16:48

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:f="http://java.sun.com/jsf/core">

<h:head>

<title>Add New Cabbie Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<br/>

<p:messages autoUpdate="true" closable="true"/>

<h:form>

<f:event type="postValidate" listener="#{adminBean.validateNoExistingOfLogin}"/>

<p:panel header="Таксист" style="width: 400px">

<p:panelGrid columns="2">

<p:outputLabel value="Логин"/>

<p:inputText id="login" value="#{adminBean.username}"/>

<p:outputLabel value="Пароль"/>

<p:password value="#{adminBean.password}"/>

<p:outputLabel value="Фамилия"/>

<p:inputText value="#{adminBean.lastname}"/>

<p:outputLabel value="Имя"/>

<p:inputText value="#{adminBean.firstname}"/>

<p:outputLabel value="Отчество"/>

<p:inputText value="#{adminBean.midllename}"/>

<p:outputLabel value="Марка автомобиля"/>

<p:inputText value="#{adminBean.brand}"/>

<p:outputLabel value="Цвет автомобиля"/>

<p:inputText value="#{adminBean.color}"/>

<p:outputLabel value="Номер автомобиля"/>

<p:inputText value="#{adminBean.number}"/>

</p:panelGrid>

<p:separator/>

<p:commandButton value="Добавить таксиста" action="#{adminBean.addNewCabbie}" icon="ui-icon-circle-plus"/>

</p:panel>

</h:form>

<p:panel style="width: 400px">

<p:commandLink value="Вернуться на страницу администратора" action="adminMainPage"/>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.2. adminMainPage/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 16:01

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:p="http://primefaces.org/ui" xmlns:f="http://java.sun.com/jsf/core">

<h:head>

<title>Admin Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<h:form id="form">

<p:dataTable id="accounts" var="account" value="#{adminBean.listOfData}" rowKey="#{account.username}"

selection="#{adminBean.selectedAccounts}" selectionMode="multiple">

<f:facet name="header">

Нажмите на кнопки справа для удаления, редактирования данных

</f:facet>

<p:column headerText="Логин">

#{account.username}

</p:column>

<p:column headerText="Фамилия">

#{account.lastname}

</p:column>

<p:column headerText="Имя">

#{account.firstname}

</p:column>

<p:column headerText="Отчество">

#{account.middlename}

</p:column>

<p:column headerText="Марка автомобиля">

#{account.brand}

</p:column>

<p:column headerText="Цвет автомобиля">

#{account.color}

</p:column>

<p:column headerText="Номер автомобиля">

#{account.number}

</p:column>

<p:column style="width: 4%">

<p:commandButton id="editDataOfCabbie" action="#{adminBean.goToEditDataOfCabbie}" icon="ui-icon-person"

title="Редактировать данные о таксисте">

<f:setPropertyActionListener value="#{account}" target="#{adminBean.selectedAccount}"/>

</p:commandButton>

</p:column>

<p:column style="width: 4%">

<p:commandButton id="editDataOfCarButton" action="#{adminBean.goToEditDataOfCarPage}" icon="ui-icon-key"

title="Редактировать данные о автомобиле">

<f:setPropertyActionListener value="#{account}" target="#{adminBean.selectedAccount}"/>

</p:commandButton>

</p:column>

<p:column style="width: 4%">

<p:commandButton id="deleteButton" action="#{adminBean.deleteCabbie}" icon="ui-icon-trash"

title="Удалить таксиста">

<f:setPropertyActionListener value="#{account}" target="#{adminBean.selectedAccount}"/>

</p:commandButton>

</p:column>

<f:facet name="footer">

<p:commandButton value="Просмотр" icon="ui-icon-search"

update=":form:display" oncomplete="PF('accountDialog').show()"/>

</f:facet>

</p:dataTable>

<p:dialog header="Информация о таксистах" widgetVar="accountDialog"

width="450" showEffect="explode" hideEffect="explode">

<p>Выбранные таксисты:</p>

<p:dataList id="display"

value="#{adminBean.selectedAccounts}"

var="selectedAcc" type="definition" emptyMessage="Таксисты не выбраны">

<p:panelGrid columns="2">

<p:outputLabel value="Логин"/>

<p:outputLabel value="#{selectedAcc.username}"/>

<p:outputLabel value="Фамилия"/>

<p:outputLabel value="#{selectedAcc.lastname}"/>

<p:outputLabel value="Имя"/>

<p:outputLabel value="#{selectedAcc.firstname}"/>

<p:outputLabel value="Отчество"/>

<p:outputLabel value="#{selectedAcc.middlename}"/>

<p:outputLabel value="Марка автомобиля"/>

<p:outputLabel value="#{selectedAcc.brand}"/>

<p:outputLabel value="Цвет автомобиля"/>

<p:outputLabel value="#{selectedAcc.color}"/>

<p:outputLabel value="Номер автомобиля"/>

<p:outputLabel value="#{selectedAcc.number}"/>

<p:outputLabel value="User ID"/>

<p:outputLabel value="#{selectedAcc.userId}"/>

<p:outputLabel value="Taxist ID"/>

<p:outputLabel value="#{selectedAcc.taxistId}"/>

</p:panelGrid>

</p:dataList>

</p:dialog>

</h:form>

<p:panel style="width: 400px">

<p:commandLink value="Добавить таксиста" action="addNewCabbie"/>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.3. adminMainPage/header.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 16:01

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Admin Header</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="header">

<div id="menu" class="container">

<ul>

<li><a jsfc="h:outputLink" value="/TaxiAgency/index.xhtml" accesskey="1" title="">Информация</a></li>

<li><a jsfc="h:outputLink" value="/TaxiAgency/noneRegisterMainPage.xhtml" accesskey="1" title="">Для клиентов</a></li>

<li><a jsfc="h:outputLink" value="/TaxiAgency/cabbie/cabbieMainPage.xhtml" accesskey="2" title="">Для таксистов</a></li>

<li class="current\_page\_item"><a jsfc="h:outputLink" value="adminMainPage.xhtml" accesskey="3" title="">Блок администратора</a></li>

</ul>

</div>

</div>

</ui:composition>

</h:body>

</html>

***9.3.4. cabbieDeleted/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 20:29

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Cabbie Deleted Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<p:panel style="width: 400px">

<p:outputLabel value="Таксист уволен"/>

<br/>

<p:commandLink value="Вернуться на страницу просмотра таксистов" action="adminMainPage"/>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.5. CabbieMainPage/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 16:01

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:p="http://primefaces.org/ui"

xmlns:f="http://java.sun.com/jsf/core">

<h:head>

<title>Cabbie Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<h:form id="form">

<p:dataTable id="orders" var="order" value="#{cabbieBean.listOfOrders}" rowKey="#{order.surname}"

selection="#{cabbieBean.selectedOrders}" selectionMode="multiple">

<f:facet name="header">

Нажмите кнопку справа от заказа, для того чтобы взять заказ.

</f:facet>

<p:column headerText="Фамилия">

#{order.surname}

</p:column>

<p:column headerText="Номер телефона">

#{order.phonenumber}

</p:column>

<p:column headerText="Время заказа">

#{order.dateOfOrdering}

</p:column>

<p:column sortBy="street" headerText="Улица">

#{order.street}

</p:column>

<p:column headerText="Дом">

#{order.house}

</p:column>

<p:column headerText="Подъезд">

#{order.ponch}

</p:column>

<p:column headerText="Комментарий">

#{order.comment}

</p:column>

<p:column headerText="Путь">

#{order.way}

</p:column>

<p:column style="width: 4%">

<p:commandButton id="takeButton" update=":form:display" action="#{cabbieBean.goTakeOrder}" icon="ui-icon-key"

title="Взять заказ">

<f:setPropertyActionListener value="#{order}" target="#{cabbieBean.selectedOrder}"/>

</p:commandButton>

</p:column>

<f:facet name="footer">

<p:commandButton value="Просмотр" icon="ui-icon-search"

update=":form:display" oncomplete="PF('orderDialog').show()"/>

<br/>

<p:commandLink value="Перейти на страницу выполнения заказа" action="#{cabbieBean.goToPageOfOrderInProgress}"/>

</f:facet>

</p:dataTable>

<p:dialog header="Подробности заказа" widgetVar="orderDialog"

width="450" showEffect="explode" hideEffect="explode">

<p>Просмотр заказов:</p>

<p:dataList id="display" value="#{cabbieBean.selectedOrders}"

var="selectedOr" type="definition" emptyMessage="Заказы не выбраны">

<p:panelGrid columns="2">

<p:outputLabel value="Фамилия"/>

<p:outputLabel value="#{selectedOr.surname}"/>

<p:outputLabel value="Номер телефона"/>

<p:outputLabel value="#{selectedOr.phonenumber}"/>

<p:outputLabel value="Время заказа"/>

<p:outputLabel value="#{selectedOr.dateOfOrdering}"/>

<p:outputLabel value="Улица"/>

<p:outputLabel value="#{selectedOr.street}"/>

<p:outputLabel value="Дом"/>

<p:outputLabel value="#{selectedOr.house}"/>

<p:outputLabel value="Подъезд"/>

<p:outputLabel value="#{selectedOr.ponch}"/>

<p:outputLabel value="Комментарий"/>

<p:outputLabel value="#{selectedOr.comment}"/>

<p:outputLabel value="Путь"/>

<p:outputLabel value="#{selectedOr.way}"/>

</p:panelGrid>

<br/>

</p:dataList>

</p:dialog>

</h:form>

</ui:composition>

</h:body>

</html>

***9.3.6. CabbieMainPage/header.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 16:01

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Cabbie Header</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="header">

<div id="menu" class="container">

<ul>

<li><a jsfc="h:outputLink" value="/TaxiAgency/index.xhtml" accesskey="1" title="">Информация</a></li>

<li><a jsfc="h:outputLink" value="/TaxiAgency/noneRegisterMainPage.xhtml" accesskey="1" title="">Для клиентов</a></li>

<li class="current\_page\_item"><a jsfc="h:outputLink" value="cabbieMainPage.xhtml" accesskey="2" title="">Для таксистов</a></li>

<li><a jsfc="h:outputLink" value="/TaxiAgency/admin/adminMainPage.xhtml" accesskey="3" title="">Блок администратора</a></li>

</ul>

</div>

</div>

</ui:composition>

</h:body>

</html>

***9.3.7. common/copyrightxhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 15:45

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Copyright</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="copyright" class="container">

<p>Copyright (c) 2013 Vladimir State University</p>

</div>

</ui:composition>

</h:body>

</html>

***9.3.8. common/header.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 17:44

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Login Header</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="header">

<div id="menu" class="container">

<ul>

<li><a jsfc="h:outputLink" value="index.xhtml" accesskey="1" title="">Информация</a></li>

<li><a jsfc="h:outputLink" value="noneRegisterMainPage.xhtml" accesskey="1" title="">Для клиентов</a></li>

<li><a jsfc="h:outputLink" value="cabbie/cabbieMainPage.xhtml" accesskey="2" title="">Для таксистов</a></li>

<li><a jsfc="h:outputLink" value="admin/adminMainPage.xhtml" accesskey="3" title="">Блок администратора</a></li>

</ul>

</div>

</div>

</ui:composition>

</h:body>

</html>

***9.3.9. common/logo.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 15:47

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:p="http://primefaces.org/ui">

<h:head>

<title>Logo</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="logo">

<h1><a href="#" class="icon icon-group"><span>Такси Агенство</span></a></h1>

<ui:fragment rendered="#{request.getRemoteUser()!=null}">

<p:outputLabel value="Добро пожаловать,#{request.getRemoteUser()}"/>

<h:form>

<p:commandButton action="#{authBackingBean.logout}" value="Logout"/>

</h:form>

</ui:fragment>

</div>

</ui:composition>

</h:body>

</html>

***9.3.10. editCarData/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 21:16

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Edit Car Data Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<br/>

<p:messages autoUpdate="true" closable="true"/>

<h:form>

<p:panel header="Автомобиль" style="width: 400px">

<p:panelGrid columns="2">

<p:outputLabel value="Марка автомобиля"/>

<p:inputText value="#{adminBean.selectedAccount.brand}"/>

<p:outputLabel value="Цвет автомобиля"/>

<p:inputText value="#{adminBean.selectedAccount.color}"/>

<p:outputLabel value="Номер автомобиля"/>

<p:inputText value="#{adminBean.selectedAccount.number}"/>

</p:panelGrid>

<p:separator/>

<p:commandButton value="Редактировать данные" action="#{adminBean.editDataOfCar}"

icon="ui-icon-arrowrefresh-1-3"/>

</p:panel>

</h:form>

<p:panel style="width: 400px">

<p:commandLink value="Вернуться на страницу администратора" action="adminMainPage"/>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.11. editTaxistData/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 21:16

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Content Edit Taxist Data</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<br/>

<p:messages autoUpdate="true" closable="true"/>

<h:form>

<p:panel header="Таксист" style="width: 400px">

<p:panelGrid columns="2">

<p:outputLabel value="Логин"/>

<p:inputText value="#{adminBean.selectedAccount.username}"/>

<p:outputLabel value="Фамилия"/>

<p:inputText value="#{adminBean.selectedAccount.lastname}"/>

<p:outputLabel value="Имя"/>

<p:inputText value="#{adminBean.selectedAccount.firstname}"/>

<p:outputLabel value="Отчество"/>

<p:inputText value="#{adminBean.selectedAccount.middlename}"/>

<p:outputLabel value="Пароль"/>

<p:password value="#{adminBean.password}"/>

</p:panelGrid>

<p:separator/>

<p:commandButton value="Редактировать данные" action="#{adminBean.editDataOfCabbie}"

icon="ui-icon-arrowrefresh-1-3"/>

</p:panel>

</h:form>

<p:panel style="width: 400px">

<p:commandLink value="Вернуться на страницу администратора" action="adminMainPage"/>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.12. index/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 15:49

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Index Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="content">

<center> <h1>Вас приветствует Владимирское такси</h1>

<h1>Чтобы сделать онлайн заказ войдите в раздел "Для клиентов"</h1>

<h1>Вы можете связаться с нами напрямую по телефону: 89049589497</h1>

<h1>Спасибо,что выбрали наше такси</h1> </center>

</div>

</ui:composition>

</h:body>

</html>

***9.3.13.index/header.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 15:48

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Index Header</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="header">

<div id="menu" class="container">

<ul>

<li class="current\_page\_item"><a jsfc="h:outputLink" value="index.xhtml" accesskey="1" title="">Информация</a></li>

<li><a jsfc="h:outputLink" value="noneRegisterMainPage.xhtml" accesskey="1" title="">Для клиентов</a></li>

<li><a jsfc="h:outputLink" value="cabbie/cabbieMainPage.xhtml" accesskey="2" title="">Для таксистов</a></li>

<li><a jsfc="h:outputLink" value="admin/adminMainPage.xhtml" accesskey="3" title="">Блок администратора</a></li>

</ul>

</div>

</div>

</ui:composition>

</h:body>

</html>

***9.3.14. noneRegisteredMainPage/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 15:58

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:p="http://primefaces.org/ui" xmlns:f="http://java.sun.com/jsf/core">

<h:head>

<title>None Register Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

<script src="http://maps.google.com/maps/api/js?sensor=false"

type="text/javascript"></script>

</h:head>

<h:body>

<ui:composition>

<br/>

<p:messages autoUpdate="true" closable="true"/>

<h:form>

<p:panel header="Клиент" style="width: 400px" >

<p:panelGrid columns="2">

<p:outputLabel value="Фамилия"/>

<p:inputText value="#{orderBean.surname}"/>

<p:outputLabel value="Телефон"/>

<p:inputText value="#{orderBean.phonenumber}"/>

</p:panelGrid>

</p:panel>

<p:panel header="Место" style="width: 400px">

<p:panelGrid columns="2">

<p:outputLabel value="Улица"/>

<p:selectOneMenu id="streets" value="#{orderBean.street}">

<f:selectItem itemLabel="" itemValue="" />

<f:selectItems value="#{orderBean.streets}" />

</p:selectOneMenu>

<p:outputLabel value="Дом"/>

<p:inputText value="#{orderBean.house}" validatorMessage="Номер дома должен быть от 1 до 300">

<f:validateLongRange minimum="1" maximum="300"/>

</p:inputText>

<p:outputLabel value="Подъезд"/>

<p:inputText value="#{orderBean.ponch}" validatorMessage="Номер подъезда должен быть от 1 до 12">

<f:validateLongRange minimum="1" maximum="12"/>

</p:inputText>

<p:outputLabel value="Куда Едем"/>

<p:inputText value="#{orderBean.way}"/>

<p:outputLabel value="Комментарий"/>

<p:inputTextarea value="#{orderBean.comment}"/>

</p:panelGrid>

</p:panel>

<p:separator />

<p:commandButton value="Заказать такси" action="#{orderBean.addOrder}" icon="ui-icon-check"/>

</h:form>

<br/>

<p:panel style="width: 400px">

<a jsfc="h:outputLink" value="/TaxiAgency/gmap.xhtml" accesskey="1" title="">Перейти к просмотру карты</a>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.15. noneRegisteredMainPage/header.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 15:58

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>None Register Header</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<div id="header">

<div id="menu" class="container">

<ul>

<li><a jsfc="h:outputLink" value="index.xhtml" accesskey="1" title="">Информация</a></li>

<li class="current\_page\_item"><a jsfc="h:outputLink" value="noneRegisterMainPage.xhtml" accesskey="1" title="">Для клиентов</a></li>

<li><a jsfc="h:outputLink" value="cabbie/cabbieMainPage.xhtml" accesskey="2" title="">Для таксистов</a></li>

<li><a jsfc="h:outputLink" value="admin/adminMainPage.xhtml" accesskey="3" title="">Блок администратора</a></li>

</ul>

</div>

</div>

</ui:composition>

</h:body>

</html>

***9.3.16. orderInProgressPage/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 1:39

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:p="http://primefaces.org/ui"

xmlns:f="http://java.sun.com/jsf/core">

<h:head>

<title>Order In Progress Page Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<h:form id="form">

<p:dataTable id="progorders" var="progorder" value="#{cabbieBean.listOfProgressOrders}"

rowKey="#{progorder.surname}" emptyMessage="У вас нету заказа в выполнении. Возьмите заказ.">

<f:facet name="header">

Нажмите кнопку справа, для того чтобы пометить заказ выполненным.

</f:facet>

<p:column headerText="Фамилия">

#{progorder.surname}

</p:column>

<p:column headerText="Номер телефона">

#{progorder.phonenumber}

</p:column>

<p:column headerText="Время заказа">

#{progorder.dateOfOrdering}

</p:column>

<p:column headerText="Улица">

#{progorder.street}

</p:column>

<p:column headerText="Дом">

#{progorder.house}

</p:column>

<p:column headerText="Подъезд">

#{progorder.ponch}

</p:column>

<p:column headerText="Комментарий">

#{progorder.comment}

</p:column>

<p:column headerText="Путь">

#{progorder.way}

</p:column>

<p:column style="width: 4%">

<p:commandButton action="#{cabbieBean.finishOrder}" title="Завершить заказ" icon="ui-icon-closethick">

<f:setPropertyActionListener value="#{progorder}" target="#{cabbieBean.selectedOrder}"/>

</p:commandButton>

</p:column>

</p:dataTable>

<p:panel style="width: 400px">

<p:commandLink value="Вернуться на страницу просмотра заказов" action="cabbieMainPage"/>

</p:panel>

</h:form>

</ui:composition>

</h:body>

</html>

***9.3.17. SuccessAddOrder/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 23:00

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Content Success Add Order</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<p:panel style="width: 800px">

<h1>Ваша заявка принята.Ждите звонка.</h1>

<a jsfc="h:outputLink" value="/TaxiAgency/index.xhtml" accesskey="1" title="">Вернуться на главную страницу</a>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.18.SuccessFinishOrder/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 2:23

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:p="http://primefaces.org/ui">

<h:head>

<title>Success Finish Order Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<p:panel style="width: 400px">

<p:outputLabel value="Заказ успешно завершен"/>

<br/>

<p:commandLink value="Вернуться на страницу просмотра заказов" action="cabbieMainPage"/>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.19 successTakeOrder/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 21.12.13

Time: 14:47

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Content Success Take Order</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<p:panel header="Заказ взят" style="width: 400px" >

<p:outputLabel value="Заказ по адресу: ул. #{cabbieBean.selectedOrder.street}, д. #{cabbieBean.selectedOrder.house},

подъезд #{cabbieBean.selectedOrder.ponch} успешно взят"/>

<br/>

<p:commandLink value="Вернуться на страницу просмотра заказов" action="#{cabbieBean.returnOnOrdersPage}"/>

</p:panel>

</ui:composition>

</h:body>

</html>

***9.3.20. takingOrder/content.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 21.12.13

Time: 14:23

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Take Order</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:composition>

<ui:fragment rendered="#{cabbieBean.alreadyHaveOrder==null}">

<p:panel header="Информация о заказе" style="width: 400px">

<p:panelGrid columns="2">

<h:outputLabel value="Фамилия"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.surname}"/>

<h:outputLabel value="Номер телефона"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.phonenumber}"/>

<h:outputLabel value="Время заказа"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.dateOfOrdering}"/>

<h:outputLabel value="Улица"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.street}"/>

<h:outputLabel value="Дом"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.house}"/>

<h:outputLabel value="Подъезд"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.ponch}"/>

<h:outputLabel value="Комментарий"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.comment}"/>

<h:outputLabel value="Путь"/>

<h:outputLabel value="#{cabbieBean.selectedOrder.way}"/>

</p:panelGrid>

<p:commandButton title="Взять заказ" icon="ui-icon-circle-check" value="Взять заказ" action="#{cabbieBean.takeConcretOrder}"/>

<br/>

<p:commandLink value="Вернуться на страницу просмотра заказов" action="cabbieMainPage"/>

</p:panel>

</ui:fragment>

<ui:fragment rendered="#{cabbieBean.alreadyHaveOrder!=null}">

<p:panel header="Уже есть заказ" style="width: 400px">

<p:outputLabel value="У вас же есть заказ, для того чтобы взять новый, завершите предыдущий."/>

<br/>

<p:commandLink value="Вернуться на страницу просмотра заказов" action="cabbieMainPage"/>

<br/>

<p:commandLink value="Перейти на страницу выполнения заказа" action="#{cabbieBean.goToPageOfOrderInProgress}"/>

</p:panel>

</ui:fragment>

</ui:composition>

</h:body>

</html>

***9.3.21. gmap.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 22:07

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:f="http://java.sun.com/jsf/core">

<h:head>

<title>Simple JSF Facelets page</title>

<script src="http://maps.google.com/maps/api/js?sensor=false"

type="text/javascript"></script>

</h:head>

<h:body>

<f:view contentType="text/html">

<p:outputLabel value="Карта Владимира"/>

<p:gmap center="56.128739,40.405863" zoom="15" type="HYBRID" style="width:1200px;height:800px" />

</f:view>

<p:panel style="width: 400px">

<a jsfc="h:outputLink" value="/TaxiAgency/noneRegisterMainPage.xhtml" accesskey="1" title="">Вернуться на страницу клиента</a>

</p:panel>

</h:body>

</html>

***9.3.22. index.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Страница входа</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/index/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/index/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.23. login.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 17:02

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui">

<h:head>

<title>Страница авторизации</title>

</h:head>

<h:body>

<p:panel header="Авторизация" style="width: 400px">

<form method="post" action="j\_security\_check">

<p:panelGrid columns="2">

<h:outputLabel for="j\_username" value="Username" />

<input type="text" name="j\_username" />

<h:outputLabel for="j\_password" value="Password" />

<input type="password" name="j\_password" />

<h:outputText value="" />

<p:panelGrid columns="2">

<input type="submit" name="submit" value="Login" />

<h:button outcome="index" value="Cancel" />

</p:panelGrid>

</p:panelGrid>

</form>

<br/>

<a jsfc="h:outputLink" value="/TaxiAgency/index.xhtml" accesskey="1" title="">Вернуться на главную страницу</a>

</p:panel>

</h:body>

</html>

***9.3.24 noneRegisteredMainPage.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 16:06

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Раздел клиентов</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/noneRegisterMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/noneRegisterMainPage/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.25. succesAddOrder.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 23:00

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Заявка принята</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/common/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/successAddOrder/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.26. admin/addNewCabbie.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 16:46

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Добавить нового таксиста</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/adminMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/addNewCabbie/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.27. admin/adminMainPage.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 16:05

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Блок администратора</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/adminMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/adminMainPage/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.28. admin/cabbieDeleted.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 20:27

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Таксист удален</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/adminMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/cabbieDeleted/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.29. admin/editCarDate.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 21:14

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Редактировать данные об автомобиле</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/adminMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/editCarData/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.30. admin/editTaxistData.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 21:15

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Редактировать данные о таксисте</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/adminMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/editTaxistData/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.31. cabbie/cabbieMainPage.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 20.12.13

Time: 16:05

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets" >

<h:head>

<title>Раздел таксиста</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/cabbieMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/cabbieMainPage/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.32. cabbie/orderInProgressPage.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 1:38

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Заказы,находящиеся в выполнении</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/cabbieMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/orderInProgressPage/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.33.cabbie/successFinishOrder.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 22.12.13

Time: 2:22

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Успешное выполнение заказа</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/cabbieMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/successFinishOrder/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.34.cabbie/successTakeOrder.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 21.12.13

Time: 14:47

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Заказ взят</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/cabbieMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/successTakeOrder/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

***9.3.34.cabbie/takingOrder.xhtml***

<?xml version="1.0" encoding="UTF-8"?>

<!--

Created by IntelliJ IDEA.

User: Позитроника

Date: 21.12.13

Time: 13:57

-->

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Взять заказ</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />

<link href="../swarming/default.css" rel="stylesheet" type="text/css" media="all" />

<link href="../swarming/fonts.css" rel="stylesheet" type="text/css" media="all" />

</h:head>

<h:body>

<ui:insert name="logotype" >

<ui:include src="/templates/common/logo.xhtml" />

</ui:insert>

<ui:insert name="header">

<ui:include src="/templates/cabbieMainPage/header.xhtml" />

</ui:insert>

<ui:insert name="content" >

<ui:include src="/templates/takingOrder/content.xhtml" />

</ui:insert>

<ui:insert name="copyright" >

<ui:include src="/templates/common/copyright.xhtml"/>

</ui:insert>

</h:body>

</html>

## 9.4. Директория WEB-INF.

***9.4.1. faces-config.xml***

<?xml version='1.0' encoding='UTF-8'?>

<faces-config xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-facesconfig\_2\_0.xsd"

version="2.0">

</faces-config>

***9.4.2. sun-web.xml***

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<!DOCTYPE sun-web-app PUBLIC "-//Sun Microsystems, Inc.//DTD

Application Server 9.0 Servlet 2.5//EN"

"http://www.sun.com/software/appserver/dtds/sun-web-app\_2\_5-0.dtd">

<sun-web-app error-url="">

<security-role-mapping>

<role-name>ADMIN</role-name>

<group-name>ADMIN</group-name>

</security-role-mapping>

<security-role-mapping>

<role-name>CABBIE</role-name>

<group-name>CABBIE</group-name>

</security-role-mapping>

<class-loader delegate="true"/>

<parameter-encoding default-charset="UTF-8"/>

</sun-web-app>

***9.4.3. web.xml***

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"

version="2.5">

<servlet>

<servlet-name>Faces Servlet</servlet-name>

<servlet-class>javax.faces.webapp.FacesServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>Faces Servlet</servlet-name>

<url-pattern>/faces/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Faces Servlet</servlet-name>

<url-pattern>\*.faces</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Faces Servlet</servlet-name>

<url-pattern>\*.xhtml</url-pattern>

</servlet-mapping>

<login-config>

<auth-method>FORM</auth-method>

<realm-name>jdbc-realm</realm-name>

<form-login-config>

<form-login-page>/faces/login.xhtml</form-login-page>

<form-error-page>/faces/loginError.xhtml</form-error-page>

</form-login-config>

</login-config>

<security-constraint>

<web-resource-collection>

<web-resource-name>Admin User</web-resource-name>

<url-pattern>/faces/admin/\*</url-pattern>

<url-pattern>/admin/\*</url-pattern>

<http-method>GET</http-method>

<http-method>POST</http-method>

</web-resource-collection>

<auth-constraint>

<role-name>ADMIN</role-name>

</auth-constraint>

</security-constraint>

<security-constraint>

<web-resource-collection>

<web-resource-name>Cabbie User</web-resource-name>

<url-pattern>/faces/cabbie/\*</url-pattern>

<url-pattern>/cabbie/\*</url-pattern>

<http-method>GET</http-method>

<http-method>POST</http-method>

</web-resource-collection>

<auth-constraint>

<role-name>CABBIE</role-name>

</auth-constraint>

</security-constraint>

<resource-ref>

<description>DB Connection</description>

<res-ref-name>jdbc/showcase</res-ref-name>

<res-type>javax.sql.DataSource</res-type>

<res-auth>Container</res-auth>

<res-sharing-scope>Shareable</res-sharing-scope>

<mapped-name>jdbc/showcase</mapped-name>

</resource-ref>

</web-app>

# 10. ДЕМОНСТРАЦИЯ ПРОТОТИПА ПРОГРАММНОЙ СИСТЕМЫ.

Заходим на страницу index.xhtml (Рис.13), которая содержит в себе приветствие, и ознакомительную информацию о кинотеатре, такую как адрес, контактные данные, а также информация о услугах, предоставляемых кинотеатром.

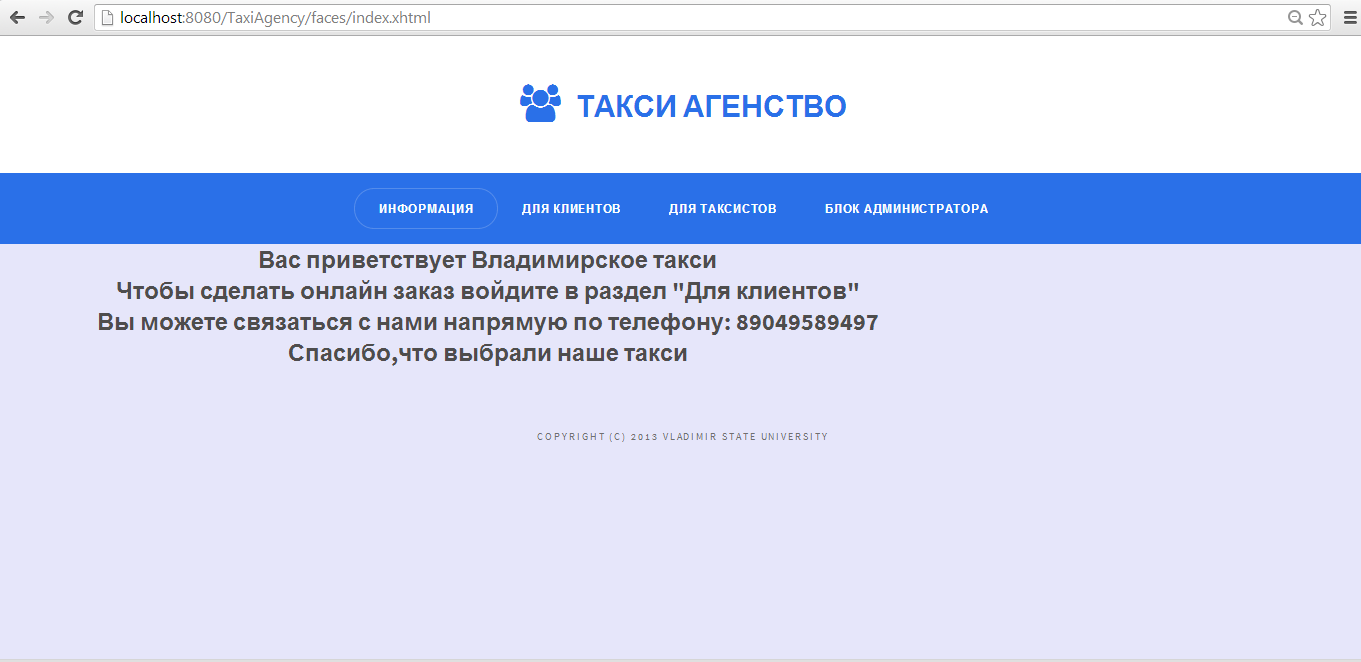


Рис.13. index.xhtml.

Перейдем на страницу noneRegisterMainPage.xhtml (Рис.14), которая позволяет произвести заказ такси онлайн, заполнив все необходимые формы.

Если все заполнено правильно, то мы видим страницу успешного завершения(рис.15),либо система подсказывает нам,какие поля заполнены неправильно(рис.16).

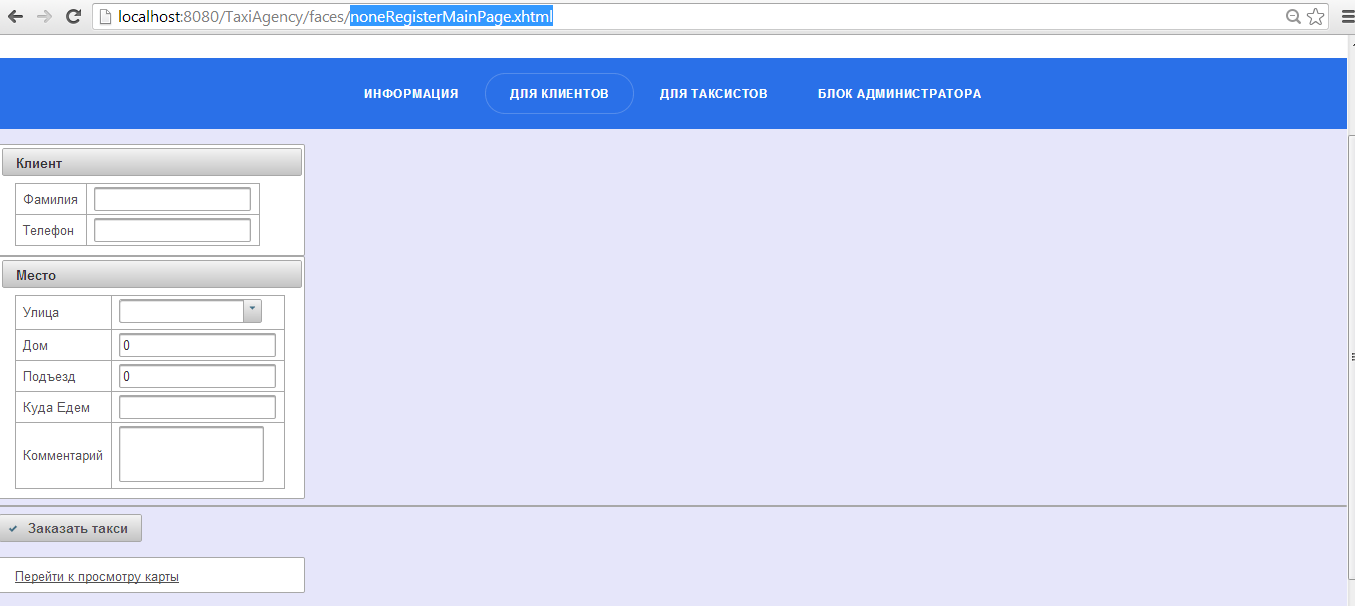


Рис.14. noneRegisterMainPage.xhtml

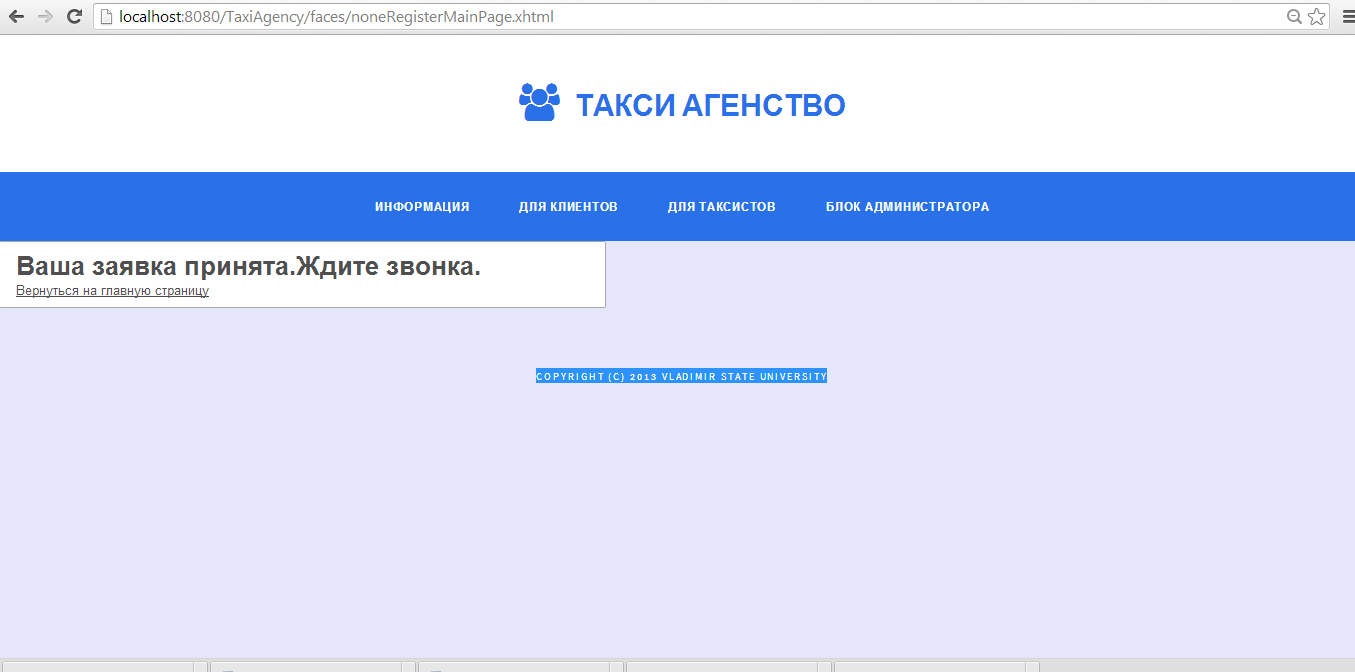


Рис.15. noneRegisterMainPage.xhtml

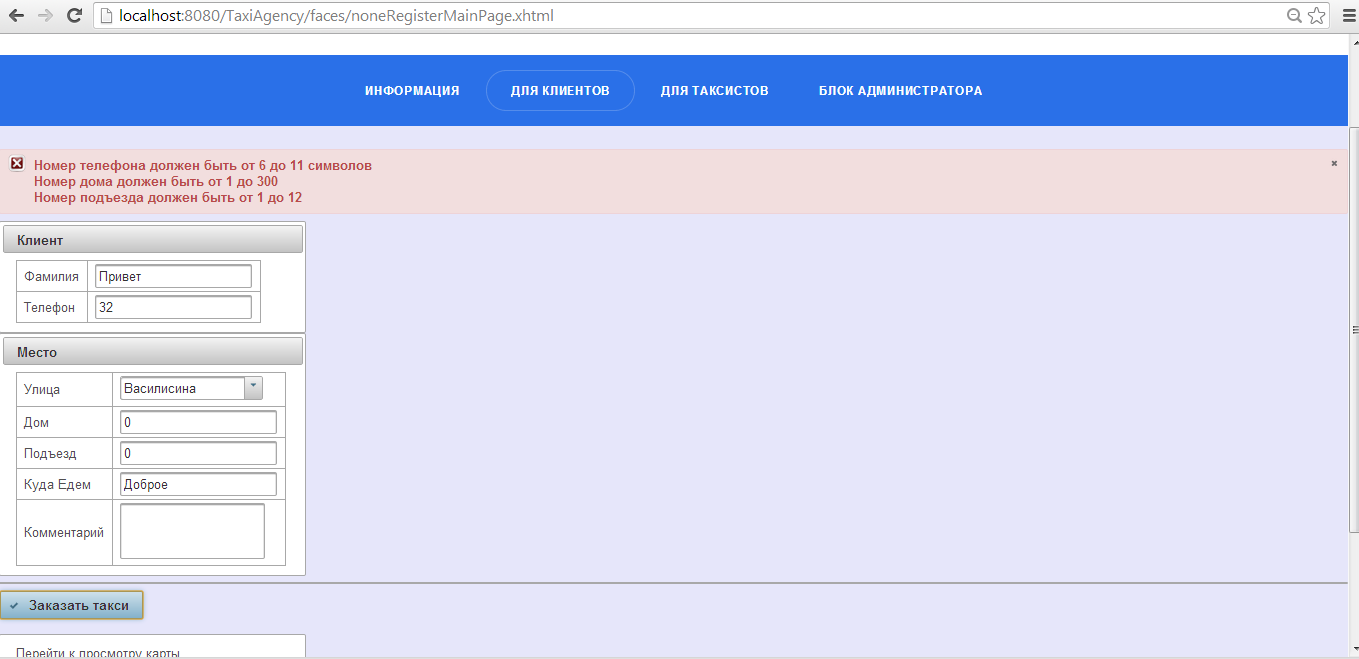


Рис.16 noneRegisterMainPage.xhtml

Если пользователь не знает своего точного местоположения, он может воспользоваться картой, реализованной на нашем сайте, нажав кнопку «перейти к просмотру карты».(рис.17)

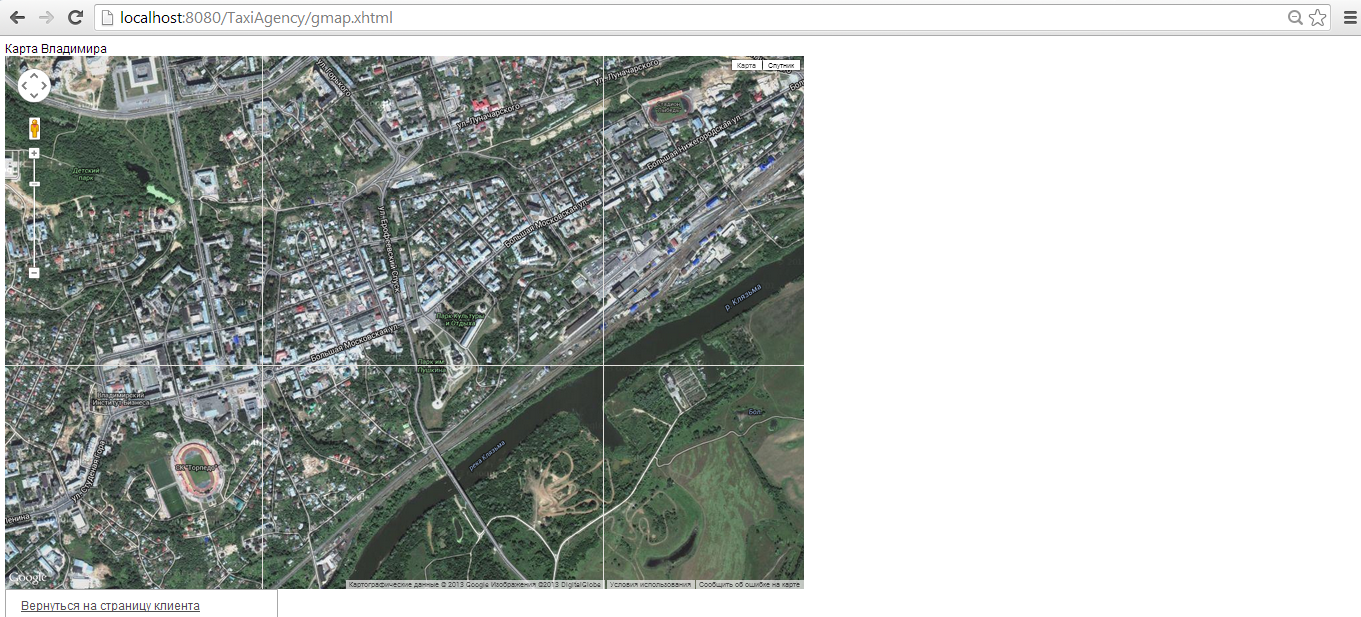


Рис. 17

По ссылке «Для таксистов» перейдем на страницу авторизации(cabbieMainPage.xhtml), и введем данные для авторизации (рис.18).

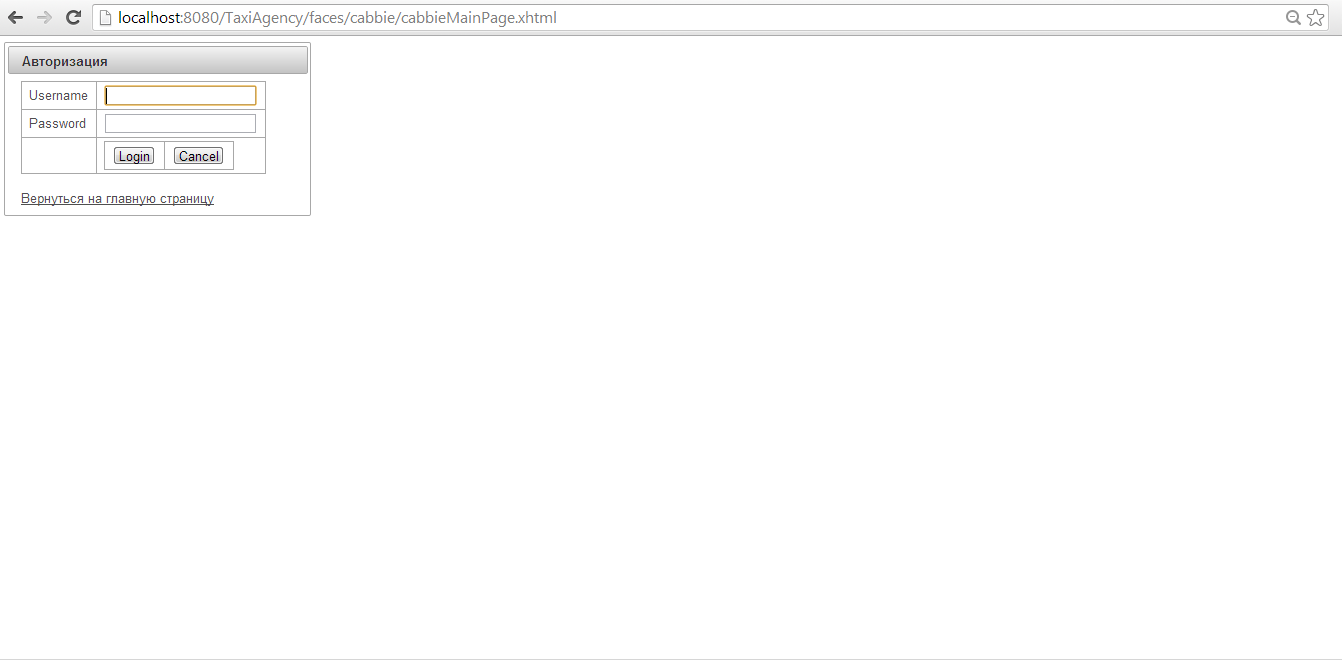


Рис.18. cabbieMainPage.xhtml

После нажатия успешного ввода данных и кнопки «Login» произойдет переход на страницу cabbieMainPage.xhtml

После авторизации страница для работы таксиста с заказами выглядит таким образом (Рис.19)

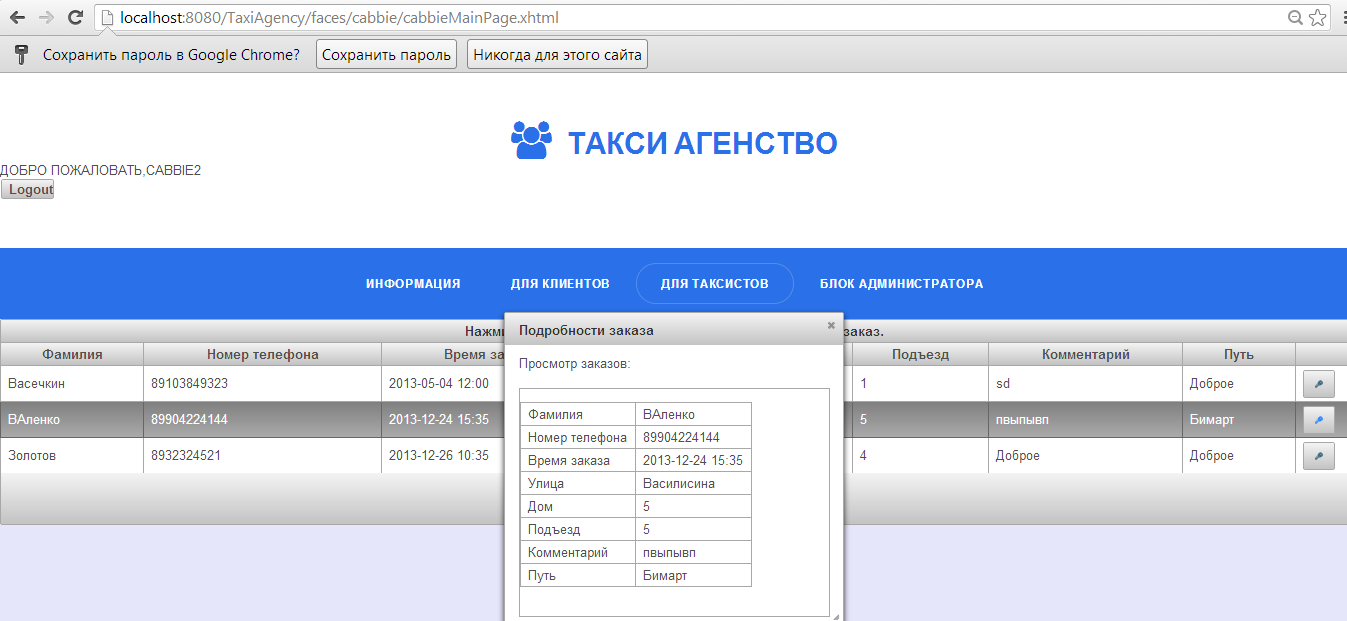


Рис.19

Далее при нажатии кнопки «взять заказ» мы переходим на страницу просмотра информации о заказе. (Рисунок 20)

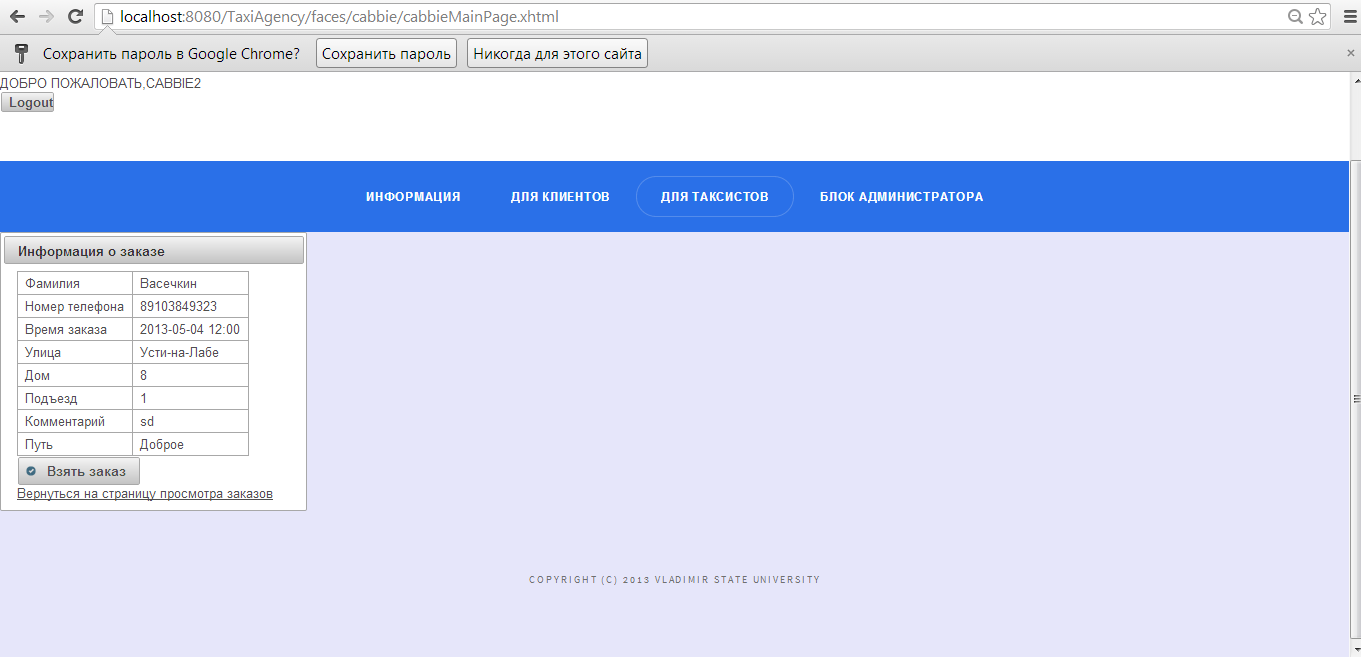


Рисунок 20

При подтверждении заказа, мы можем вернуться на страницу просмотра заказа,однако взять следующий заказ, не закончив предыдущий не можем.(Рисунок 21)

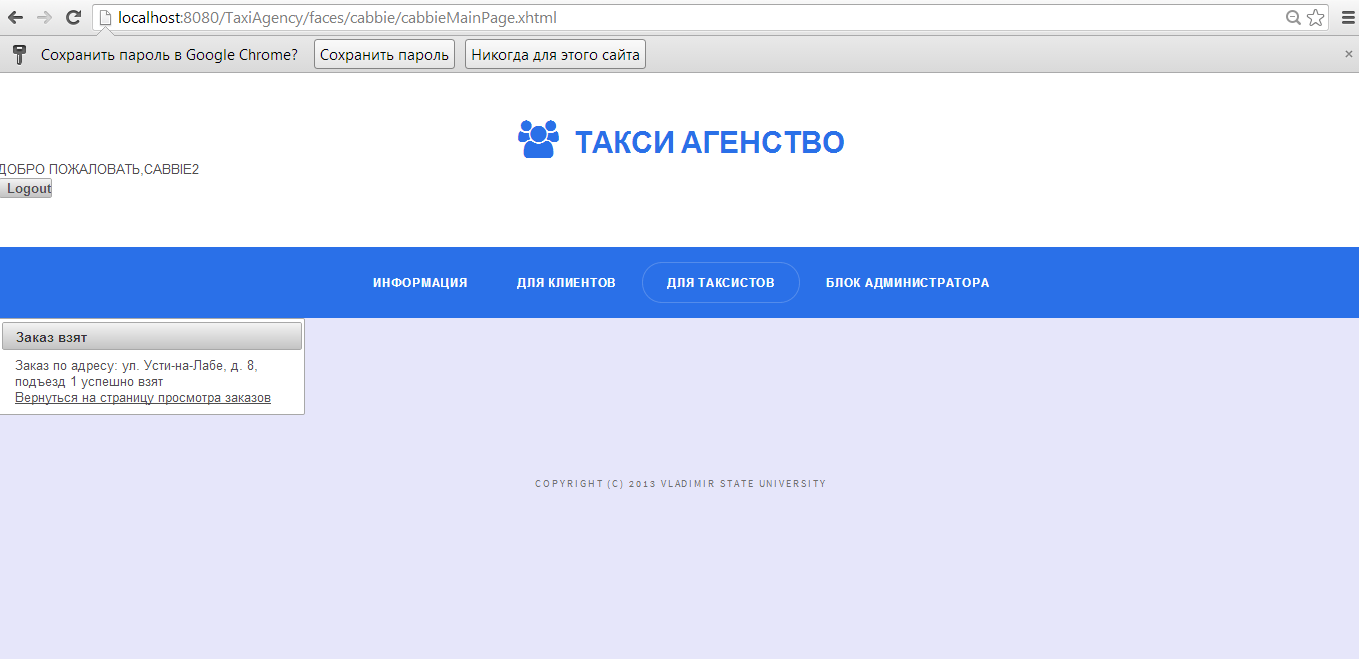


Рисунок 21

Поэтому мы просматриваем свой текущий заказ.(рисунок 22)

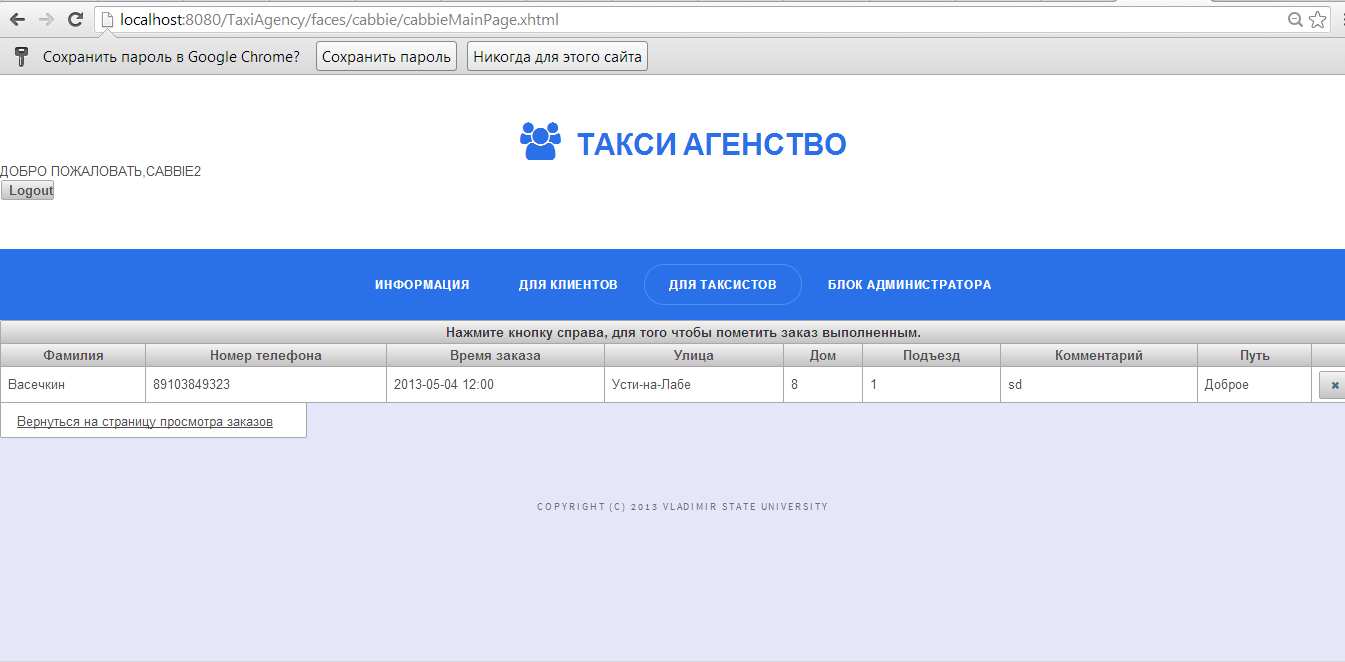


Рисунок 22

И завершаем его, нажимая на кнопку, завершить заказ, и перед нами появляется уведомление, что заказ завершен. (Рисунок 23)

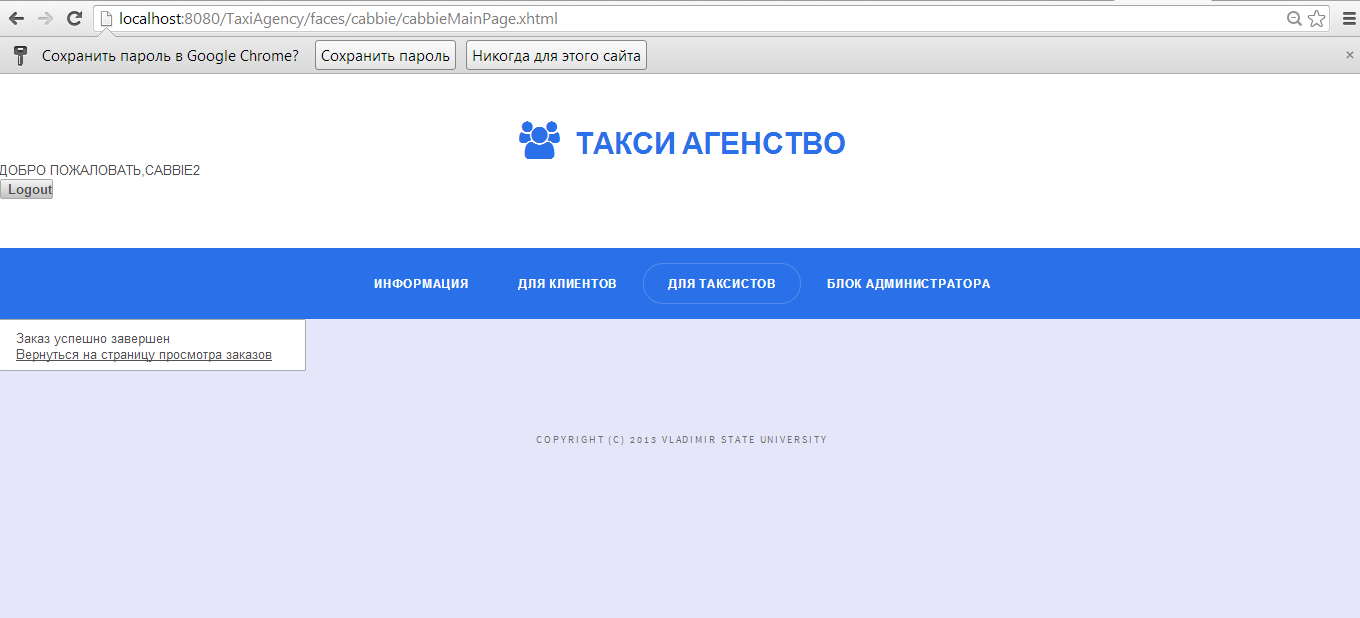


Рисунок 23

Если же мы зайдем в систему,используя роль Администратор, то увидим следующую страницу, в которой собраны данные о таксистах и их машинах.

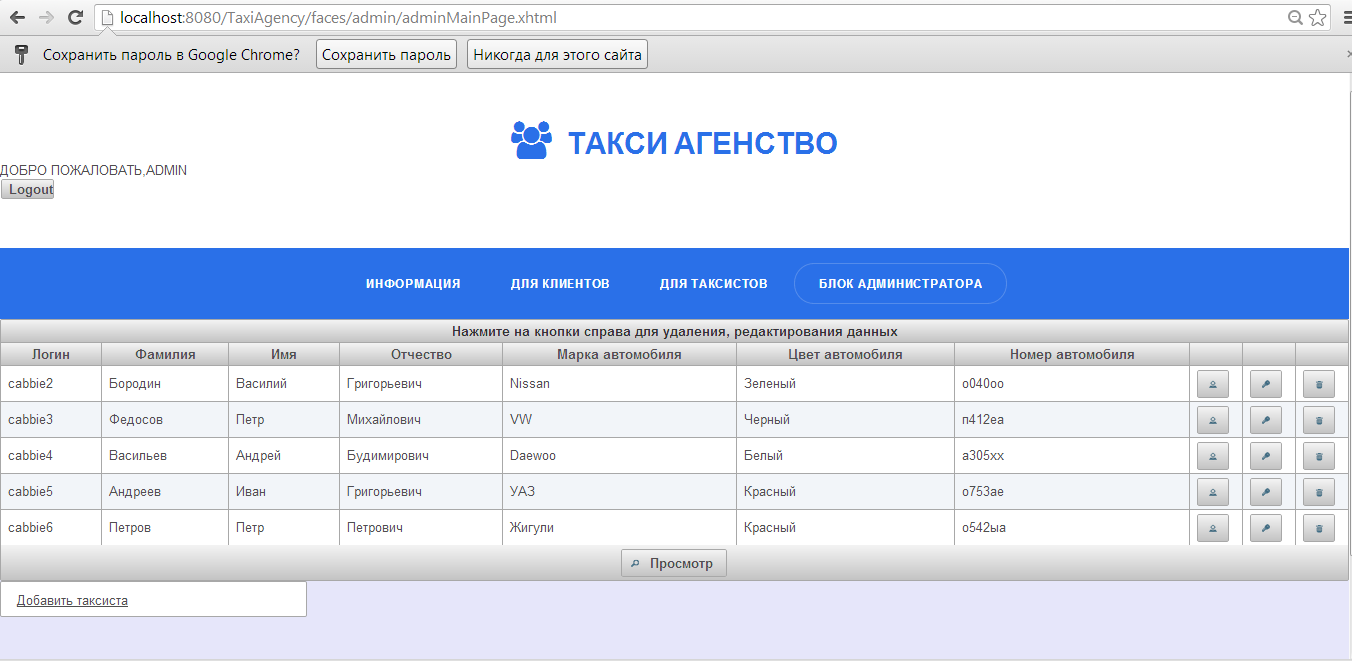
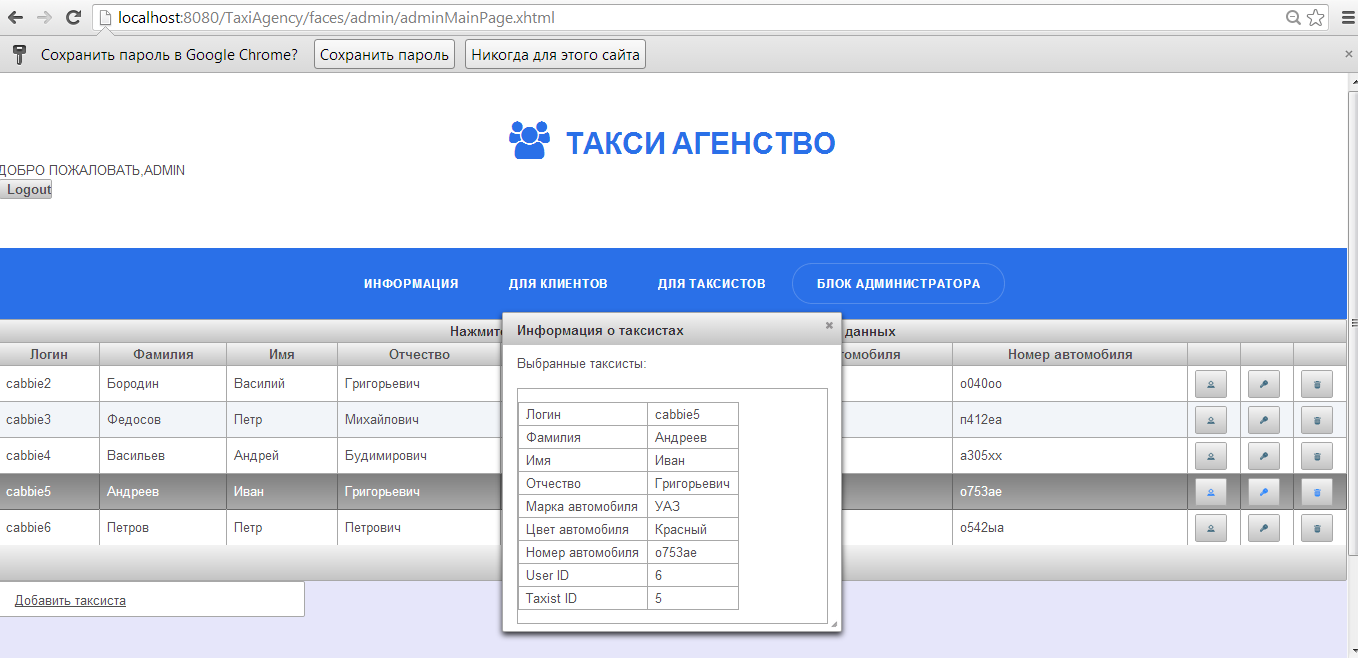


Рисунок24

Также мы можем просмотреть подробную информацию об определенном таксисте/таксистах, благодаря выделению нужной нам строки/строк и нажатию кнопки просмотр.

рисунок25

Нажимая на кнопку «редактировать таксиста» мы попадаем на страничку редактирования, в которой необходимо изменить необходимые поля и ввести пароль таксиста, либо вернуться на страницу администратора.

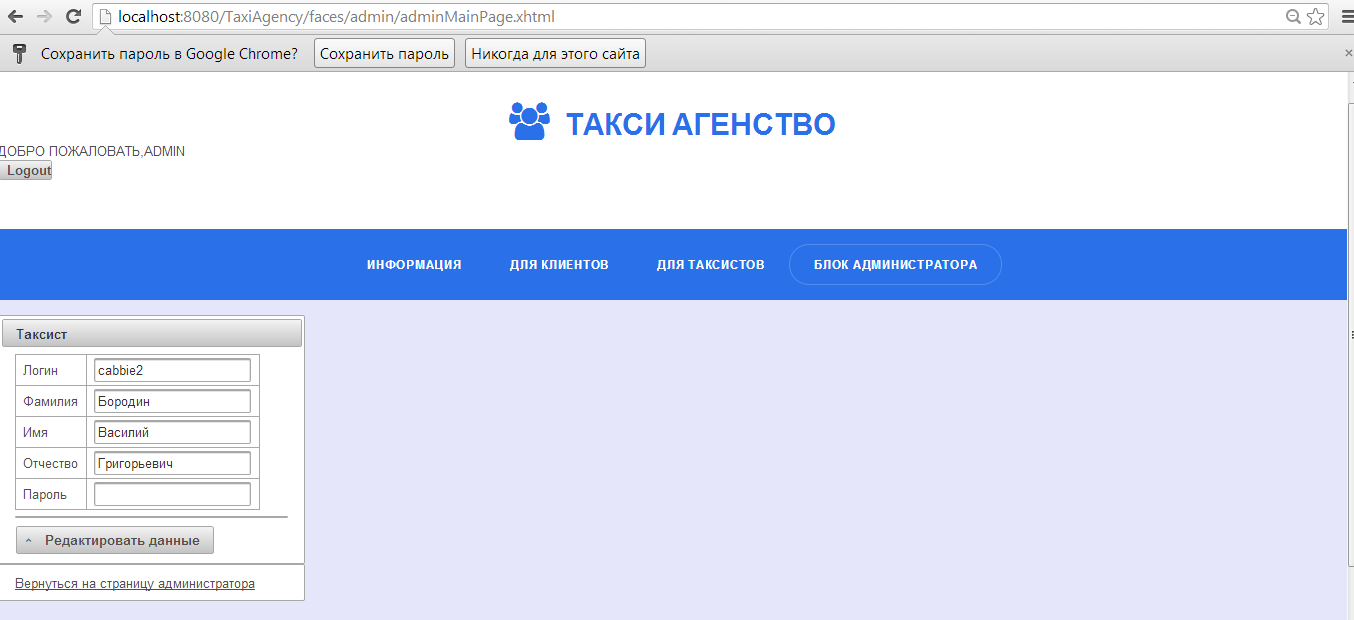


Рисунок26

Аналогично мы можем редактировать информацию о машинах, правда нам уже не надо вводить пароль определенного таксиста.

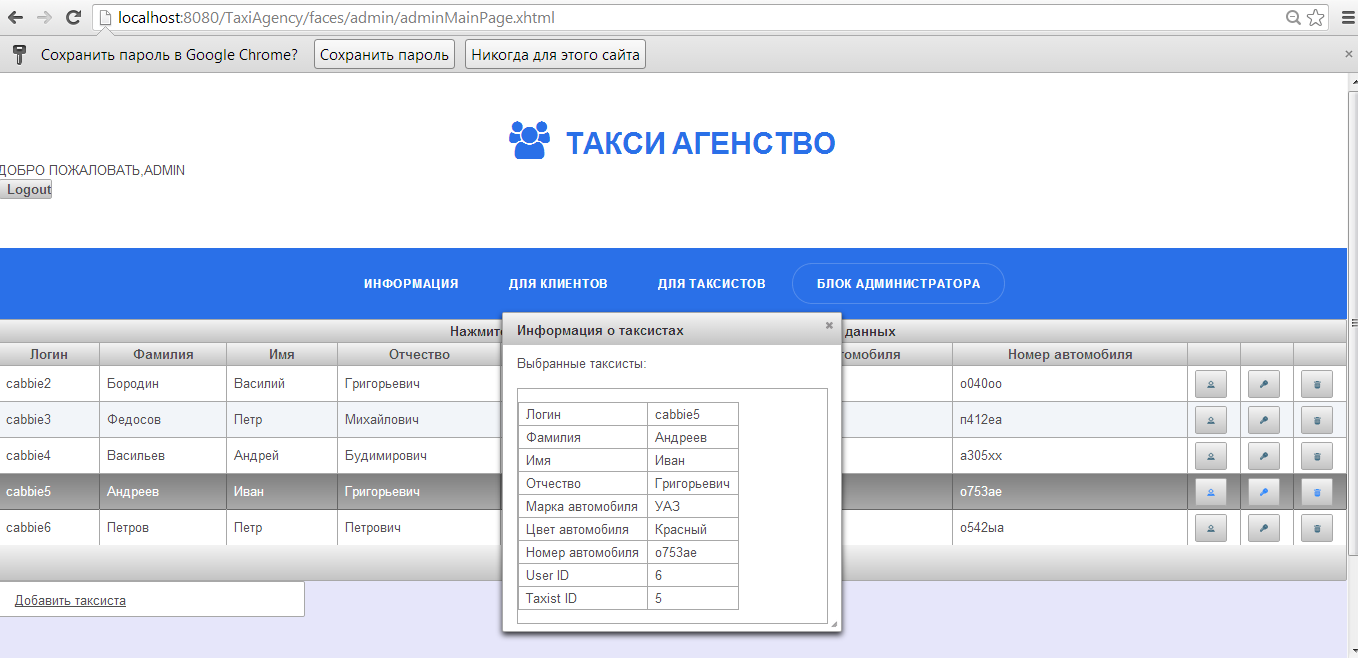


Рисунок27

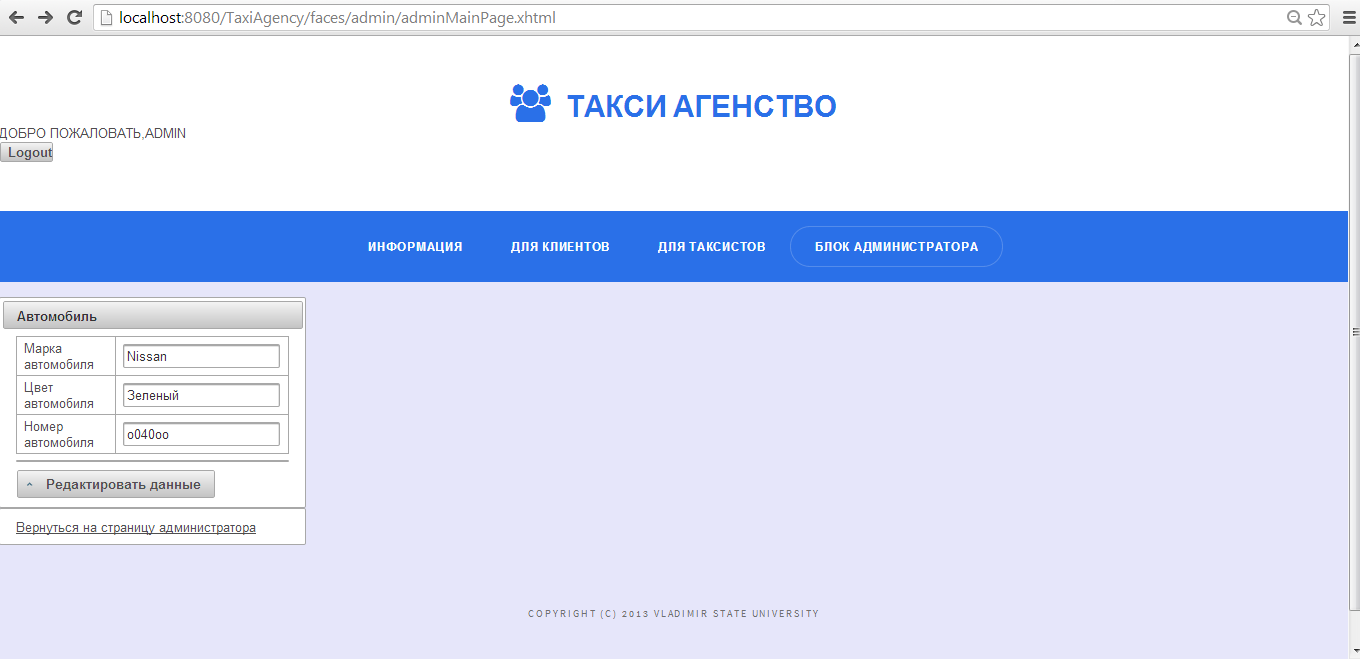


Рисунок28

Администратор имеет право удалить/уволить определенного таксиста, нажимая на кнопку удалить таксиста.

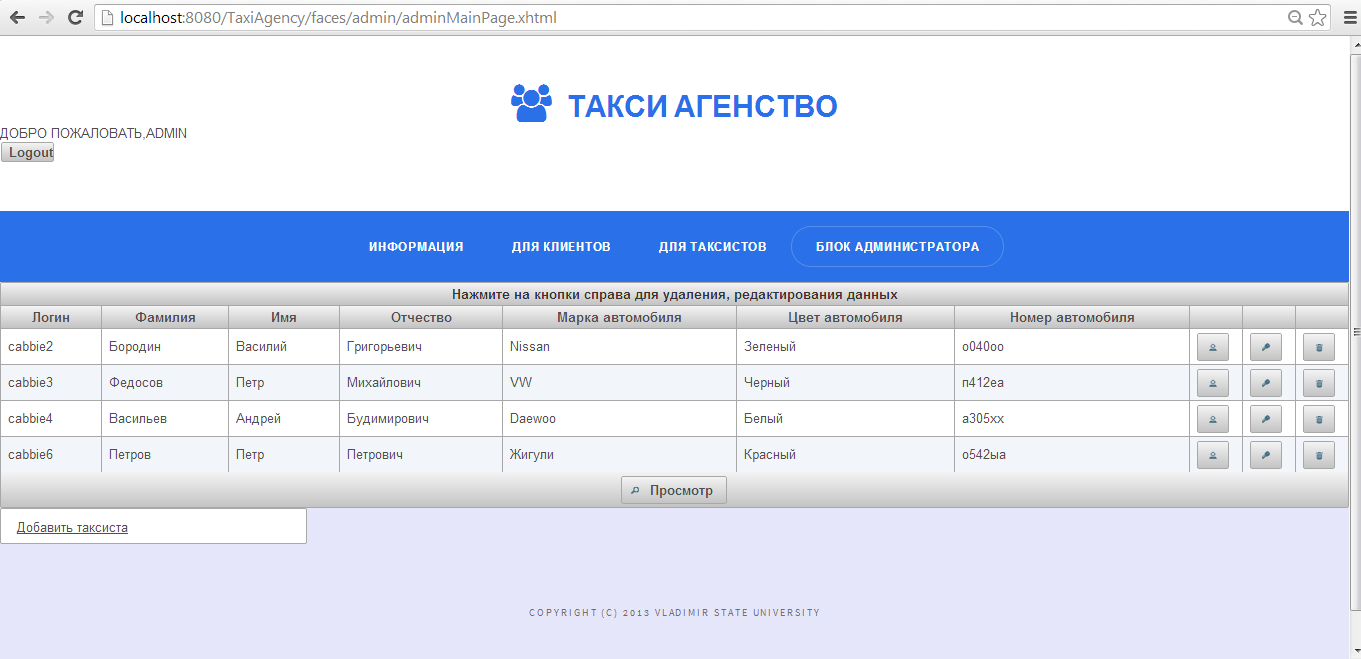


Рисунок29

И последняя функция администратора, он имеет право добавить нового таксиста,однако нам запрещено добавить таксиста с уже существующим логином.

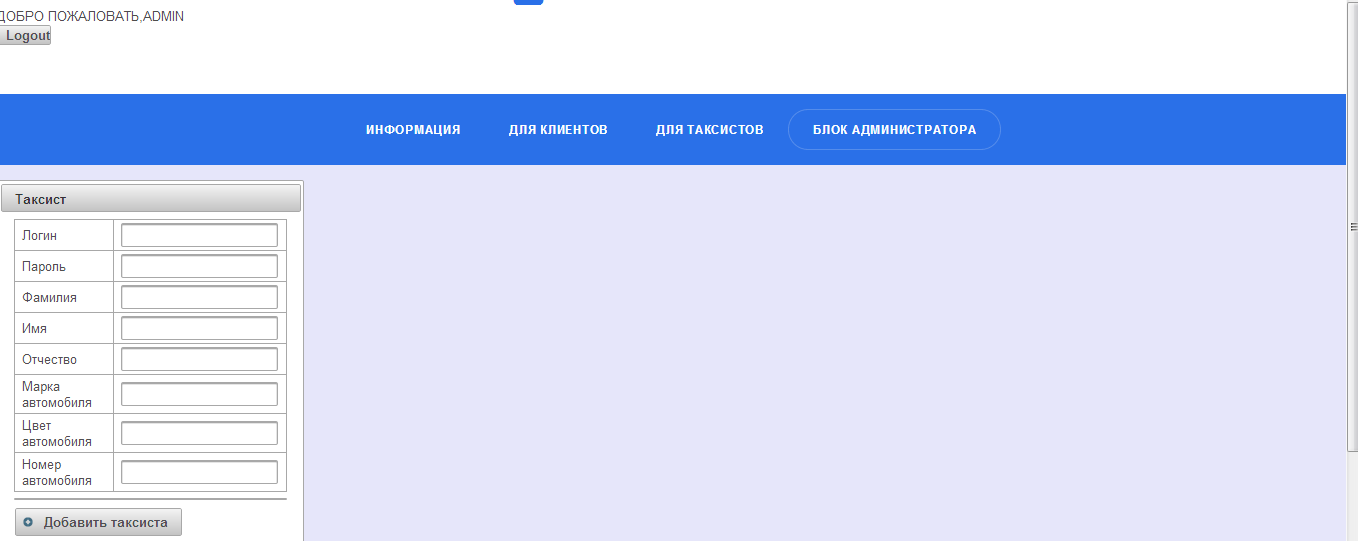


Рисунок30

# 

# 11. ЗАКЛЮЧЕНИЕ.

В ходе выполнения курсовой работы был спроектирован и создан рабочий прототип информационной системы «Служба такси». Создание прототипа происходило по плану, указанном в задании к курсовой работе. Каждый предыдущий этап разработки являлся основой для следующего этапа. Сначала был проведен анализ предметной области, на его основе была построена структура бизнес-логики, и структура веб-интерфейса. Затем, опираясь на структуру бизнес-логики, была проведена реализация бизнес-логики, и также, опираясь на структуру веб-интерфейс, был реализован веб-интерфейс. Последним завершающим этапом было соединение бизнес-логики и веб-интерфейса в одно целое. Все указанные технологии в задании к курсовой работе нашли своё применение в той или иной части информационной системы.

# 12.СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.

1. Будилов, В. А. Интернет-программирование на Java / В. А. Будилов – СПб.: БХВ-Петербург, 2003.

2. Дейтел, Х. М. Технологии программирования на Java 2 / Х. М. Дей-тел, П. Дж. Дейтел, С. И. Сантри – М.: ООО "Бином-Пресс", 2003.

3. Гери, Д. М. JavaServer Faces : Библиотека профессионала / Дэвид М. Гери, Кей С. Хорстманн – М. : Вильямс, 2008. – 576 с. – ISBN 978-5-8459-1396-8, 978-0-13-173886-7

4. <http://www.mkyong.com/>