|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ»

ДИСЦИПЛИНА «БАЗЫ ДАННЫХ»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Система идентификации автомобиля по номерному знаку на пропускном пункте\_\_**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** М. М. Фомин

(Подпись, дата)

Студент ИУ6-43 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  И.М.Мамадаев

(Подпись, дата)

*2018 г*

**Оглавление**

Техническое задание……………………………………………………………...3

Глава 1. Проектирование…………………………………………………………5

1.1 Анализ предметной области…………………………………………..5

1.2 Бизнес процессы пользователей информационной системы…….…7

1.3 Выделение сущностей…………………………………………………8

1.4 Проектирование схемы базы данных…………………………………9

Глава 2. Реализация………………………………………………………...……10

2.1 Пользовательская аутентификация………………………………….10

2.2 Добавление гостя……………………………………………………...12

2.3 Добавление жильца…………………………………………………...13

2.4 Добавление автомобиля………………………………………………14

2.4 Создание пропуска……………………………………………………15

2.4 Добавление КПП и района …………………………………………16

2.5 Постраничная авторизация…………………………………………..17

Глава 3. Инструкция пользователя «Сотрудник охраны»………...…………..20

Глава 4. Инструкция пользователя «Житель»………………….....……….…..26

Глава 5. Инструкция пользователя «Администратор»……….…...…………..27

Заключение………………………………………………………………………29

Приложения……………………………………………………………………...30

Приложение А…………………………………………………………….30

Приложение Б……………………………………………………………..33

**Техническое задание на разработку системы идентификации автомобиля по номерному знаку на пропускном пункте**

1. Введение
   1. Полное название

Система идентификации автомобиля по номерному знаку на пропускном пункте

* 1. Общие сведения

Система предназначена для хранения, добавления и быстрого поиска данных об автомобилях.

1. Назначения и цели создания системы

Система станет инструментом охраны на пропускном пункте, с помощью неё они смогут проверять всю информацию о пропускаемом ими автомобиле, а именно: к какому дому он направляется, на какое время.

Основным предназначением системы является возможность добавления жителями частной территории номеров автомобилей их гостей. Данная функция позволит уменьшить время задержки на контрольно-пропускном пункте, так как номер автомобиля будет уже храниться в базе данных.

Основными целями создания данной системы являются:

* Создание единой базы жителей частной территории;
* Создание базы частных территорий города;
* Создание базы гостей, которых также требуется пропускать на КПП;
* Снижение времени пропуска на КПП;
* Систематизация информации о пропускаемом автомобиле.

1. Общие требования к системе
   1. Требования к базе данных

В базе данных должна храниться информация:

* О жителях частной территории;
* О частных территориях города;
* Об автомобилях гостей;
* О количестве гостей у каждого жителя.
  1. Требования к отчётам

Система должна давать возможность получить следующие отчеты:

* Количество гостей у жителя и информация о них;
* Информация о жителях частной территории;
* Информация о частных территориях города;
* Статистика по гостям каждой частной территории.
  1. Требования к пользователям

В системе должны быть предусмотрены два вида пользователей:

Житель частной территории. Данный пользователь должен иметь возможность:

* Добавлять гостей
* Добавлять других жителей с автомобилями из его дома
* Изменять данные о жителях и гостях своего дома

Охрана на КПП. Данный пользователь должен иметь возможность:

* Получать информацию об автомобиле и его владельце по номеру
* Получать статистику по посещениям о каждом доме
* Получать статистику о посещениях гостей
* Вводить в систему новых сотрудников охраны
* Удалять гостей
  1. Требования к надёжности

Система должна обеспечить надежное хранение информации (защиту от сбоев, защиту от несанкционированного доступа).

* 1. Требования к составу и параметрам технических программных средств

Компьютер, поддерживающий операционную систему «Windows 7» и выше.

1. Требования к программной документации

В состав документации входят пояснительная записка, руководство пользователя, альбом печатных форм.

1. Порядок оформления и предъявления результатов работы

Результаты представляются в виде отлаженной информационной системы, выполняющей функции данного технического задания.

**Глава 1. Проектирование**

**1.1 Анализ предметной области**

Проанализируем техническое задание для понимания, какие бизнес-процессы будут реализованы в данной системе и какие данные должны находиться в базе данных.

Во многих городах России существуют частные районы, ограниченные забором и осуществляющим ограниченный въезд на территорию (рис. 1)



Рисунок 1 – Пример частной территории

Для въезда на подобную территорию необходимо пройти контрольно-пропускной пункт, для прохода через который необходимо предъявить пропуск, либо, для лиц, передвигающихся на транспортном средстве, иметь номер автомобиля в базе данных (рис.2), чтобы автоматизированная система смогла открыть шлагбаум при приближении автомобиля.

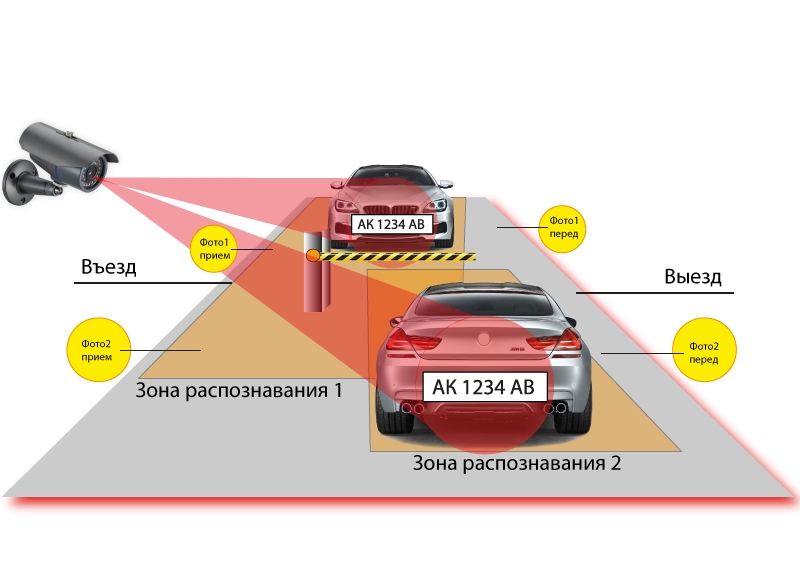


Рисунок 2 – Пример системы распознания автомобильного номера

* + 1. **Люди**

Так как основной функцией любого контрольно-пропускного пункта является допуск определенных людей на частную территорию, то в базе данных необходима таблица, в которой будут находиться данные о людях, которых охране нужно будет пропускать или же, наоборот, не пропускать на территорию.

* + 1. **Транспортные средства**

В данном разделе будет вестись учёт транспортных средств, на которых жители частной территории, а также их гости будут проезжать через КПП. В данной таблице будет находиться основная информация о транспортном средстве: марка автомобиля или мотоцикла, кузов, цвет и, конечно, номер транспортного средства.

* + 1. **Сотрудники охраны**

В базе данных также должна храниться информация о сотрудниках охраны, их смена и номер контрольного пункта частной территории, так как их может быть несколько.

* + 1. **Пропуски**

В данном разделе будет находиться информация и пропускаемом человеке, кем он является – постоянным жителем или гостем, на каком транспортном средстве он передвигается. Данная таблица в основном будет необходима для людей, которые будут проходить КПП без автомобиля.

* 1. **Бизнес процессы пользователей информационной системы**

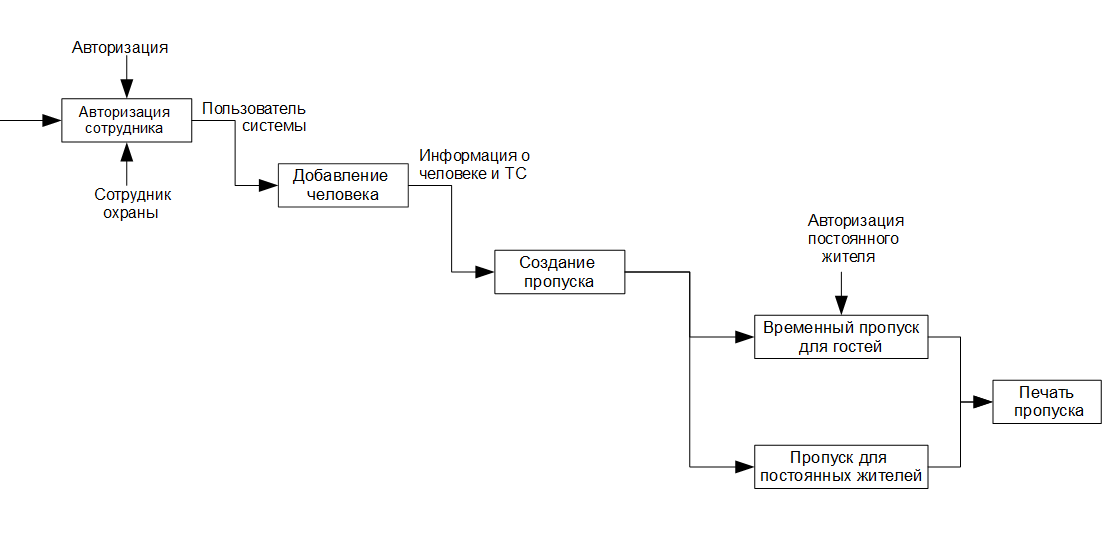


Рисунок 3 – БП «Создание временного или постоянного пропуска»

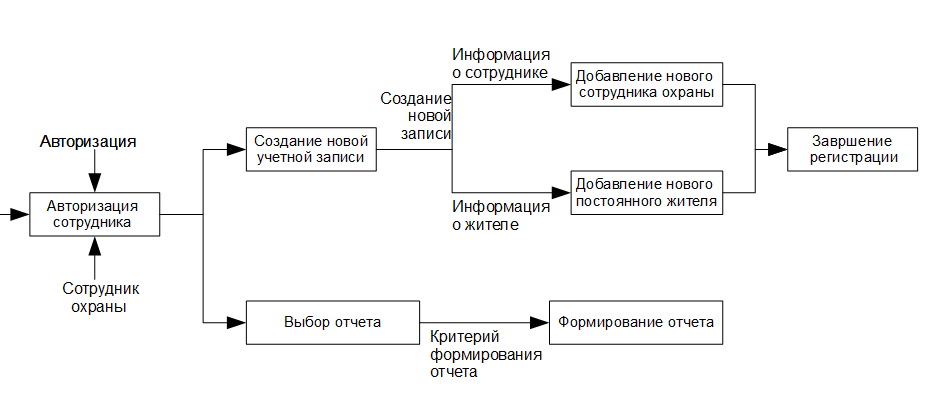


Рисунок 4 – БП пользователя «Сотрудник охраны»

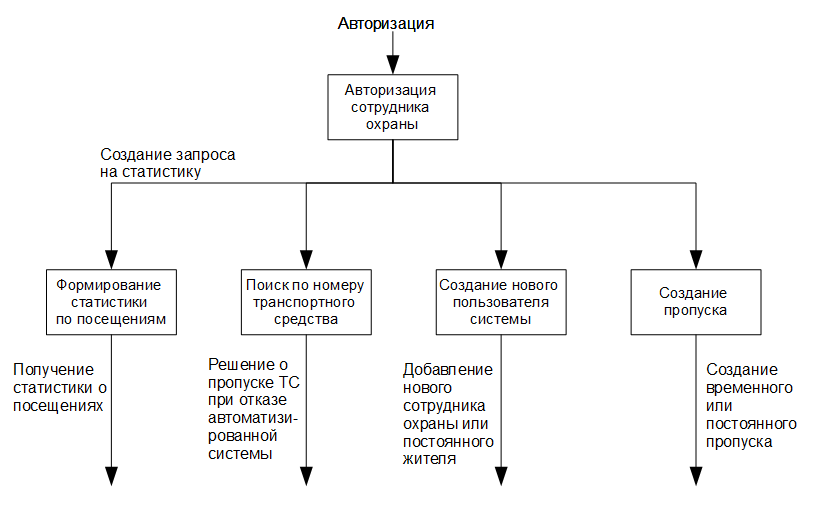


Рисунок 5 – Диаграмма всех возможностей пользователя «Сотрудник охраны»

* 1. **Выделение сущностей**

Основываясь на бизнес-процессах, описанных в п.1.2, можно выделить сущности, которые будет необходимо реализовать при создании базы данных.

* Транспортные средства (автомобили)
* Сотрудники охраны
* Контрольно-пропускные пункты
* Люди (Постоянные жители и гости)
* Пропуски

Кроме того, для создания сервиса авторизации и аутентификации понадобятся отдельные сущности: «пользователь-роль» и «роль-страница», для реализации разграничения прав и возможностей различных пользователей в системе.

Также будут добавлены сущности «Города» и «Районы», так как в различных городах могут быть несколько подобных различных частных территорий, для проезда в которые необходим пропуск.

* 1. **Проектирование схемы базы данных**

На рисунке 6 приведена схема базы данных без реализации разграничения прав доступа.

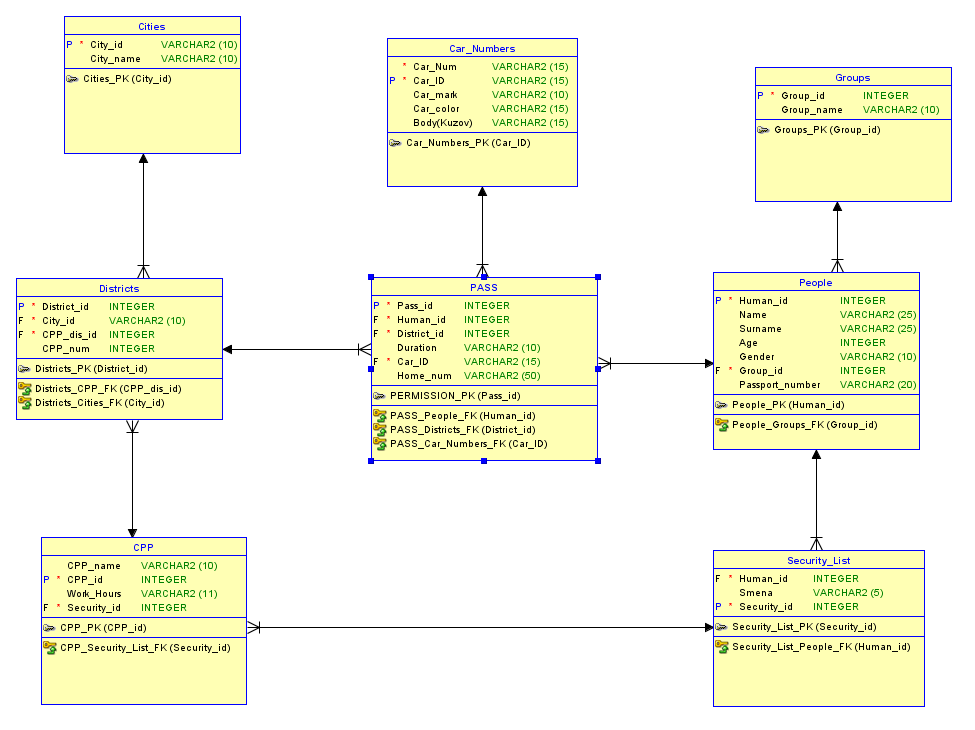


Рисунок 6 – Схема базы данных

Код формирования всех объектов БД представлена в приложении А.

**Глава 2. Реализация**

**2.1 Пользовательская аутентификация**

В связи с особенностями среды разработки ORACLE APEX 5, стандартная, встроенная пользовательская аутентификация работает только с пользователями непосредственно среды APEX

Первая страница, которую видит пользователь при входе в систему – страница 101 генерируется автоматически. Для хранения различных пользователей с различными правами доступа была создана таблица USER\_ACCOUNT (рисунок 7).

Создадим также таблицу ROLE\_PAGE для того чтобы разграничить права доступа к различным страницам. Схема данной таблицы и введенные данные приведены на рисунке 8-9.

Далее, создадим пакет для обработки логина и пароля пользователя (код представлен в приложении Б). Добавим функцию, которая покажет пользователю причину, по которой он не смог получить доступ к своему аккаунту.

Затем, создадим в системе форму с сообщениями об ошибках при входе, добавив на страницу поле для вывода сообщения (рисунок 10) и изменим процесс Login (рисунок 11) на новую схему авторизации.

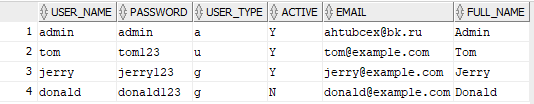


Рисунок 7 – Таблица USER\_ACCOUNT

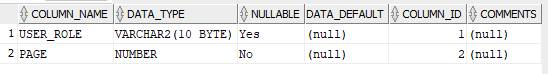


Рисунок 8 – Схема таблицы ROLE\_PAGE

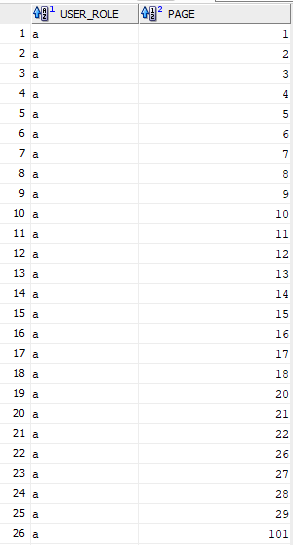


Рисунок 9 – Страницы доступа для пользователя с ролью «а» (администратор)

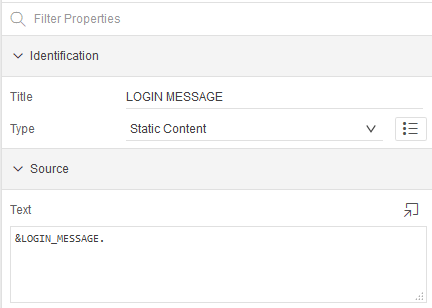


Рисунок 10 - Блок вывода сообщения при входе

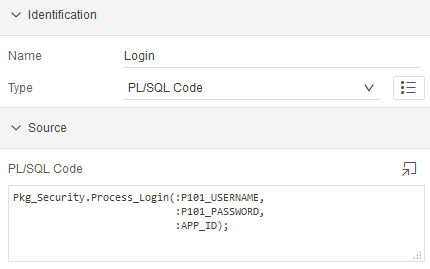


Рисунок 11- Настройка процесса Login

**2.2 Добавление пользователя**

Функция добавления пользователя будет доступна только аккаунтам с доступом «а» - администраторам. Форма создания нового пользователя, а также форма создания аккаунта приведена на рисунках 12-13.

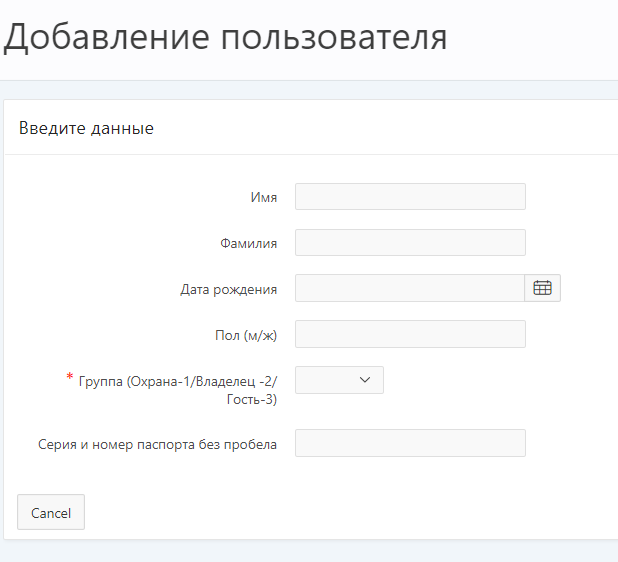


Рисунок 12 – Форма создания нового пользователя системы

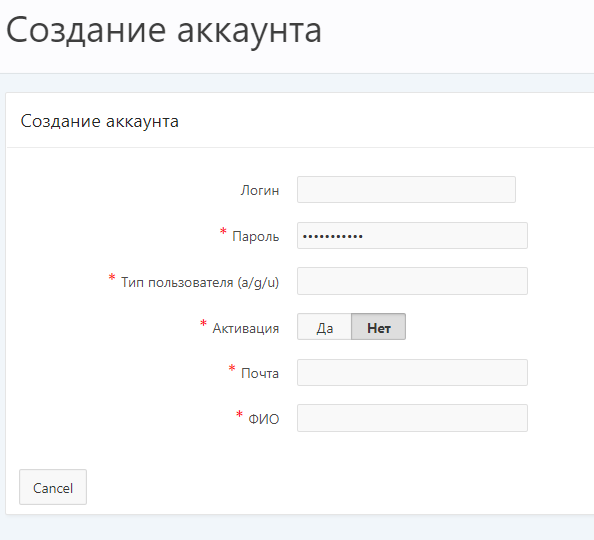


Рисунок 13 – Форма создания аккаунта

**2.3 Добавление гостя**

Для добавления гостя, то есть пользователя, который будет фигурировать в базе данных, но иметь доступа к системе не будет, была создана соответствующая страница (рисунок 14), доступная как пользователям класса «owners» (люди, проживающие в частном районе), так и сотрудники охраны класса «guards».

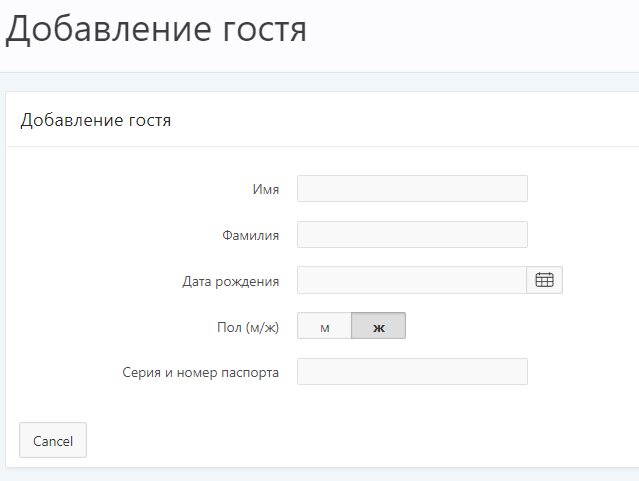


Рисунок 14 – Форма добавления гостя

**2.4 Добавление жильца**

Добавление жильца, как и добавление гостя, требует небольшого количества информации. Различия между этими формами заключается лишь в том, что гостя могут добавить только жители района, а самого жителя района может добавить только администратор или сотрудник соответствующего поста охраны. Форма добавления жителя района приведена на рисунке 15.

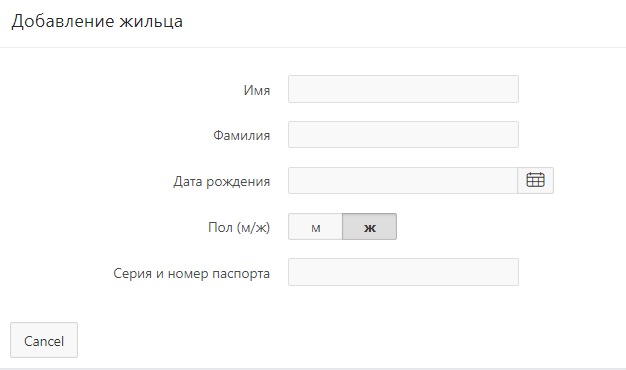


Рисунок 15 – Форма добавления жителя

**2.5 Добавление автомобиля**

Так как некоторые пользователи будут иметь транспортные средства, то необходимо предусмотреть возможность добавления транспортного средства. Для этого была создана форма, приведенная на рисунке 16.

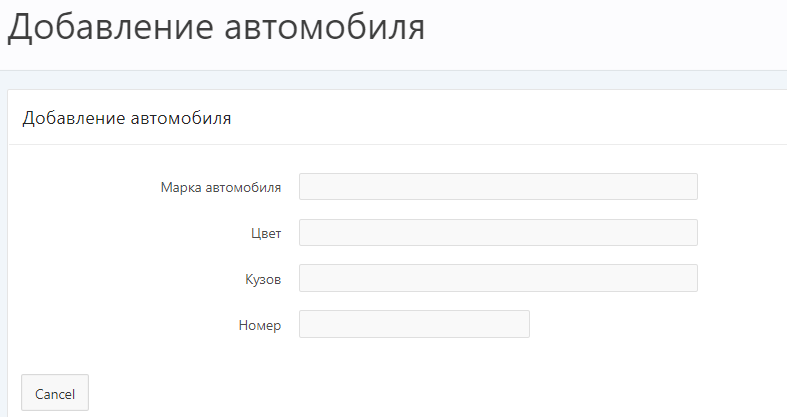


Рисунок 16 – Форма добавления автомобиля

**2.6 Создание пропуска**

Создание пропуска возможно лишь для пользователя, уже добавленного в базу данных. Таким образом, форма создание нового пропуска для пользователя и его автомобиля (если он имеется) приведена на рисунке 17. Код выборки автомобиля приведен на рисунке 18-19. Код для выборки данных для полей «пользователь» и «район» аналогичен.

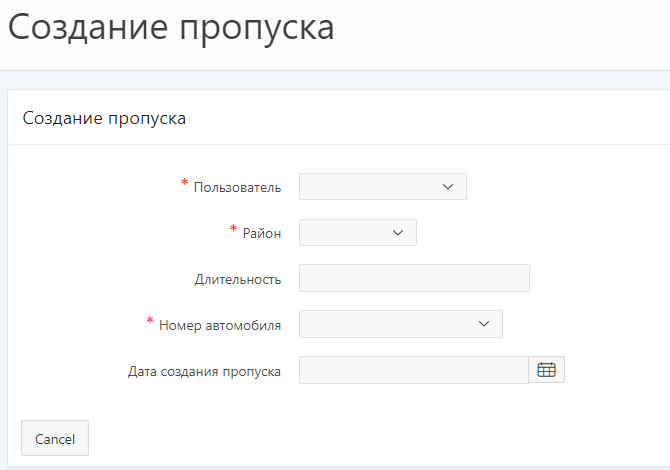


Рисунок 17 – Форма создания пропуска

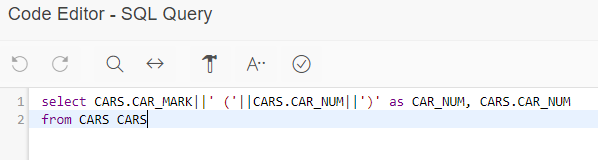


Рисунок 18 – SQL-код для выборки автомобилей

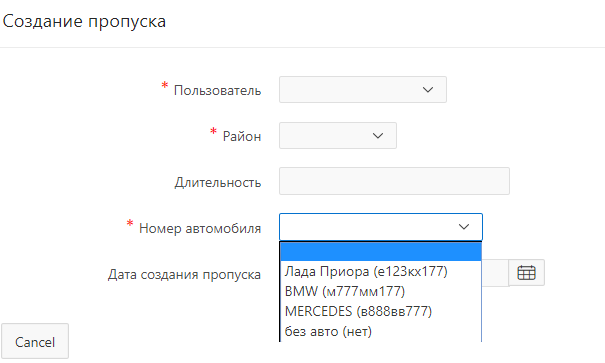


Рисунок 19 – Выборка автомобиля

**2.7 Добавление КПП и района**

Данные формы имеют идентичную странице «Добавление гостя» структуру, поэтому приведу лишь изображения того, как они выглядят на рисунках 20-21.

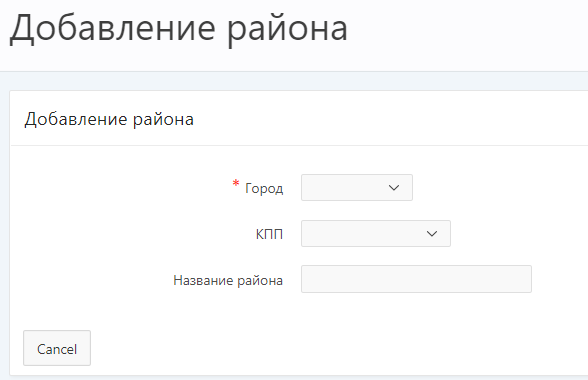


Рисунок 20 – Форма добавления района

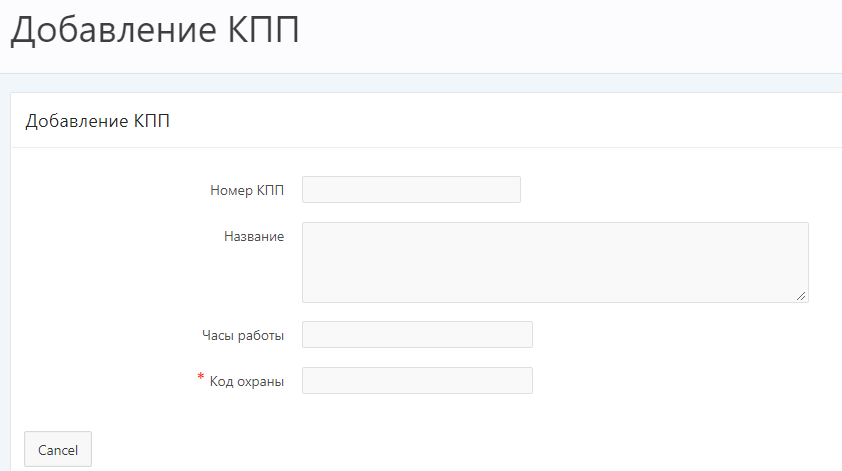


Рисунок 21 – Форма добавления КПП

**2.8 Постраничная авторизация**

Для создания постраничной авторизации была создана таблица «Role-Page», в которой показано, какому пользователю разрешен доступ к выбранной странице. Воспользуемся APEX представлением APEX\_APPLICATION\_LIST\_ENTRIES, в котором есть информация, какие страницы прикреплены к каким объектам навигации. Для того, чтобы не поддерживать отдельно еще одну таблицу со связью Роль -> Объект навигации, создадим представление, связывающее роли и List Entries на основе связи Роль –> Страница (код представлен в приложении Б). Затем изменим навигационное меню в компонентах программы, чтобы каждый пользователь видел только те страницы, к котором он имеет доступ. Меню доступа администратор, сотрудника охраны и жильца приведены соответственно на рисунках 22-24.

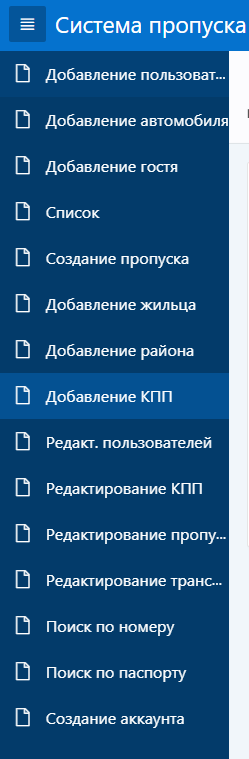


Рисунок 22 – Меню администратора

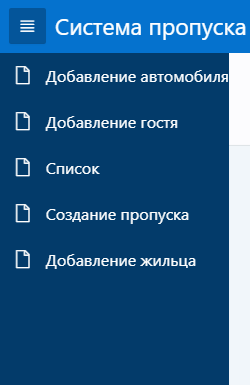


Рисунок 23 – Меню сотрудника охраны

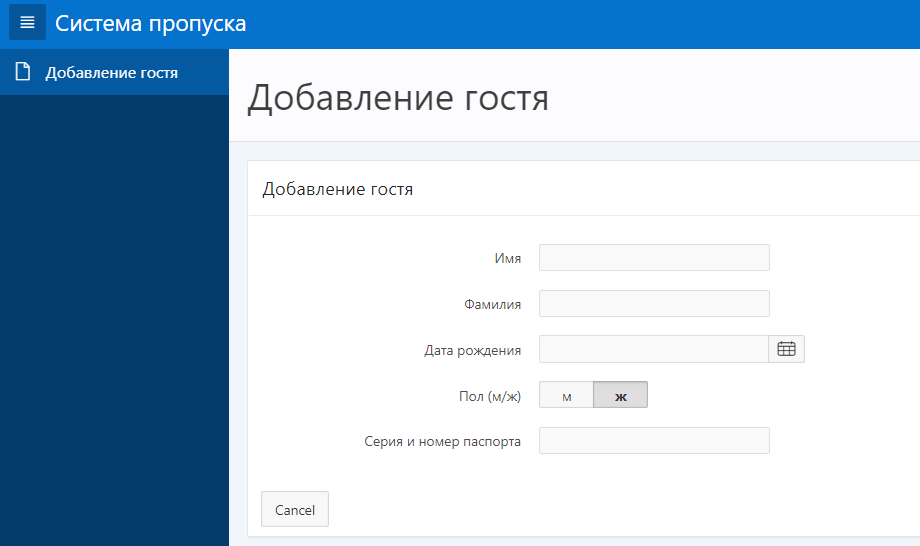


Рисунок 24 –Страница приложения для жителя

**Глава 3. Инструкция пользователя «Сотрудник охраны»**

**Введение**

Основная цель создания данного приложения и базы данных – уменьшение времени на КПП при въезде в район, ограниченный шлагбаумом. Таким образом, для начала работы сотрудника охраны ему необходимо пройти авторизацию по логину и паролю, созданному администратором.

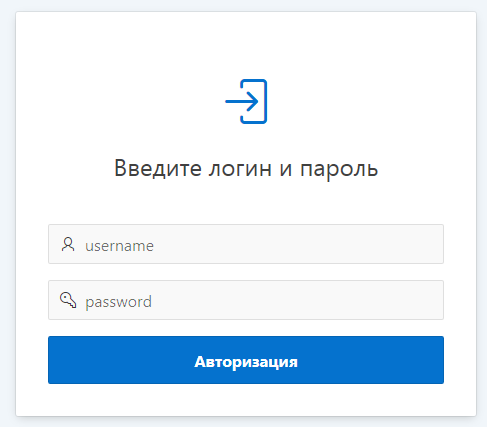


Рисунок 1 - Страница авторизации

После успешного выполнения аутентификации вы увидите начальную страницу и меню с Вашими возможностями в левой части страницы (рис. 2).

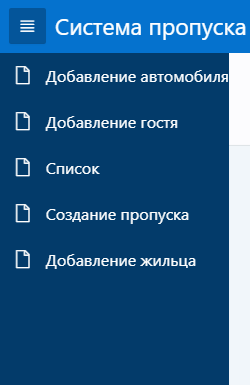


Рис. 2 Начальная страница и навигационное меню

**Добавление автомобиля, жильца и гостя**

В разделе «Добавление автомобиля» вам необходимо ввести данные транспортного средства, изображенные на рисунке 2.

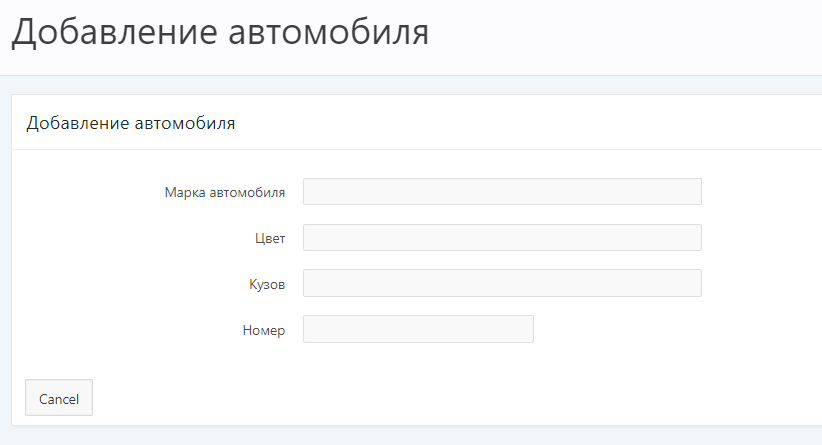


Рисунок 2 – Форма добавления автомобиля

В разделе «Добавление гостя» и «Добавление жильца» необходимо ввести, соответственно, данные гостя и жильца. Интерфейсы страниц приведены на рисунках 3-4.

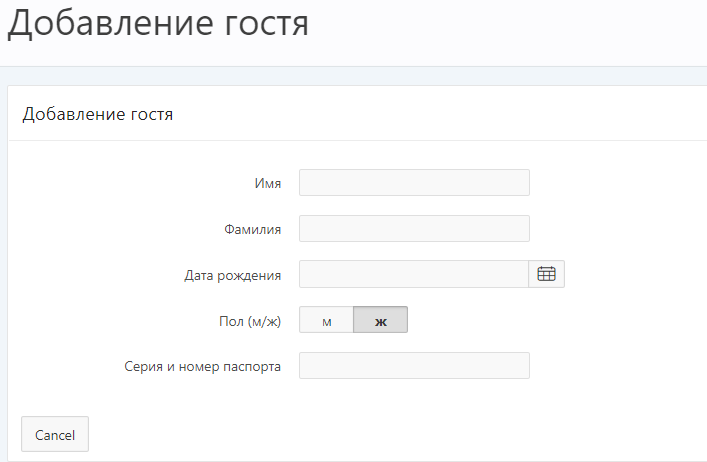


Рисунок 3 – Форма добавления гостя

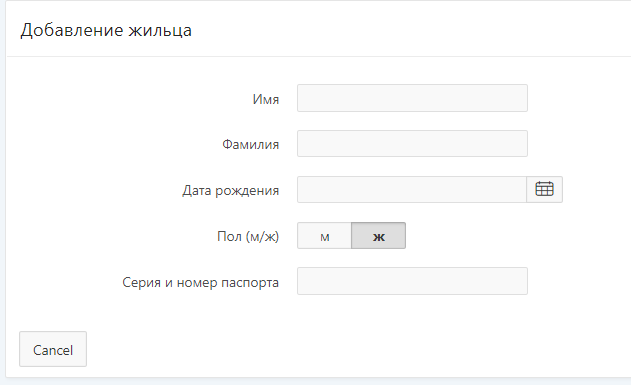


Рисунок 4 – Форма добавления жильца

**Список пользователей**

Для того, чтобы попасть на страницу списка всех людей, нажмите на вкладку «Список», в котором будет находиться информация о всех пользователях и их транспортных средствах.

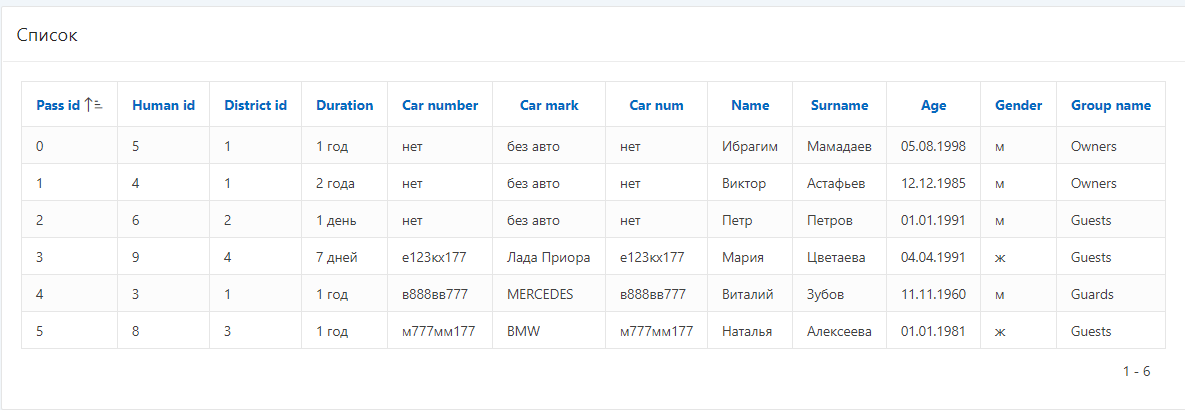


Рисунок 5 – Форма, содержащая список пользователей

**Создание пропуска**

Основной страницей, на которой будет происходить большая часть работы сотрудника охраны, будет «Создание пропуска». Для доступа на данную страницу необходимо нажать на соответствующее поле в навигационном меню слева.

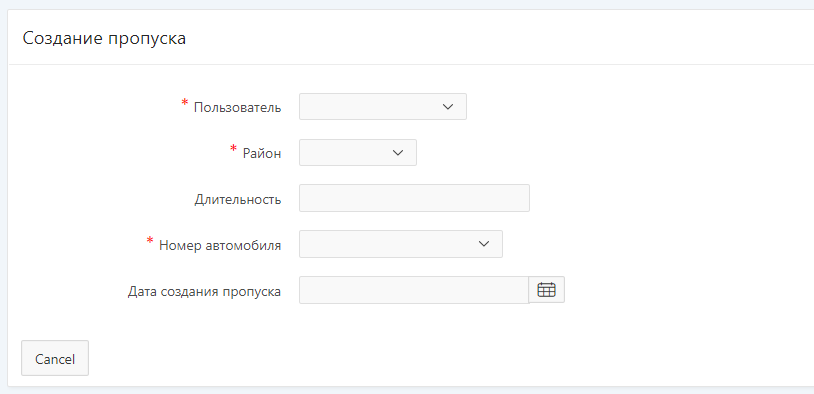


Рисунок 6 – Форма создания пропуска

Для создания пропуска необходимо:

1. Выбрать пользователя, для которого будет создаваться пропуск (рисунок 7).

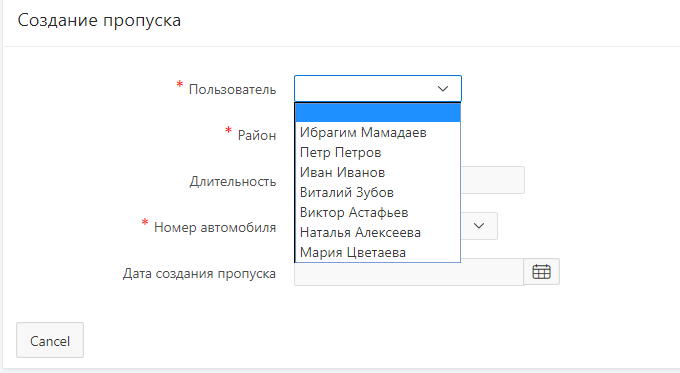


Рисунок 7 – Форма выбора пользователя

2. Выбрать район, в который будет пропускаться пользователь (рисунок 8).

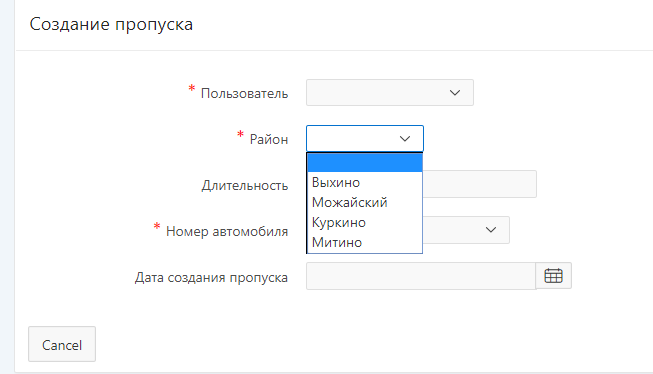


Рисунок 8 – Форма выбор района

3. Ввести длительность действия пропуска до прекращения его действия.

4. Выбрать автомобиль из списка автомобилей, введенных ранее (рисунок 9).

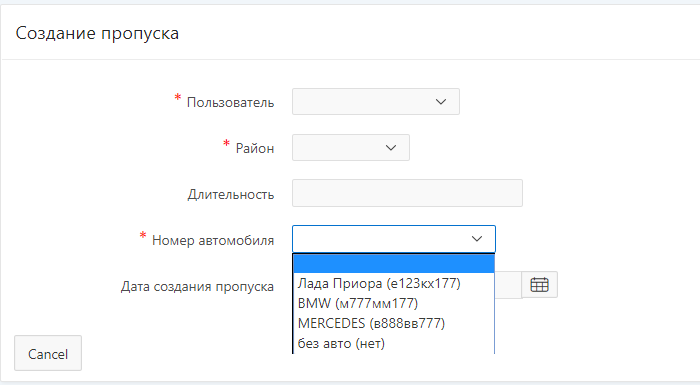


Рисунок 9 – Форма выбора автомобиля

5.Выбрать дату создания пропуска, с которой будет исчислять срок пропуска (рисунок 10).

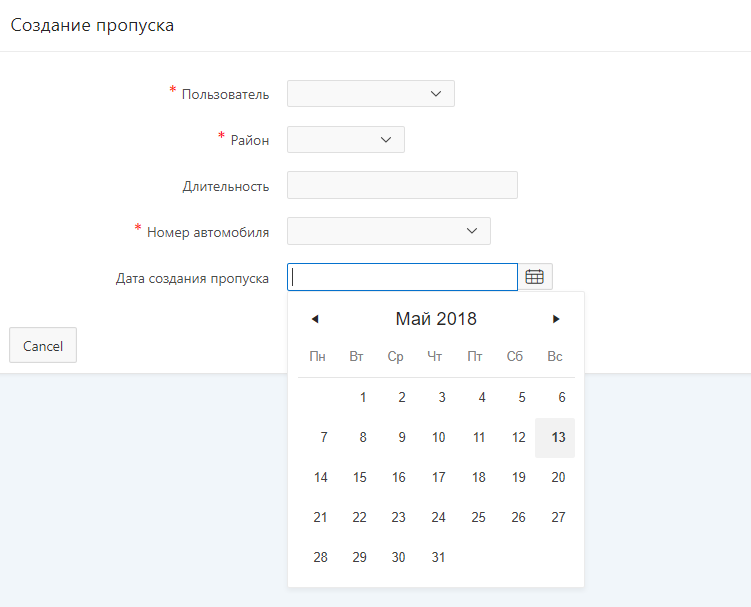


Рисунок 10 – Форма выбора даты создание пропуска

**Глава 4. Инструкция пользователя «Житель»**

Для пользователя «житель» доступна возможность в случае необходимости добавления гостя, который уже находится на КПП или планирует посетить ваш район.

Для этого необходимо пройти авторизацию под своей учетной записью (рисунок 1).

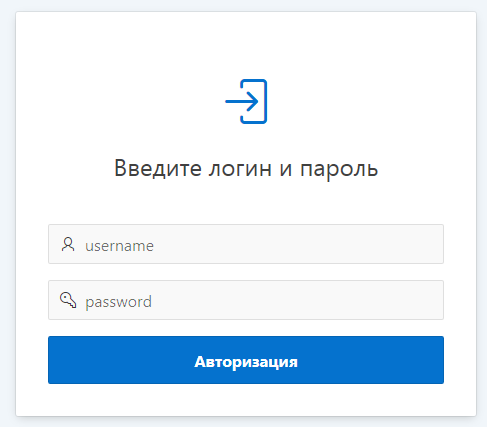


Рисунок 1 - Страница авторизации

В разделе «Добавление гостя» необходимо ввести данные гостя. Интерфейс страницы приведены на рисунке 2. После подтверждения создания, гость, ожидающий пропуска на КПП, получит свой пропуск от сотрудника охраны.

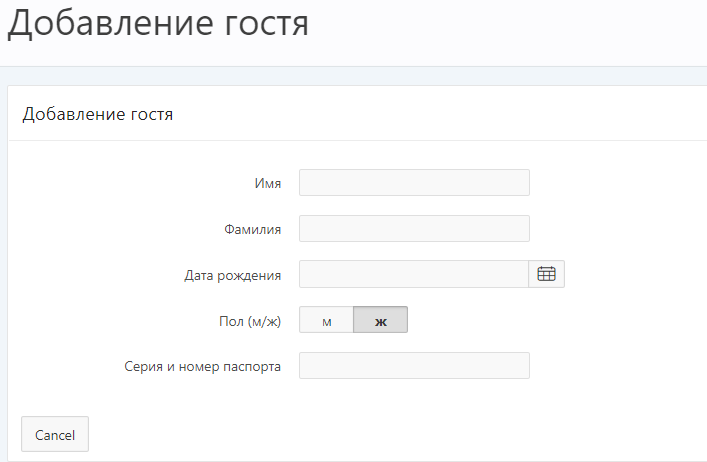


Рисунок 3 – Добавление гостя

**Глава 5. Инструкция пользователя «Администратор»**

Для администратора в данной системе имеется доступ ко всем страницам, доступным другим видам пользователей, а также страницы для добавления и создания аккаунтов, а также редактирование данных в таблицах. Соответствующие страницы приведены на рисунках 1-5.

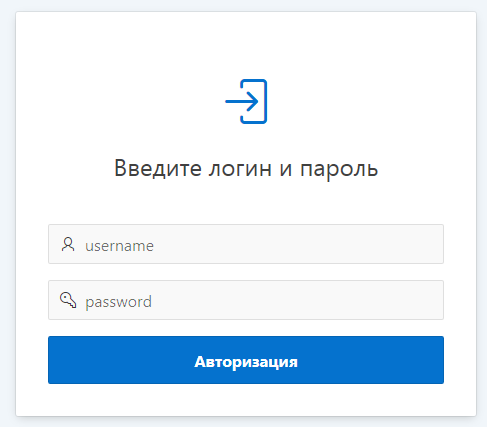


Рисунок 1 - Страница авторизации

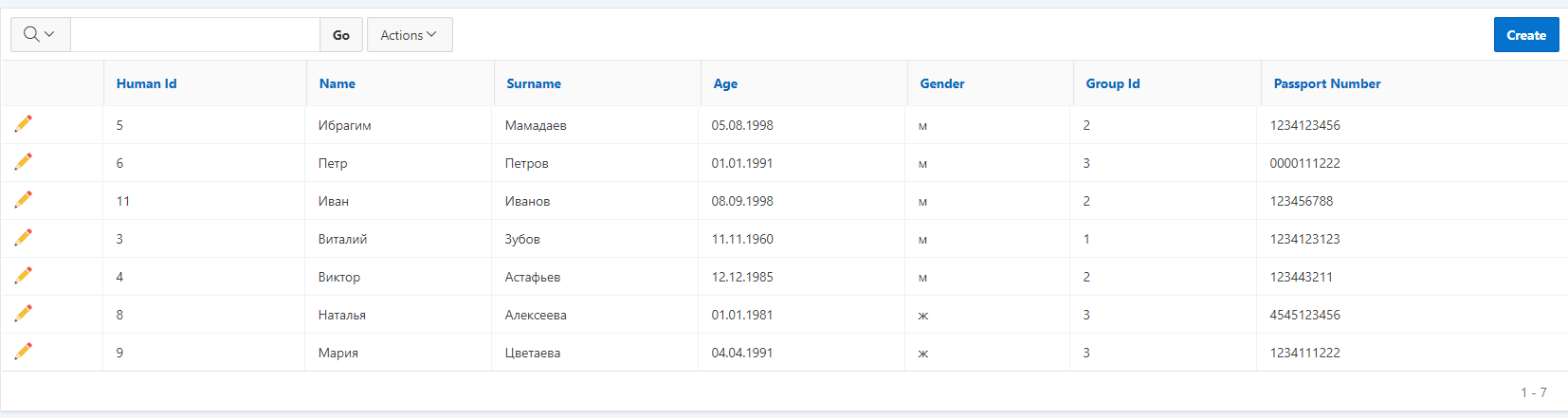


Рисунок 2 – Редактирование пользователей



Рисунок 3 – Поиск по номеру

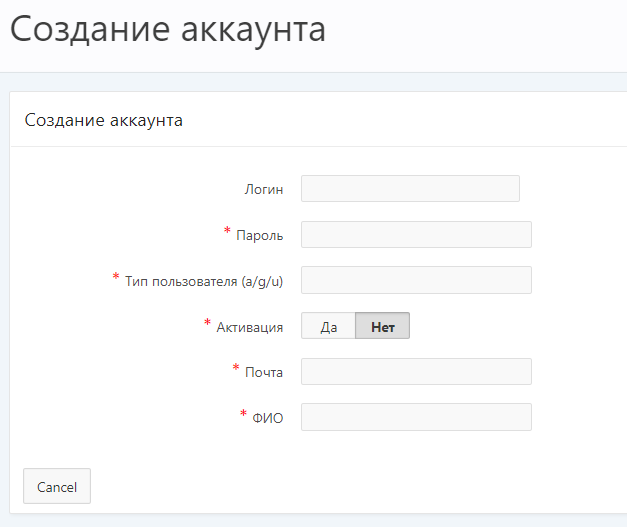


Рисунок 4 – Создание аккаунта

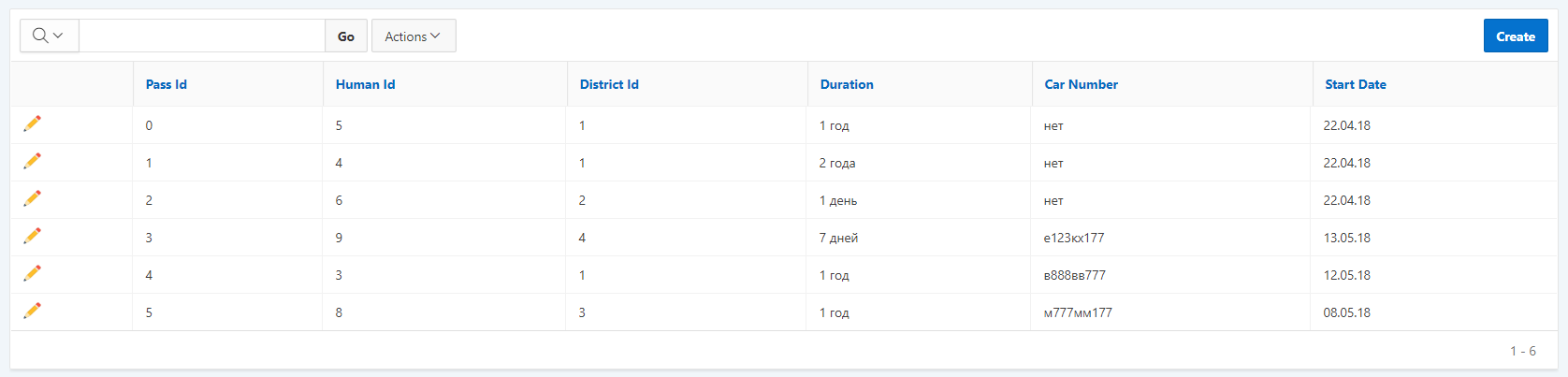


Рисунок 5 – Редактирование и поиск по таблице пропусков

**Заключение**

В результате работы была создана система идентификации автомобиля по номерному знаку на пропускном пункте, в основе которой находится база данных. Основа работы базы данных – СУБД Oracle Database Express Edition 11g, интерфейсы разработаны при помощи Oracle APEX версии 5.1.

Во время выполнения работы были проведены следующие действия:

* Исследование предметной области, изучение возможных сущностей и связей между ними;
* Проектирование базы данных, её реализация и доработка;
* Разработка интерфейсов пользователей;
* Разработка методов авторизации и доступа для разных типов пользователей;
* Подключение и настройка сервиса для создания печатных форм;

Использование данной системы позволит автоматизировать работу сотрудников охраны по пропуску автомобилей на КПП. Время задержки гостей, чьих автомобилей нет в базе данных сведено к минимуму, путем добавления возможности дистанционно вносить информацию в соответствующую таблицу.

Однако для полного функционала, а именно автоматизации работы шлагбаума, данной системы недостаточно, так как требуются и другие функции, такие как распознание камерой номера автомобиля и автоматическое открытие шлагбаума, которые не могу быть реализованы в данной среде.

Таким образом, можно сделать вывод, что использование данной системы возможно, но её функционала не недостаточно для полноценной автоматизации работы КПП.

**Приложения**

**Приложение А**

Скрипт, формирующий объекты БД приведен в листинге 1

Листинг 1 – Формирование объектов базы данных

-- Generated by Oracle SQL Developer Data Modeler 17.2.0.188.1059

-- at: 2018-03-11 14:28:12 GMT+03:00

-- site: Oracle Database 11g

-- type: Oracle Database 11g

CREATE TABLE car\_numbers (

car\_num VARCHAR2(15) NOT NULL,

car\_id VARCHAR2(15) NOT NULL,

car\_mark VARCHAR2(10),

car\_color VARCHAR2(15),

"Body(Kuzov)" VARCHAR2(15)

);

ALTER TABLE car\_numbers ADD CONSTRAINT car\_numbers\_pk PRIMARY KEY ( car\_id );

CREATE TABLE cities (

city\_id VARCHAR2(10) NOT NULL,

city\_name VARCHAR2(10)

);

ALTER TABLE cities ADD CONSTRAINT cities\_pk PRIMARY KEY ( city\_id );

CREATE TABLE cpp (

cpp\_name VARCHAR2(10),

cpp\_id INTEGER NOT NULL,

work\_hours VARCHAR2(11),

security\_id INTEGER NOT NULL

);

ALTER TABLE cpp ADD CONSTRAINT cpp\_pk PRIMARY KEY ( cpp\_id );

CREATE TABLE districts (

district\_id INTEGER NOT NULL,

city\_id VARCHAR2(10) NOT NULL,

cpp\_dis\_id INTEGER NOT NULL,

cpp\_num INTEGER

);

ALTER TABLE districts ADD CONSTRAINT districts\_pk PRIMARY KEY ( district\_id );

CREATE TABLE groups (

group\_id INTEGER NOT NULL,

group\_name VARCHAR2(10)

);

ALTER TABLE groups ADD CONSTRAINT groups\_pk PRIMARY KEY ( group\_id );

CREATE TABLE pass (

pass\_id INTEGER NOT NULL,

human\_id INTEGER NOT NULL,

district\_id INTEGER NOT NULL,

duration VARCHAR2(10),

car\_id VARCHAR2(15) NOT NULL,

home\_num VARCHAR2(50)

);

ALTER TABLE pass ADD CONSTRAINT permission\_pk PRIMARY KEY ( pass\_id );

CREATE TABLE people (

human\_id INTEGER NOT NULL,

name VARCHAR2(25),

surname VARCHAR2(25),

age INTEGER,

gender VARCHAR2(10),

group\_id INTEGER NOT NULL,

passport\_number VARCHAR2(20)

);

ALTER TABLE people ADD CONSTRAINT people\_pk PRIMARY KEY ( human\_id );

CREATE TABLE security\_list (

human\_id INTEGER NOT NULL,

smena VARCHAR2(5),

security\_id INTEGER NOT NULL

);

ALTER TABLE security\_list ADD CONSTRAINT security\_list\_pk PRIMARY KEY ( security\_id );

ALTER TABLE cpp

ADD CONSTRAINT cpp\_security\_list\_fk FOREIGN KEY ( security\_id )

REFERENCES security\_list ( security\_id );

ALTER TABLE districts

ADD CONSTRAINT districts\_cities\_fk FOREIGN KEY ( city\_id )

REFERENCES cities ( city\_id );

ALTER TABLE districts

ADD CONSTRAINT districts\_cpp\_fk FOREIGN KEY ( cpp\_dis\_id )

REFERENCES cpp ( cpp\_id );

ALTER TABLE pass

ADD CONSTRAINT pass\_car\_numbers\_fk FOREIGN KEY ( car\_id )

REFERENCES car\_numbers ( car\_id );

ALTER TABLE pass

ADD CONSTRAINT pass\_districts\_fk FOREIGN KEY ( district\_id )

REFERENCES districts ( district\_id );

ALTER TABLE pass

ADD CONSTRAINT pass\_people\_fk FOREIGN KEY ( human\_id )

REFERENCES people ( human\_id );

ALTER TABLE people

ADD CONSTRAINT people\_groups\_fk FOREIGN KEY ( group\_id )

REFERENCES groups ( group\_id );

ALTER TABLE security\_list

ADD CONSTRAINT security\_list\_people\_fk FOREIGN KEY ( human\_id )

REFERENCES people ( human\_id );

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 8

-- CREATE INDEX 0

-- ALTER TABLE 16

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 0

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 0

-- CREATE MATERIALIZED VIEW 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 0

-- WARNINGS 0

**Приложение Б**

Пакет для обработки логина и пароля пользователя

Create Or Replace Package Pkg\_Security Is

Function Authenticate\_User(p\_User\_Name Varchar2

,p\_Password Varchar2) Return Boolean;

-----

Procedure Process\_Login(p\_User\_Name Varchar2

,p\_Password Varchar2

,p\_App\_Id Number);

End Pkg\_Security;

/

Create Or Replace Package Body Pkg\_Security Is

Function Authenticate\_User(p\_User\_Name Varchar2

,p\_Password Varchar2) Return Boolean As

v\_Password User\_Account.Password%Type;

v\_Active User\_Account.Active%Type;

Begin

If p\_User\_Name Is Null Or p\_Password Is Null Then

-- Write to Session, Notification must enter a username and password

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Please enter Username and password.');

Return False;

End If;

----

Begin

Select u.Active

,u.Password

Into v\_Active

,v\_Password

From User\_Account u

Where u.User\_Name = p\_User\_Name;

Exception

When No\_Data\_Found Then

-- Write to Session, User not found.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'User not found');

Return False;

End;

If v\_Password <> p\_Password Then

-- Write to Session, Password incorrect.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Password incorrect');

Return False;

End If;

If v\_Active <> 'Y' Then

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'User locked, please contact admin');

Return False;

End If;

---

-- Write user information to Session.

--

Apex\_Util.Set\_Session\_State('SESSION\_USER\_NAME'

,p\_User\_Name);

---

---

Return True;

End;

--------------------------------------

Procedure Process\_Login(p\_User\_Name Varchar2

,p\_Password Varchar2

,p\_App\_Id Number) As

v\_Result Boolean := False;

Begin

v\_Result := Authenticate\_User(p\_User\_Name

,p\_Password);

If v\_Result = True Then

-- Redirect to Page 1 (Home Page).

Wwv\_Flow\_Custom\_Auth\_Std.Post\_Login(p\_User\_Name -- p\_User\_Name

,p\_Password -- p\_Password

,v('APP\_SESSION') -- p\_Session\_Id

,p\_App\_Id || ':1' -- p\_Flow\_page

);

Else

-- Login Failure, redirect to page 101 (Login Page).

Owa\_Util.Redirect\_Url('f?p=&APP\_ID.:101:&SESSION.');

End If;

End;

End Pkg\_Security;

/

Представление, связывающее роли и List Entries

SET DEFINE OFF;

CREATE OR REPLACE VIEW ROLE\_LIST\_ENTRY ("USER\_ROLE", "LIST\_ENTRY\_ID") AS

select distinct p.user\_role, l.list\_entry\_id

from apex\_application\_list\_entries l

join role\_page p on l.entry\_target like 'f?p=&APP\_ID.:'||p.page||':&%SESSION%' or entry\_target is null

where l.list\_name = 'Desktop Navigation Menu'

and l.application\_id = 105;