Титульная часть

# ВВЕДЕНИЕ

ООО «Артсек» специализируется на разработке своих решений, в сфере безопасности, в спектр которых входят: видеонаблюдение, системы контроля доступа, системы распознавания государственных номеров.

Целью выпускной квалификационной работы является разработка распределённой мультифакторной системы контроля доступа, в функционал которой входит: автоматизированная работа, выдача пользователям пин-кодов, и активирование их на специальной пропускной панели.

В состав системы входят: программы для автоматизации работы системы, панели администратора для контролирования системы, WEB-сервисы для взаимодействия составляющих, по сети интернет, мобильное приложение для конечного пользователя, в функционал которой входит, авторизация, регистрация, и просмотр текущего пин-кода.

Для реализации программного комплекса, выбраны такие технологии как C#, .NET Framework, Xamarin для настольных и мобильных решений и PHP, Laravel для WEB-приложений.

# ОБЩАЯ ЧАСТЬ

* 1. Цель разработки

Целью разработки является обеспечение функционирования распределённой мультифакторной системы для контроля доступа на объекте, а также сокращение персонала обслуживающего программный комплекс.

* 1. Средства разработки
     1. Технические средства

Для разработки программного комплекса использовался персональный компьютер Xiaomi mi notebook air, обладающий характеристиками, представленными в Таблице 1.

Таблица 1 - Технические средства разработки

|  |  |  |
| --- | --- | --- |
| № | Тип оборудование | Наименование оборудование |
| 1 | 2 | 3 |
| Ноутбук Xiaomi Mi Notebook Air 13.3 | | |
| 1 | Размер экрана | 13.3” |
| 2 | Разрешение экрана | 1920x1080 |
| 3 | Линейка процессора | Intel Core i5 |
| 4 | Количество ядер процессора | 8 |
| 5 | Оперативная память | 8 ГБ |
| 6 | Тип видеокарты | дискретная |
| 7 | Видеокарта | NVIDIA GeForce MX250 |
| 8 | Конфигурация накопителей | SSD |
| 9 | Общий объем всех накопителей | 256 ГБ |
| 10 | Операционная система | Windows 10 Home |

* + 1. Программные средства

Для разработки программного комплекса использовались программные средства и среды, описанные в Таблице 2.

Таблица 2 - Программные средства разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип средства | Название средства | Назначение |
| 1 | 2 | 3 | 4 |
| 1 | Инструментальное средство разработки программных решений | Visual Studio 2019 Community 16.4.4 | Разработка программного комплекса для функционирования системы |
| 2 | Кроссплатформенная сборка веб-сервера | XAMPP 3.2.4 | Разработка WEB составляющих системы |
| 3 | Редактор кода | Atom 1.45.0 | Редактирование кода |
| 4 | Средство проектирования | Draw.io 12.9.4 | Разработка схем для проектирования приложения |
| 5 | Текстовый редактор | Microsoft Word 2016 16.0.4266.1001 | Разработка документации, формирование отчётных документов по шаблонам |

# СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

Разработать комплекс программного обеспечения для распределённой мультифакторной системы контроля доступа.

### Входные данные

В качестве входных данных используются данные из уже готовых баз данных ЛОМ и СКУД.

### Выходные данные

В качестве выходных данных, предоставляется информация об пользователях и прочих не относящихся к внутренней системы информации.

### Подробные требования к проекту

В данной системе контроля доступа должны быть реализованны следующие составляющие:

1. Консольное приложение на языке C#, для автоматизации системы;
2. Графическое приложение на языке C#, для конфигурирования настроек консольного приложения;
3. WEB-сервис на языке php, для безопасного взаимодействия с базой данных из сети, через другое программное обеспечение;
4. WEB-панель администратора, интерфейс понятный для человека, для взаимодействия с базой данных.
5. Мобильное приложение, для конечного пользователя;

## Внешняя спецификация

### Описание задачи

Необходимо разработать распределённую мультифакторную систему контроля доступа, включающее в себя следующие процессы: реализация автоматизированной работы, администрирование через WEB-панель, получение активного пин-кода через мобильное приложение.

### Входные и выходные данные

### Метод

При разработке системы контроля доступа были использованы методы объектно-ориентированного программирования, которые включают в себя:

1. Инкапсуляция - сокрытие реализации программных частей для безопасности. В данной предметной области используется только публичные методы и переменные, т.к. не требуют защиты.
2. Полиморфизм - возможность дополнять объект функционалом. Возможность выступать объекту в разных формах. Классический полиморфизм - замещение, переопределение методов. ad hoc полиформизм - перегрузка методов, поведение в зависимости от данных. Подмена объектов во время выполнения программы с одинаковыми методами через интерфейс.
3. Абстракция - В ООП это придание объекту характеристик, которые четко выделяет его на фоне остальных, определяя его концептуальные границы. Абстрагирование - В ООП это способ выделить набор значимых характеристик объекта, исключая из рассмотрения не значимые. Соответственно абстракция это набор таких характеристик.

При разработке WEB-сервиса использовался протокол авторизации Oauth 2, и Bearer способ авторизации.

### Тесты

Подробно сценарии тестов рассмотрены в приложении Б “Сценарии тестов и результатов тестовых испытаний”. На рисунке 1 представлена схема тестирования программного комплекса.



Рисунок 1 - Схема тестирования

### Контроль целостности данных

Все аномальные ситуации в программном комплексе перечислены в таблице 3.

Таблица 3 - Критические тесты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Ситуация | Аномалия | Реакция программы | Примечание |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Преждевременное нажатие клавиши продолжения | Программа не выполняет свою задачу до конца | Не все данные будут обработаны | Для обработки всей информации необходимо подождать пару секунд |

## Проектирование

### Схема архитектуры программы

На рисунке 2 подробно описана вся архитектуры системы программного комплекса.

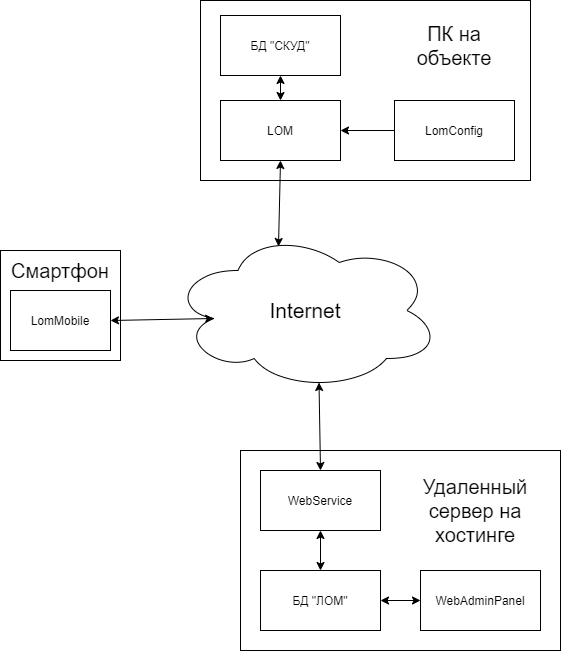


Рисунок 2 - Схема архитектуры программы

1. LOM – консольное приложение выполняет следующие задачи: генерация пин-кодов в базе данных “ЛОМ”, выгрузку пин-кодов из одной базы данных “ЛОМ” в базу данных “СКУД”, выполнение этих функций в определённое время по расписанию.
2. LomConfig – графическое приложение, для конфигурации настроек консольного приложения.
3. WebService – WEB-сервис, для безопасного взаимодействия с базой данных через сеть. В функционал входит, аутентификация, регистрация, авторизация и взаимодействие с таблицами в базе данных “LOM”.
4. WebAdminPanel – WEB-панель администратора, в функции которой входит: авторизация, редактирование пользователей, организаций, списка прав, просмотр уведомлений и журнала событий, генерация пин-кодов.
5. LomMobile – приложение для конечного пользователя, в функционал, которой входит: авторизация, регистрация, просмотр информации пользователя, проверка подключения к сети.

### Функциональная схема задачи

На рисунке 3 представлена общая функциональная схема, которая отражает возможности приложения с помощью описания всех функций имеющихся в программном комплексе.

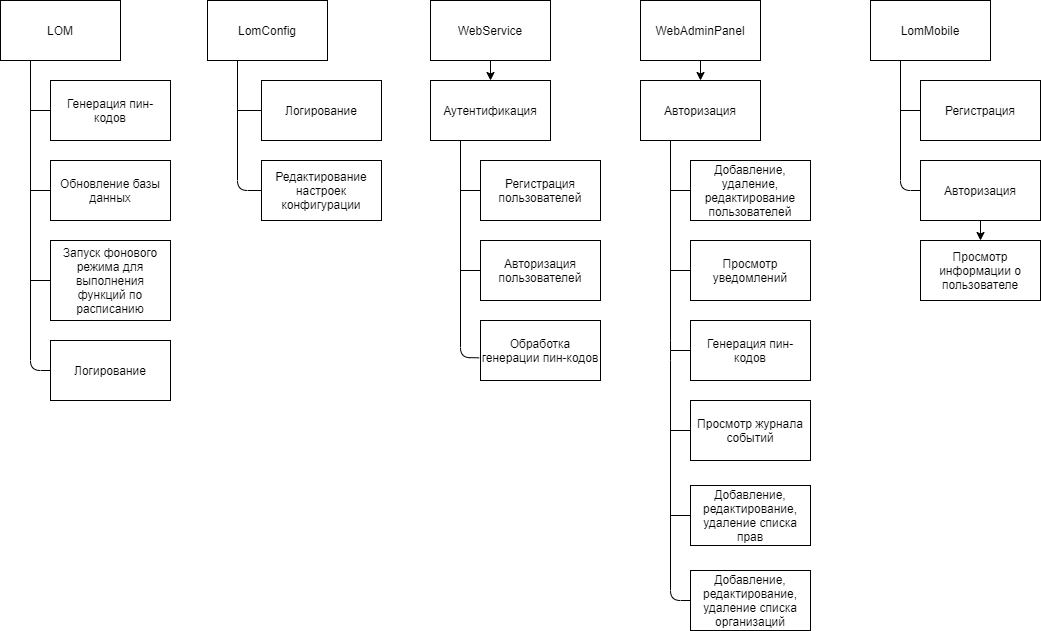


Рисунок 3 - Функциональная схема

### Структурная схема программы

На рисунке 4 показана структурная схема консольного приложения LOM, а в таблице 4 подробное описание, каждой компоненты.

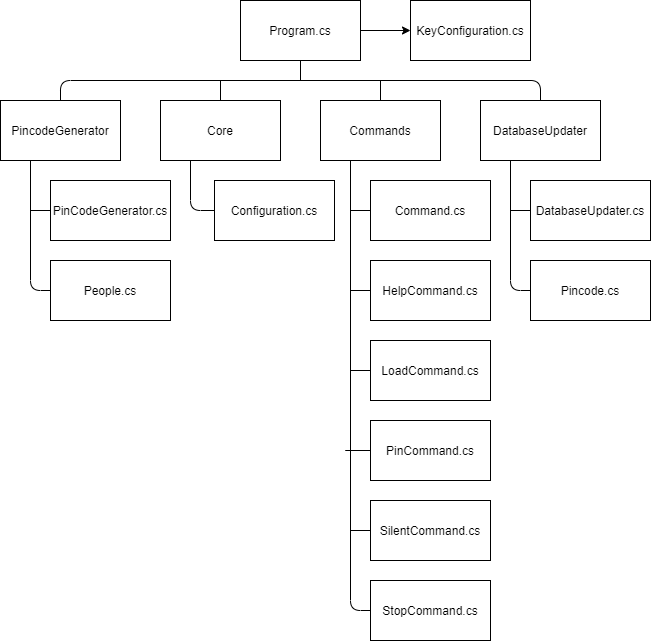


Рисунок 4 - Структурная схема LOM

Таблица 4 - Пояснение структурной схемы LOM

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | 2 | 3 |
| 1 | KeyConfiguration.cs | Класс хранящий в себе, информацию обо всех командах |
| 2 | Program.cs | Стартовая точка работы программы |
| 3 | PinCodeGenerator.cs | Логика, генерации пин-кодов |
| 4 | People.cs | Модель таблицы People |
| 5 | Configuration.cs | Статический класс, хранящий все настройки конфигурации |
| 6 | Command.cs | Базовый класс для команд |
| 7 | HelpCommand.cs | Класс для отображения справки |
| 8 | LoadCommand.cs | Команда вызова логики обновления базы данных |

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | 2 | 3 |
| 9 | PinCommand.cs | Команда вызова логики генерации пин-кодов |
| 10 | SilentCommand.cs | Вызов фонового процесса, для выполнения функций по расписанию |
| 11 | DatabaseUpdater.cs | Логика обновления базы данных |
| 12 | Pincode.cs | Модель таблицы Pincode |

На рисунке 5 показана структурная схема графического приложения LomConfig, а в таблице 5 подробное описание, каждой компоненты.

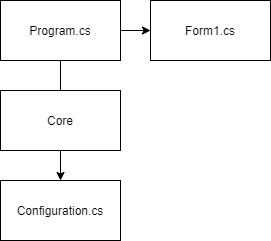


Рисунок 5 - Структурная схема LomConfig

Таблица 5 - Пояснительная структурной схемы LomConfig

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | 2 | 3 |
| 1 | Program.cs | Стартовая точка программы |
| 2 | Form1.cs | Главное окно игры |
| 3 | Configuration.cs | Статический класс, хранящий все настройки конфигурации |

На рисунке 6 показана структурная схема мобильного приложения LomMobile, а в таблице 6 подробное описание, каждой компоненты.

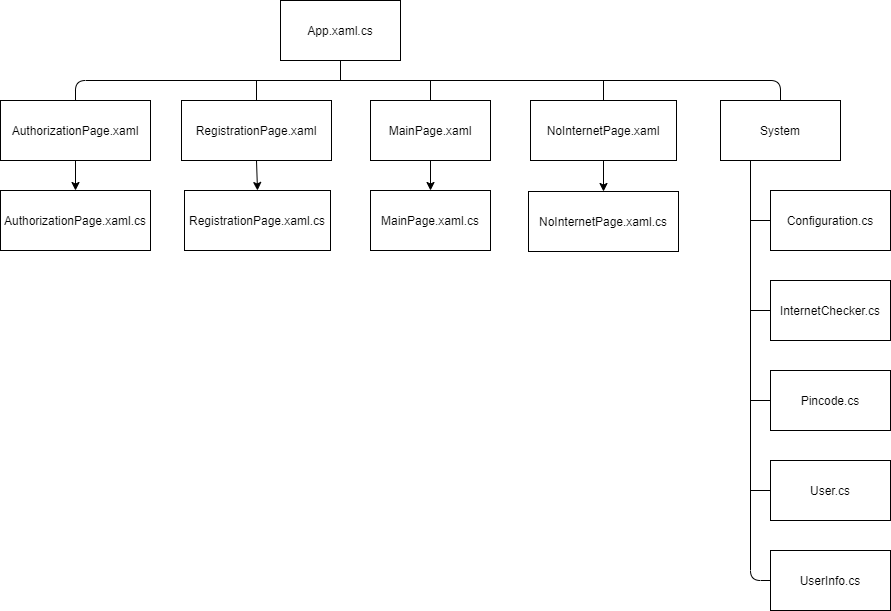


Рисунок 6 - Структурная схема LomMobile

Таблица 6 - Пояснение структурной схемы LomMobile

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | 2 | 3 |
| 1 | App.xaml.cs | Стартовая точка программы |
| 2 | AuthorizationPage.xaml.cs | Разметка страницы авторизации |
| 3 | AuthorizationPage.xaml | Логика авторизации |
| 4 | RegistrationPage.xaml.cs | Разметка страницы регистрации |
| 5 | RegistrationPage.xaml | Логика регистрации |
| 6 | MainPage.xaml.cs | Разметка страницы с информацией |
| 7 | MainPage.xaml | Логика страницы с информацией |
| 8 | NoInternetPage.xaml.cs | Разметка страницы отсутствия подключения |
| 9 | NoInternetPage.xaml | Логика страницы отсутствия подключения |
| 10 | Configuration.cs | Кофигурация приложения |
| 11 | InternetChecker.cs | Логика проверки подключения |
| 12 | Pincode.cs | Модель таблицы Pincode |
| 13 | User.cs | Модель таблицы User |
| 14 | UserInfo.cs | Модель таблицы UserInfo |

На рисунке 7 показана структурная схема WEB-панели администратора LomAdmin, а в таблице 7 подробное описание, каждой компоненты.

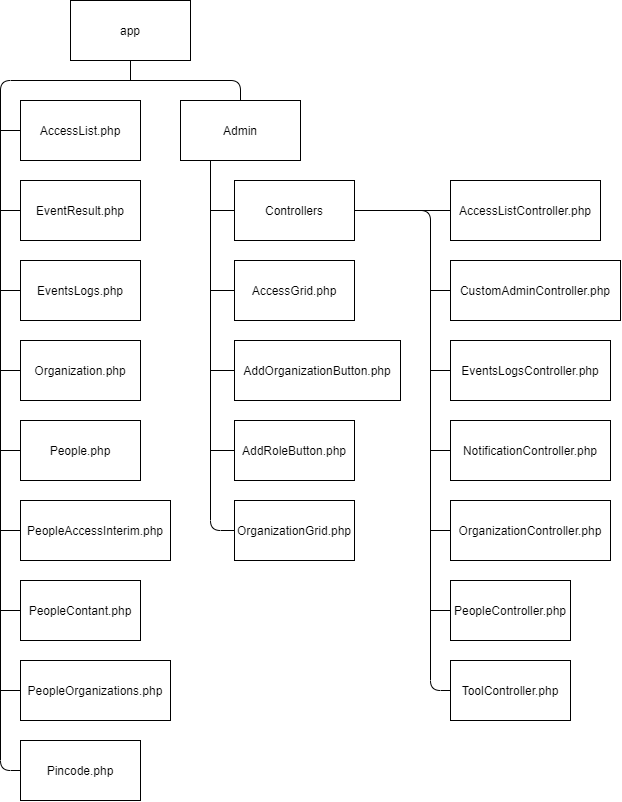


Рисунок 7 - Структурная схема LomAdmin

Таблица 7 – Пояснение структурной схемы LomAdmin

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | 2 | 3 |
| 1 | AccessList.php | Модель таблицы сущности AccessList |
| 2 | EventResult.php | Модель таблицы сущности EventResult |
| 3 | EventsLogs.php | Модель таблицы сущности EventsLogs |
| 4 | Organization.php | Модель таблицы сущности Organization |
| 5 | People.php | Модель таблицы сущности People |
| 6 | PeopleAccessInterim.php | Модель таблицы сущности PeopleAccesInterim |
| 7 | PeopleContact.php | Модель таблицы сущности PeopleContact |
| 8 | PeopleOrganization.php | Модель таблицы сущности PeopleOrganization |
| 9 | Pincode.php | Модель таблицы сущности Pincode |
| 10 | AccessGrid.php | Переопределённый класс, для таблицы списка прав |
| № | Название | Описание |
| 1 | 2 | 3 |
| 11 | AddOrganizationButton.php | Переопределённый класс, для кнопки добавления организации |
| 12 | AddRoleButton.php | Переопределённый класс, для кнопки добавления прав |
| 13 | OrganizationGrid.php | Переопределённый класс, для таблицы списка организаций |
| 14 | AccessListController.php | Контроллер для модели AccessList |
| 15 | CustomAdminController.php | Переопределённый класс, для изменённого контроллера |
| 16 | EventsLogsController.php | Контроллер для модели EventsLogs |
| 17 | NotificationController.php | Контроллер для модели Notification |
| 18 | OrganizationController.php | Контроллер для модели Organization |
| 19 | PeopleController.php | Контроллер для модели People |
| 20 | ToolController.php | Контроллер для модели Tool |

На рисунке 8 показана структурная схема графического WEB-сервиса LomApi, а в таблице 8 подробное описание, каждой компоненты.

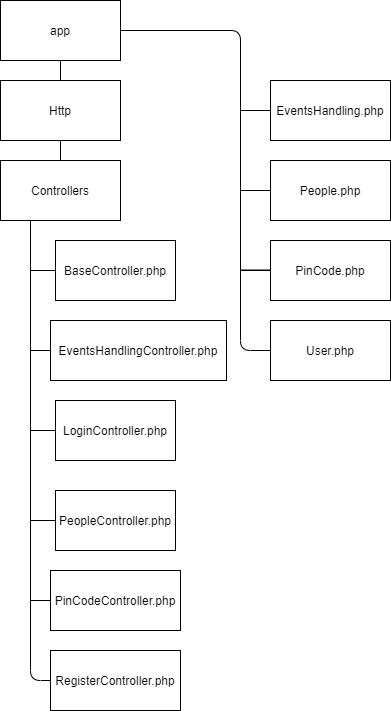


Рисунок 8 - Структурная схема LomApi

Таблица 8 - Пояснение структурной схемы LomApi

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | 2 | 3 |
| 1 | BaseController.php | Унаследованный контроллер с переопределённой логикой. |
| 2 | EventsHandlingController.php | Контроллер для обработки данных EventsHandling. |
| 3 | LoginController.php | Контроллер для авторизации. |
| 4 | PeopleController.php | Контроллер для обработки данных People. |
| 5 | PinCodeController.php | Контроллер для обработки данных PinCode. |
| 6 | RegisterController.php | Контроллер для регистрации. |
| 7 | EventsHandling.php | Модель для таблицы EventsHandling. |
| 8 | People.php | Модель для таблицы People. |
| 9 | PinCode.php | Модель для таблицы PinCode. |
| 10 | User.php | Модель для таблицы User. |

### Схема данных

ERD модель является описанием создаваемой базы данных, в которой представлены сущности базы данных (название таблиц) и атрибуты сущности (названия столбцов таблиц.), изображена на рисунке 9.

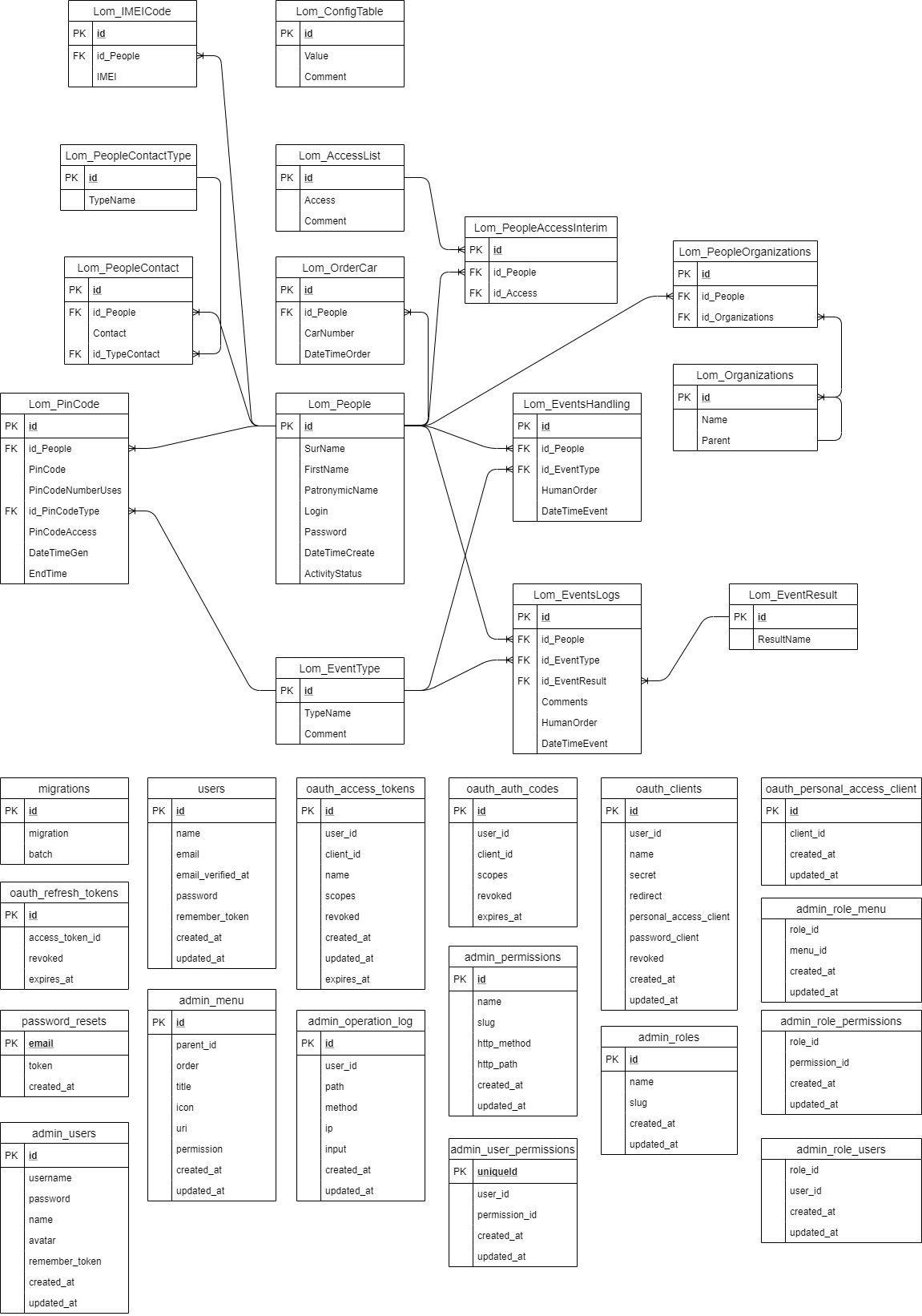


Рисунок 9 - ERD модель системы

Словарь данных описан в таблице 9.

Таблица 9 - Словарь данных ERD модели

| Наименование поля | Ключ | Тип данных | Описание |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Lom\_AccessList | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| Access |  | Varchar(254) | Указывается доступ, который будет привязан к определенному человеку в таблице “PeopleAccessInterim”. |
| Comment |  | Varchar(254) | Краткое описание доступа. |
| Lom\_ConfigTable | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| Value |  | Varchar(254) | Данное значение используется триггером для выполнения необходимых условий или задач. |
| Comment |  | Varchar(254) | Описывается для чего используется данное значение. |
| Lom\_EventResult | | | |
| id |  | Int(11) | Идентификационный номер |
| ResultName |  | Varchar(254) | Указываются, возможные варианты причин или невыполнения определённого события. |
| Lom\_EventsHandling | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Указывается пользователь, который вызвал событие. |
| id\_EventType | Внешний | Int(11) | Указывается как событие вызвал пользователь в приложении. |
| HumanOrder |  | Varchar(254) | Используется, если пользователь заказал ГРЗ в приложении. |
| DateTimeEvent |  | Datetime | Дата/время вызова события |
| Lom\_EventsLogs | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Заполняется триггером: EventsHandlingTr из таблицы “EventsHandling”, указывается пользователь, который вызвал событие. |
| id\_EventType | Внешний | Int(11) | Заполняется триггером: EventsHandlingTr из таблицы “EventsHandling”, указывается как событие вызвал пользователь в приложении. |
| id\_EventResult | Внешний | Int(11) | Заполняется триггером: EventsHandlingTr из таблицы “EventsHandling”, указывается будет ли выполнено или не выполнено определенное событие. |

| Наименование поля | Ключ | Тип данных | Описание |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Comments |  | Varchar(254) | Заполняется триггером: EventsHandlingTr из таблицы «EventsHandling», используется, если человек заказал ГРЗ в приложении. |
| HumanOrder |  | Varchar(254) | Заполняется триггером: EventsHandlingTr из таблицы «EventsHandling», используется, если человек заказал ГРЗ в приложении. |
| DateTimeEvent |  | datetime | Дата/время вызова события |
| Lom\_EventType | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| TypeName |  | Varchar(254) | Указывается тип события, который может вызвать пользователь в приложении. |
| Comment |  | Varchar(254) | Описание типа |
| Lom\_IMEICode | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Указывается пользователь, к которому привязан IMEI. |
| IMEI |  | Varchar(254) | Указывается IMEI код телефона пользователя. IMEI необходим для предотвращения кражи связки логин/пароль, а также вход в аккаунт с другого устройства. |
| Lom\_OrderCar | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Заполняется триггером EventsLogsTr из таблицы “EventsLogs”, указывается человек, который заказал ГРЗ. |
| CarNumber |  | Varchar(254) | Заполняется триггером: EventsLogsTr из таблицы «EventsLogs», указывается номер автомобиля |
| DateTimeOrder |  | Datetime | Дата/время заказа ГРЗ. |
| Lom\_Organizations | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| Name |  | Varchar(254) | Указывается название организации, к которой относятся работники. |
| Parent | Внешний | Int(11) | Указывается родительская организация |
| Lom\_People | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| SurName |  | Varchar(254) | Фамилия пользователя. |
| FirstName |  | Varchar(254) | Имя пользователя |
| PatronymicName |  | Varchar(254) | Отчество пользователя |
| Login |  | Varchar(100) | Логин пользователя |
| Наименование поля | Ключ | Тип данных | Описание |
| 1 | 2 | 3 | 4 |
| Password |  | Varchar(100) | Пароль пользователя |
| DateTimeCreate |  | Datetime | Дата/время создания аккаунта |
| ActivityStatus |  | Varchar(254) | Определяется активность аккаунта пользователя. Если аккаунт не будет активирован, тогда пользователь не сможет выполнять какие-либо действия в приложении |
| Lom\_PeopleAccessInterim | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Указывается человек к которому будет привязан определённый доступ из таблицы “AccessList”. |
| id\_Access | Внешний | Int(11) | Указывается уровень доступа из таблицы “AccessList”. |
| Lom\_PeopleContact | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Указывается пользователь, к которому будут привязаны контакты. |
| Contact |  | Varchar(254) | Указываются контакты человека, например, номер телефона, почта. |
| id\_TypeContact | Внешний | Int(11) | Указывается тип контакта из таблицы “PeopleContactType”. |
| Lom\_PeopleContactType | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| TypeName |  | Varchar(11) | Указывается тип контакта, который будет использоваться в таблице “PeopleContact”. |
| Lom\_PeopleOrganizations | | | |
| id | Первичный | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Указывается человек к которому привязывается организация, в которой работает. |
| id\_Organizationa | Внешний | Int(11) | Указывается уровень организации, в которой работает человек. |
| Lom\_PinCode | | | |
| id |  | Int(11) | Идентификационный номер |
| id\_People | Внешний | Int(11) | Заполняется триггером: EventsLogsTr из таблицы “EventsLogs”, указывается человек, который заказал генерациюю Пин-Кода. |
| PinCode |  | Varchar(10) | Заполняется триггером: EventsLogsTr. |
| PinСodeNumberUses |  | Int(11) | Количество проходов по одному пин-коду |
| id\_PincodeType | Внешний | Int(11) | Тип пин-кода, для чего создается. |

| Наименование поля | Ключ | Тип данных | Описание |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| PinCodeAccess |  | Varchar(100) | Права доступа человека, которые применяются к пин коду. |
| DateTimeGen |  | Datetime | Дата/время генерации Пин-кода. |
| EndTime |  | Datetime | Дата окончания Пин-Кода вычисляется триггером «EventsLogsTr», срок жизни ПинКода определяется в таблице «ConfigTable». |
| admin\_menu | | | |
| id | Первичный | Int(10) | Идентификационный номер |
| parent\_id |  | Int(11) | Ссылка на родительский узел |
| order |  | Int(11) | Назначение |
| title |  | Varchar(50) | Заголовок |
| icon |  | Varchar(50) | Иконка |
| permission |  | Varchar(255) | Узел прав |
| created\_at |  | timestamp | Дата создания |
| updated\_at |  | timestamp | Дата изменения |
| admin\_operation\_log | | | |
| id | Первичный | Int(10) | Идентификационный номер |
| user\_id |  | Int(11) | Id пользоватлея спровоцировавшего запись |
| path |  | Varchar(255) | Путь, который вызвал запись |
| method |  | Varchar(10) | Тип метода |
| ip |  | Varchar(255) | Ip адрес запроса |
| input |  | Text | Параметры запроса |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| admin\_permissions | | | |
| id | Первичный | Int(10) | Идентификационный номер |
| name |  | Varchar(50) | Наименование узла |
| slug |  | Varchar(50) | Слаг |
| http\_method |  | Varchar(255) | Метод запроса |
| http\_path |  | Text | Путь запроса |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestanp | Дата изменения |
| admin\_roles | | | |
| id | Первичный | Int(10) | Идентификационный номер |
| name |  | Varchar(50) | Наименование |
| slug |  | Varchar(50) | Слаг |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| admin\_role\_menu | | | |
| role\_id | Первичный | Int(11) | Id роли |
| menu\_id |  | Int(11) | Id меню |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| admin\_role\_permissions | | | |
| role\_id | Первичный | Int(11) | Id роли |
| Наименование поля | Ключ | Тип данных | Описание |
| 1 | 2 | 3 | 4 |
| permission\_id |  | Int(11) | Id прав |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| admin\_role\_users | | | |
| role\_id | Первичный | Int(11) | Id роли |
| User\_id |  | Int(11) | Id пользователя |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| admin\_users | | | |
| id | Первичный | Int(10) | Идентификационный номер |
| username |  | Varchar(190) | Имя пользователя |
| password |  | Varchar(60) | Пароль |
| name |  | Varchar(255) | Имя |
| avatar |  | Varchar(255) | Аватар |
| remember\_token |  | Varchar(100) | Токен, при утере пароля |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| admin\_user\_permissions | | | |
| user\_id |  | Int(11) | Id пользователя |
| permission\_id |  | Int(11) | Id прав |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamo | Дата обновления |
| migrations | | | |
| id | Первичный | Int(10) | Идентификационный номер |
| migration |  | Varchar(255) | Наименование миграции |
| batch |  | Int(11) | Наименование группы |
| oauth\_access\_tokens | | | |
| id | Первичный | Varchar(100) | Идентификационный номер |
| user\_id |  | Bigint(20) | Id пользователя |
| client\_id |  | Bigint(20) | Id клиента |
| name |  | Varchar(255) | Наименование |
| scopes |  | Text | Цель токена |
| revokes |  | Tinyint(1) | Отмена |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| expires\_at |  | Datetime | Дата истечения |
| oauth\_auth\_codes | | | |
| id | Первичный | Varchar(100) | Идентификационный номер |
| user\_id |  | Bigint(20) | Id пользователя |
| client\_id |  | Bigint(20) | Id клиента |
| scopes |  | Text | Цель кода |
| revoked |  | Tinyint(1) | Отмена |
| expires\_at |  | Datetime | Дата истчения |
| oauth\_clients | | | |
| id | Первичный | Bigint(20) | Идентификационный номер |
| user\_id |  | Bigint(20) | Id пользователя |
| name |  | Varchar(255) | Наименование |
| secret |  | Varchar(100) | Секретный ключ |
| Наименование поля | Ключ | Тип данных | Описание |
| 1 | 2 | 3 | 4 |
| redirect |  | Text | Переадресация |
| personal\_access\_client |  | Tinyint(1) | Персональный доступ клиента |
| password\_client |  | Tinyint(1) | Пароль клиента |
| revoked |  | Tinyint(1) | Отмена |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата изменения |
| oauth\_personal\_access\_cleints | | | |
| id | Первичный | Bigint(20) | Идентификационный номер |
| client\_id |  | Bigint(20) | Id клиента |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата обновления |
| oauth\_refresh\_tokens | | | |
| id | Первичный | Varchar(100) | Идентификационный номер |
| access\_token\_id |  | Varchar(100) | Id токена доступа |
| revoked |  | Tinyint(1) | Отмена |
| expires\_at |  | Datetime | Дата истчения |
| password\_resets | | | |
| email | Первичный | Varchar(255) | Почта для востановления |
| token |  | Varchar(255) | Токен востановления |
| created\_at |  | Timestamp | Дата создания |
| users | | | |
| id | Первичный | Bigint(20) | Идентификационный номер |
| name |  | Varchar(255) | Имя пользователя |
| email |  | Varchar(255) | Почта |
| email\_verified\_at |  | Timestamp | Дата подтвердения почты |
| password |  | Varchar(255) | Пароль |
| remember\_token |  | Varchar(100) | Токендля востановления |
| created\_at |  | Timestamp | Дата создания |
| updated\_at |  | Timestamp | Дата обновления |

### Схема пользовательского интерфейса

Схема графического приложения LomConfig представлена на рисунке 10. Приложение состоит из одного окна.



Рисунок 10 - Схема приложения LomConfig

Схема WEB-панели администратора показана на рисунке 11.

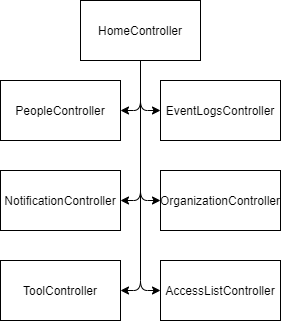


Рисунок 11 - Схема WEB-панели администратора

Схема мобильного приложения показана на рисунке 12.

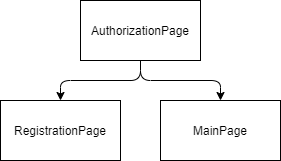


Рисунок 12 - Схема мобильного приложения

### Укрупненный алгоритм

В программном комплексе, в консольном приложение LOM, представлен укрупненный алгоритм, по преобразованию пин-кода в другую форму. Блок схема алгоритма представлена на рисунке 13.

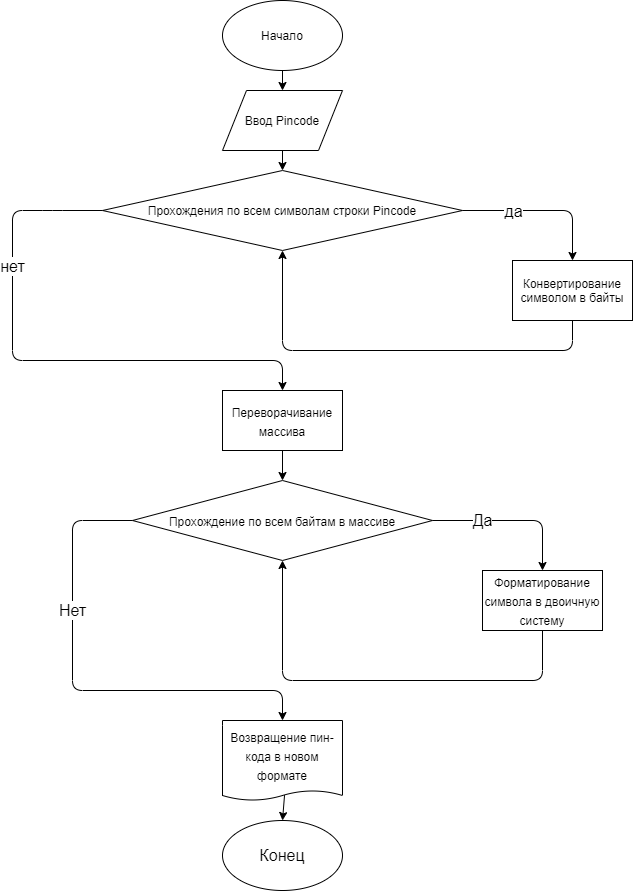


Рисунок 13 - Укрупненный алгоритм

### Алгоритм

На рисунке 14 представлена блок-схема того же алгоритма показанного на рисунке 13, но его подробное описание.

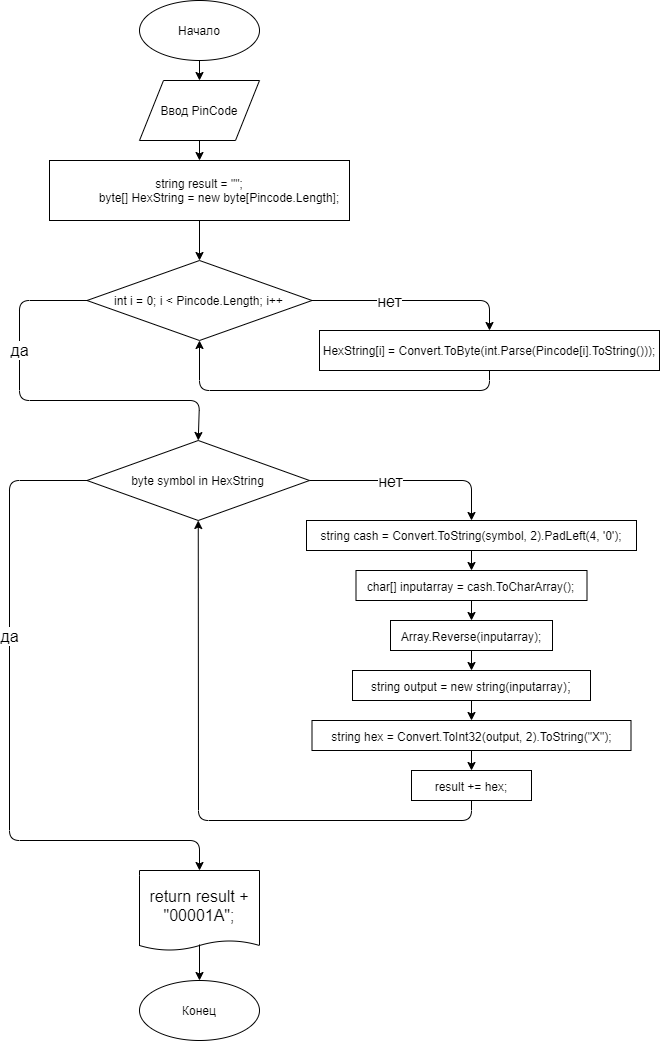


Рисунок 14 - Алгоритм формирования пин-кода

## Результаты работы программы

В результате разработки проекта, были получены и закреплены знания об ООП, об построение WEB-сервисов и протокола HTTP.

На выходе была разработана распределённая мультифакторная система контроля доступа, цель которой была автоматизировать процесс контроля доступа.

Была произведена оценка результатов работы программ, что позволяет сделать вывод о целесообразности и эффективности использования вычислительных средств в рамках предприятия ООО «Артсек».

На рисунке 15 представлена работа консольного приложения.

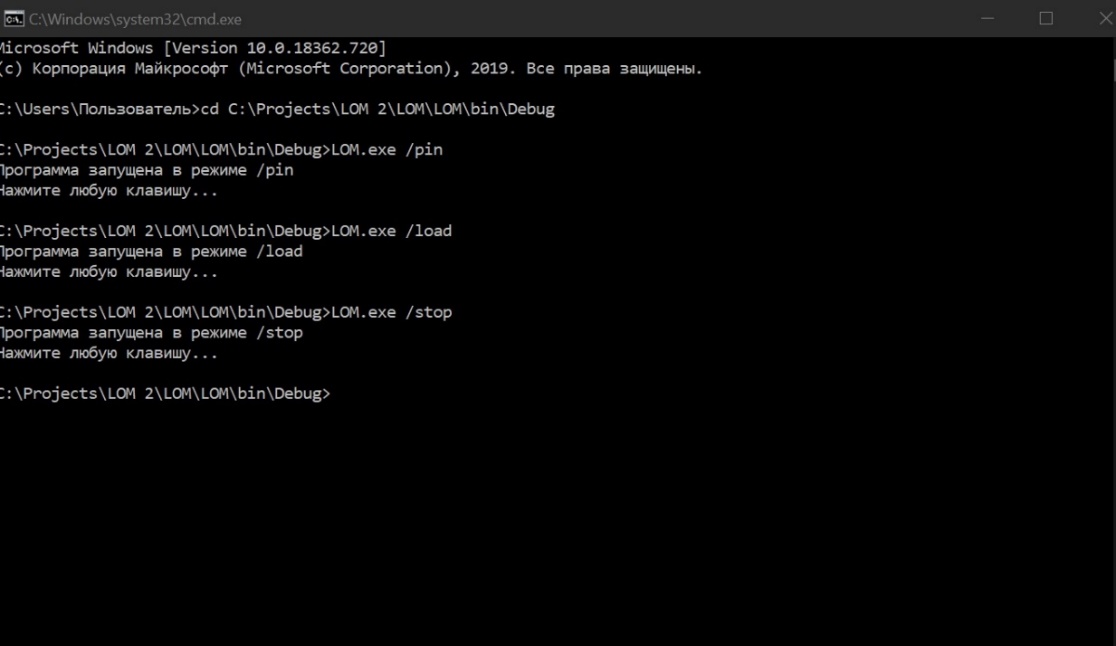


Рисунок 15 - Главное окно консольного приложения

На рисунке 16 представлена работа приложения конфигурирования.

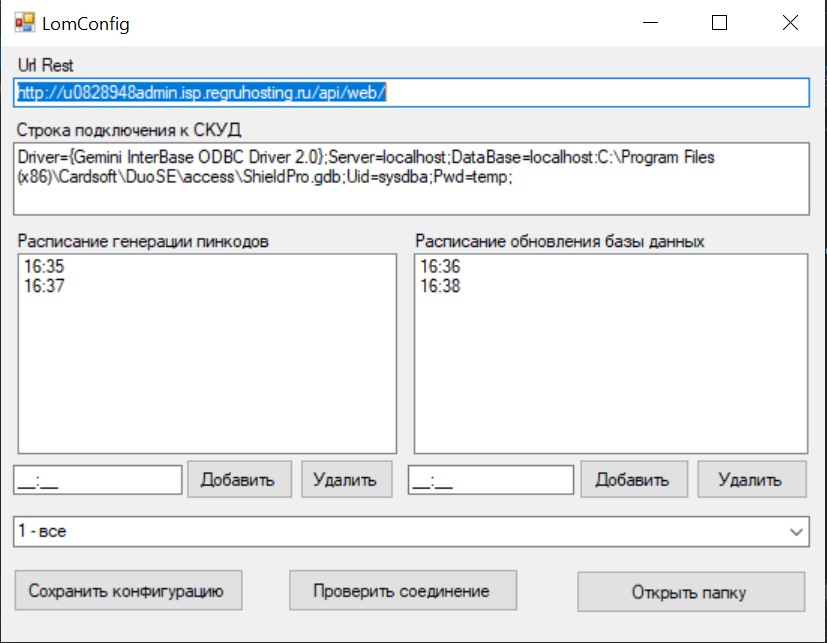


Рисунок 16 - Главное окно приложения конфигурирования

На рисунке 17 представлена работа WEB-панели администратора.

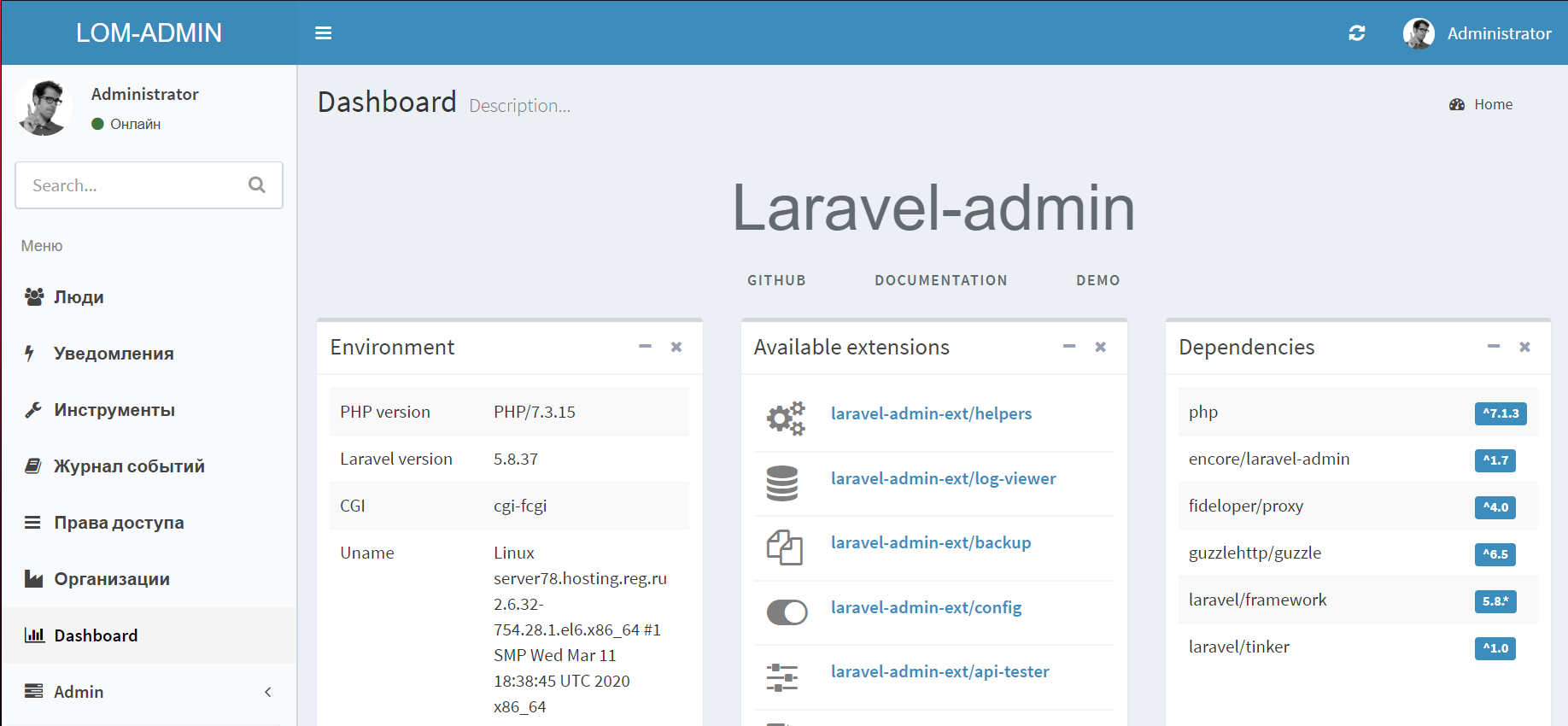


Рисунок 17 - Домашняя страница WEB-панели администратора

На рисунке 18 представлена работа авторизации мобильного приложения.

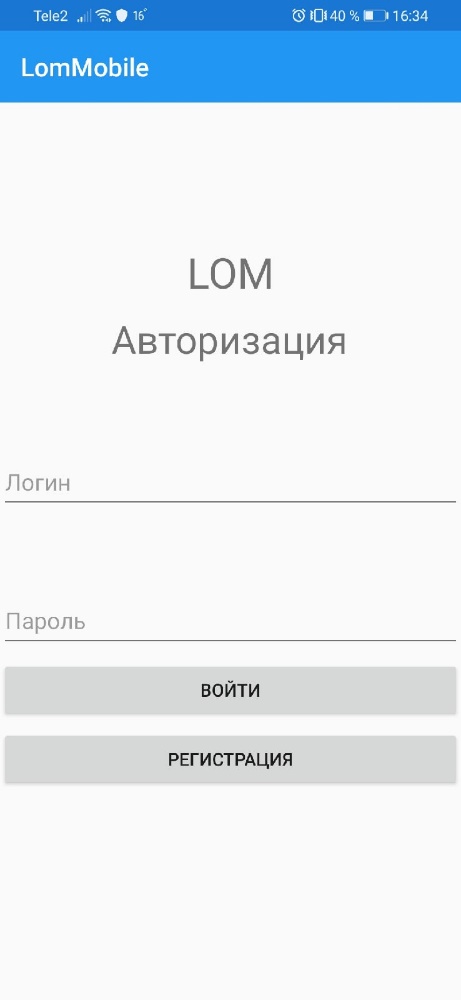


Рисунок 18 - Авторизация мобильного приложения

На рисунке 19 представлен профиль авторизированного пользователя.



Рисунок 19 - Просмотр информации пользователя

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## Инструментальные средства разработки

Для разработки консольного и графического приложения, были выбраны IDE Visual Studio 2019 и .NET Framework. Также были использованы внешние библиотеки, такие Newtonsoft.JSON для парсинга JSON и IniParser, для работы с конфигурационными файлами Ini.

Для WEB составляющих был применён язык программирования PHP с фреймворком Laravel, с расширением Laravel-admin и протокола авторизации Oauth 2.

Для мобильного приложения использовалась технология Xamarin, на базе .NET Framework и языка C#.

Для оформления документации, был использован Word и draw.io.

## Отладка программы

В процессе разработки встречались моменты, когда было необходимо производить отладку. В качестве инструмента, для отладки была выбран встроенные отладчик в Visual Studio 2019. Пример процесса отладки представлен на рисунке 20.

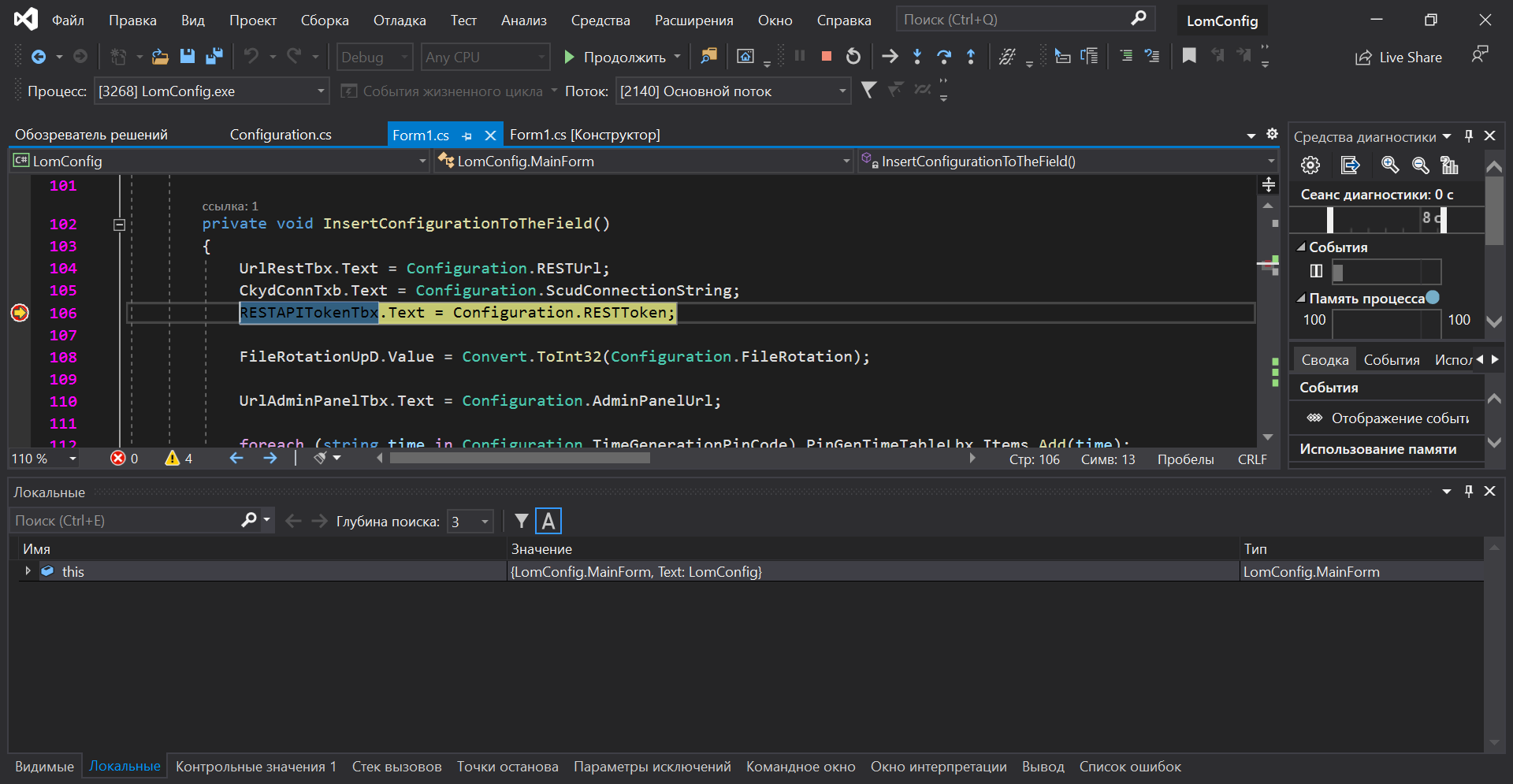


Рисунок 20 - Процесс отладки

## Защитное программирование

В процессе разработки для корректного ввода данных времени, была сделана проверка корректности формата данных через исключения.

try

{

DateTime Time = Convert.ToDateTime(DBUpdaterMTbx.Text);

DBUpdateTimeTableLbx.Items.Add(Time.ToString("H:mm"));

CheckChanges();

}

catch(Exception ex)

{

MessageBox.Show("Неверный формат данных!");

}

Также в мобильном приложение предусмотрены методы проверки валидации при регистрации и авторизации.

bool SurName = true, FirstName = true, PatronymicName = true, Login = true, Password = true, RepeatPassword = true;

if(SurNameEntr.Text == null || SurNameEntr.Text == "")

{

SurNameErrorLbl.Text = "Введите фамилию";

SurName = false;

}

else if (SurNameEntr.Text == null || SurNameEntr.Text.Length < 3)

{

SurNameErrorLbl.Text = "Фамилия должна быть длинее 3-ёх символов";

SurName = false;

}

if (FirstNameEntr.Text == null || FirstNameEntr.Text == "")

{

FirstNameErrorLbl.Text = "Введите имя";

FirstName = false;

}

else if (FirstNameEntr.Text == null || FirstNameEntr.Text.Length < 3)

{

FirstNameErrorLbl.Text = "Имя должно быть длинее 2-ух символов";

FirstName = false;

}

if(LoginEntr.Text == null || LoginEntr.Text == "")

{

LoginErrorLbl.Text = "Введите логин";

Login = false;

}

else if (LoginEntr.Text == null || LoginEntr.Text.Length < 6)

{

LoginErrorLbl.Text = "Логин должен быть длинее 6-ти символов";

Login = false;

}

if (PasswordEntr.Text == null || PasswordEntr.Text == "")

{

PasswordErrorLbl.Text = "Введите пароль";

Password = false;

}

else if (PasswordEntr.Text == null || PasswordEntr.Text.Length < 6)

{

PasswordErrorLbl.Text = "Пароль должен быть длинее 6-ти символов";

Password = false;

}

if (RepeatPasswordEntr.Text == null || PasswordEntr.Text != RepeatPasswordEntr.Text)

{

RepeatPasswordErrorLbl.Text = "Пароли не совпадают";

RepeatPassword = false;

}

if (SurName && FirstName && PatronymicName && Login && Password && RepeatPassword) return true;

else return false;

## Характеристика программы

В процессе разработки приложения использовались встроенные классы .NET Framework, внешние библиотеки Newtonsoft.JSON, IniParser и собственная библиотека IziLog. Для разработки WEB-составляющих применялся Laravel, Laravel-admin, Laravel-passport, библиотека GuzzleHttp. Для мобильного приложения технология Xamarin.

Более подробное описание программных модулей описаны в приложение Б Исходный текст программы.

# ЗАКЛЮЧЕНИЕ

В изложенной дипломной работе была разработана распределённая мультифакторная система контроля доступа цель использования, которой автоматизирование процесса контроллирования входящих лиц в объект, в состав которой входят, прикладные программы для автоматизации, WEB-панели администратора, WEB-сервис, а также приложение для конечного пользователя.

В процессе проектирования были получены навыки анализа необходимой предметной области, закреплены навыки составления логической модели, функциональной схемы, структурной схемы, диаграммы бизнес-процессов, диаграммы классов, схемы тестирования, схемы интерфейса. Были улучшены и закреплены знания ООП, и знания языков программирования C# и PHP.

Были проведены оценки работы программного комплекса и сделаны выводы, о целесообразности применения программного продукта.

# СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

1. Полное руководство по языку программирования С# 8.0 и платформе .NET Core 3 [Электронный ресурс]: Рабочая программа / [сайт] https://metanit.com/sharp/tutorial/
2. Руководство по языку C# [Электронный ресурс]: Рабочая программа / [сайт] <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
3. Руководство по C# [Электронный ресурс]: Рабочая программа / [сайт] <https://professorweb.ru/my/csharp/charp_theory/level1/index.php>
4. Руководство по PHP [Электронный ресурс]: Рабочая программа / [сайт] <https://www.php.net>
5. Руководство по Laravel [Электронный ресурс]: Рабочая программа / [сайт] <https://laravel.com/docs/7.x>
6. Руководство по Laravel-Admin [Электронный ресурс]: Рабочая программа / [сайт] <https://laravel-admin.org/docs/#/>
7. Инструкция по Laravel-Passport [Электронный ресурс]: Рабочая программа / [сайт] <https://laravel.demiart.ru/create-rest-api-with-authentication/>
8. Роберт Мартин: Чистый код: создание, анализ и рефакторинг. Библиотека программиста / Мартин Р. – 1 изд. Питер СПБ – 464 с.
9. Эндрю Троелсен: Язык программирования C# 7 и платформы .NET и .NET Core / Троелсен Эн. – 1 изд. Вильямс – 1328 с.
10. Рихтер Джеффри: CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Джеффри Р. – 4 изд. Питер – 896 с.
11. Стив Макконнелл: Совершенный код. Мастер-класс / Макконнелл С. – 1 изд. БХВ-Петербург – 896 с.
12. Албахари Бен: C# 7.0. Справочник. Полное описание языка / Бен А. – 1 изд. O'Reilly – 1024 с.
13. Альтхофф Кори: Сам себе программист. Как научиться программировать и устроиться в Ebay? / Кори А. – 1 изд. Бомбора – 208 с.
14. Стивенс Род: Алгоритмы. Теория и практическое применение / Род С. – 1 изд. Эксмо – 544 с.
15. Кевин Янк: PHP и MySQL. От новичка к профессионалу / Янк К. – 1 изд. Эксмо – 381 с.
16. Игорь Симдянов: PHP 7 / Симдянов И. – 1 изд. БХВ-Петербург – 1071 с.
17. Джош Локхарт: Современный PHP. Новые возможности и передовой опыт / Локхарт Д. – 1 изд. ДМК Пресс – 306 с.
18. Максим Кузнецов: PHP на примерах / Кузнецов М. – 1 изд. БХВ-Петербург – 398 с.
19. Веллинг Люк: Разработка веб-приложений с помощью PHP и MySQL / Люк В. – 1 изд. Диалектика – 768 с.
20. Мэтт Стаффер: Laravel. Полное руководство / Стаффер М. – 1 изд. Питер – 512 с.

# ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ

АННОТАЦИЯ

В приложении Текст программы продемонстрирован код приложения всех модулей.

В разделе Модули программы представлена таблица всех модулей системы с описание и размером каждого модуля.

В разделе Текст программы и модулей представлен код каждого модуля системы.

1. Модули программы

Таблица 1 - Модули программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название | Описание | Кол-во строк | Вес |
| 1 | 2 | 3 | 4 | 5 |
| Консольное приложение LOM | | | | |
| 1 | KeyConfiguration.cs | Класс хранящий в себе, информацию обо всех командах | 17 | 1 КБ |
| 2 | Program.cs | Стартовая точка работы программы | 169 | 7 КБ |
| 3 | PinCodeGenerator.cs | Логика, генерации пин-кодов | 85 | 4 КБ |
| 4 | People.cs | Модель таблицы People | 16 | 1 КБ |
| 5 | Configuration.cs | Статический класс, хранящий все настройки конфигурации | 39 | 3 КБ |
| 6 | Command.cs | Базовый класс для комманд | 17 | 1 КБ |
| 7 | HelpCommand.cs | Класс для отображения справки | 20 | 1 КБ |
| 8 | LoadCommand.cs | Команда вызова логики обновления базы данных | 19 | 1 КБ |
| 9 | PinCommand.cs | Команда вызова логики генерации пин-кодов | 18 | 1 КБ |
| 10 | SilentCommand.cs | Вызов фонового процесса, для выполнения функций по расписанию | 205 | 7 КБ |
| 11 | DatabaseUpdater.cs | Логика обновления базы данных | 173 | 7 КБ |
| 12 | Pincode.cs | Модель таблицы Pincode | 12 | 1 КБ |
| Графическое приложение LomConfig | | | | |
| 13 | Program.cs | Стартовая точка программы | 22 | 1 КБ |
| 14 | Form1.cs | Главное окно игры | 489 | 21 КБ |
| 15 | Configuration.cs | Статический класс, хранящий все настройки конфигурации | 39 | 3 КБ |
| Мобильное приложение | | | | |
| 16 | App.xaml.cs | Стартовая точка программы | 28 | 1 КБ |
| 17 | AuthorizationPage.xaml | Разметка страницы авторизации | 59 | 3 КБ |
| 18 | AuthorizationPage.xaml.cs | Логика авторизации | 170 | 6 КБ |
| № | Название | Описание | Кол-во строк | Вес |
| 1 | 2 | 3 | 4 | 5 |
| 19 | RegistrationPage.xaml | Разметка страницы регистрации | 84 | 4 КБ |
| 20 | RegistrationPage.xaml.cs | Логика регистрации | 167 | 1 КБ |
| 21 | MainPage.xaml | Разметка страницы с информацией | 43 | 2 КБ |
| 22 | MainPage.xaml.cs | Логика страницы с информацией | 104 | 4 КБ |
| 23 | NoInternetPage.xaml | Разметка страницы отсутствия подключения | 15 | 1 КБ |
| 24 | NoInternetPage.xaml.cs | Логика страницы отсутствия подключения | 20 | 1 КБ |
| 25 | Configuration.cs | Кофигурация приложения | 10 | 2 КБ |
| 26 | InternetChecker.cs | Логика проверки подключения | 36 | 1 КБ |
| 27 | Pincode.cs | Модель таблицы Pincode | 15 | 1 КБ |
| 28 | User.cs | Модель таблицы User | 12 | 1 КБ |
| 29 | UserInfo.cs | Модель таблицы UserInfo | 16 | 1 КБ |
| WEB-панель администратора | | | | |
| 30 | AccessList.php | Модель таблицы сущности AccessList | 20 | 1 КБ |
| 31 | EventResult.php | Модель таблицы сущности EventResult | 20 | 1 КБ |
| 32 | EventsLogs.php | Модель таблицы сущности EventsLogs | 20 | 1 КБ |
| 33 | Organization.php | Модель таблицы сущности Organization | 20 | 1 КБ |
| 34 | People.php | Модель таблицы сущности People | 20 | 1 КБ |
| 35 | PeopleAccessInterim.php | Модель таблицы сущности PeopleAccesInterim | 20 | 1 КБ |
| 36 | PeopleContact.php | Модель таблицы сущности PeopleContact | 20 | 1 КБ |
| 37 | PeopleOrganization.php | Модель таблицы сущности PeopleOrganization | 20 | 1 КБ |
| 38 | Pincode.php | Модель таблицы сущности Pincode | 20 | 1 КБ |
| 39 | AccessGrid.php | Переопределённый класс, для таблицы списка прав | 40 | 1 КБ |
| 40 | AddOrganizationButton.php | Переопределённый класс, для кнопки добавления организации | 34 | 1 КБ |
| 41 | AddRoleButton.php | Переопределённый класс, для кнопки добавления прав | 34 | 1 КБ |
| 42 | OrganizationGrid.php | Переопределённый класс, для таблицы списка организаций | 40 | 1 КБ |
| 43 | AccessListController.php | Контроллер для модели AccessList | 69 | 2 КБ |
| 44 | CustomAdminController.php | Переопределённый класс, для изменённого контроллера | 20 | 1 КБ |
| 45 | EventsLogsController.php | Контроллер для модели EventsLogs | 95 | 3 КБ |
| 46 | NotificationController.php | Контроллер для модели Notification | 94 | 3 КБ |
| 47 | OrganizationController.php | Контроллер для модели Organization | 94 | 3 КБ |
| 48 | PeopleController.php | Контроллер для модели People | 351 | 12 КБ |
| № | Название | Описание | Кол-во строк | Вес |
| 1 | 2 | 3 | 4 | 5 |
| 49 | ToolController.php | Контроллер для модели Tool | 62 | 4 КБ |
| WEB-сервис | | | | |
| 50 | BaseController.php | Унаследованный контроллер с переопределённой логикой. | 37 | 1 КБ |
| 51 | EventsHandlingController.php | Контроллер для обработки данных EventsHandling. | 27 | 1 КБ |
| 52 | LoginController.php | Контроллер для авторизации. | 32 | 1 КБ |
| 53 | PeopleController.php | Контроллер для обработки данных People. | 39 | 1 КБ |
| 54 | PinCodeController.php | Контроллер для обработки данных PinCode. | 24 | 1 КБ |
| 55 | RegisterController.php | Контроллер для регистрации. | 38 | 2 КБ |
| 56 | EventsHandling.php | Модель для таблицы EventsHandling. | 29 | 1 КБ |
| 57 | People.php | Модель для таблицы People. | 38 | 1 КБ |
| 58 | PinCode.php | Модель для таблицы PinCode. | 20 | 1 КБ |
| 59 | User.php | Модель для таблицы User. | 40 | 1 КБ |

2. Текст программы и модулей

2.1. KeyConfiguration.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LOM

{

public static class KeyConfiguration

{

public static Command[] List = new Command[] { new SilentCommand(),

new PinCommand(),

new LoadCommand(),

new StopCommand(),

new HelpCommand()};

}

}

2.2. Program.cs

using System;

using System.Diagnostics;

using System.IO;

using System.Text;

using IniParser;

using IniParser.Model;

using IziLog;

using IziLog.Records;

namespace LOM

{

class Program

{

static void Main(string[] args)

{

try

{

IziLog.Configuration.PathToLogFile = Configuration.DefaultDirectoryPathToLog;

IziLog.Configuration.FileRotation = Convert.ToInt32(Configuration.FileRotation);

LoggerServer.Start();

if (IsFirstLaunch()) FirstStartConfiguration();

else GetConfigurationFromFile();

string Mode = GetApplicationMode(args);

string KeyString = args[0].ToLower();

ExecuteApplication(KeyString);

string Name = Process.GetCurrentProcess().ProcessName;

Command CurrentCommand = GetCommandObject(KeyString);

CurrentCommand.Run();

Console.WriteLine("Нажмите любую клавишу...");

Console.ReadKey();

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

}

private static string GetApplicationMode(string[] args)

{

switch (args.Length)

{

case 0:

{

Console.WriteLine("/help - вызов справки\n/silent - запуск программы в штатном режиме\n/pin - генерация пин-кодв\n/load - Выгрузка базы данных\n/stop - остановка LOM");

throw new Exception("Программа запущена без использования ключей, остановка программы");

break;

}

case 1:

{

return args[0];

}

default:

{

throw new Exception("Введено недопустимое кол-во ключей");

break;

}

}

return "";

}

private static void ExecuteApplication(string Key)

{

if (IsKeyExist(Key)) RunApplicationInCurrentMode(Key);

}

private static bool IsKeyExist(string Key)

{

foreach(Command CommandObject in KeyConfiguration.List)

{

if (CommandObject.CommandName.Equals(Key)) return true;

}

throw new Exception(Key + " - такого ключа не существует!");

}

private static void RunApplicationInCurrentMode(string Mode)

{

Console.WriteLine("Программа запущена в режиме " + Mode);

}

private static Command GetCommandObject(string Key)

{

foreach(Command CommandObject in KeyConfiguration.List)

{

if (CommandObject.CommandName == Key) return CommandObject;

}

throw new NullReferenceException();

}

private static bool IsFirstLaunch()

{

if (!Directory.Exists(Configuration.DefaultDirectoryPath)) return true;

return false;

}

private static void FirstStartConfiguration()

{

Directory.CreateDirectory(Configuration.DefaultDirectoryPath);

using (FileStream fstream = new FileStream(Configuration.DefaultDirectoryPathToIni, FileMode.OpenOrCreate))

{

string contains = "[Main]\n" +

"UrlREST = " + Configuration.RESTUrl + "\n" +

"RESTToken = " + Configuration.RESTToken + "\n" +

"FileRotation = " + Configuration.FileRotation + "\n" +

"ScudConnectionString = " + Configuration.ScudConnectionString + "\n" +

"AdminPanelUrl = " + Configuration.AdminPanelUrl + "\n" +

"TimeGenerationPincode = " + FromArrayToString(Configuration.TimeGenerationPinCode) + "\n" +

"TimeUpdateDatabase = " + FromArrayToString(Configuration.TimeUpdateDatabase) + "\n" +

"ParentOrg = " + Configuration.ParentOrg;

byte[] array = Encoding.Default.GetBytes(contains);

fstream.Write(array, 0, array.Length);

}

Logger.Log(new InfoRecord("Процесс создания домашней директории и файла конфигурации, завершен успешно"));

GetConfigurationFromFile();

}

public static void GetConfigurationFromFile()

{

Logger.Log(new InfoRecord("Считывание конфигурации..."));

FileIniDataParser IniParser = new FileIniDataParser();

IniData IniData = IniParser.ReadFile(Configuration.DefaultDirectoryPathToIni);

Configuration.RESTUrl = IniData["Main"]["UrlREST"];

Configuration.RESTToken = IniData["Main"]["RESTToken"];

Configuration.FileRotation = IniData["Main"]["FileRotation"];

Configuration.ScudConnectionString = IniData["Main"]["ScudConnectionString"];

Configuration.AdminPanelUrl = IniData["Main"]["AdminPanelUrl"];

Configuration.TimeGenerationPinCode = FromStringToArray(IniData["Main"]["TimeGenerationPincode"]);

Configuration.TimeUpdateDatabase = FromStringToArray(IniData["Main"]["TimeUpdateDatabase"]);

Configuration.ParentOrg = IniData["Main"]["ParentOrg"];

Logger.Log(new InfoRecord("Считывание конфигурации, прошло успешно"));

}

private static string FromArrayToString(string[] TimeTable)

{

string FinalString = "";

foreach (string Time in TimeTable) FinalString += Time + " ";

return FinalString;

}

private static string[] FromStringToArray(string Times)

{

return Times.Split(' ');

}

}

}

2.3. PinCodeGenerator.cs

using System;

using System.Collections.Generic;

using System.Net.Http;

using System.Threading.Tasks;

using Newtonsoft.Json.Linq;

using IziLog;

using IziLog.Records;

using System.Net;

using System.Net.Http.Headers;

namespace LOM

{

class PinCodeGenerator

{

private static string RestMethodToGetPeopleId = "people";

private static string RestMethodToPostEventsHandling = "events-handling";

public static async Task IterationExecute()

{

try

{

Logger.Log(new InfoRecord("Старт генерации пин-кодов в базе данных СКУД"));

string JsonContent = await GetPeopleId();

List<People> Peoples = JSONConvertToIntList(JsonContent);

SendRequestToGeneratePincode(Peoples);

}

catch (Exception ex)

{

Logger.Log(new ErrorRecord(ex.Message));

}

}

private static async Task<string> GetPeopleId()

{

string JsonContent;

using (HttpClient Client = new HttpClient())

{

Client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

HttpResponseMessage response = await Client.GetAsync(Configuration.RESTUrl + RestMethodToGetPeopleId);

JsonContent = await response.Content.ReadAsStringAsync();

}

Logger.Log(new InfoRecord("Был получен JSON файл с PEOPLE: " + JsonContent));

return JsonContent;

}

private static List<People> JSONConvertToIntList(string JsonContent)

{

JArray JArray = JArray.Parse(JsonContent);

return JArray.ToObject<List<People>>();

}

private static async void SendRequestToGeneratePincode(List<People> Peoples)

{

using (HttpClient Client = new HttpClient())

{

foreach (People People in Peoples)

{

Client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

HttpRequestMessage Request = new HttpRequestMessage(HttpMethod.Post, Configuration.RESTUrl + RestMethodToPostEventsHandling);

Request.Content = GenerateJsonOfEventHandling(People);

HttpResponseMessage Response = await Client.SendAsync(Request);

Logger.Log(new InfoRecord("Генерация пин-кода для " + People.FirstName + " " + People.SurName));

}

}

}

private static FormUrlEncodedContent GenerateJsonOfEventHandling(People People)

{

int status = 110;

var keyValues = new List<KeyValuePair<string, string>>();

keyValues.Add(new KeyValuePair<string, string>("id\_People", People.id.ToString()));

keyValues.Add(new KeyValuePair<string, string>("id\_EventType", status.ToString()));

keyValues.Add(new KeyValuePair<string, string>("HumanOrder", "Нет"));

return new FormUrlEncodedContent(keyValues);

}

}

}

2.4. People.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LOM

{

class People

{

public int id { get; set; }

public string SurName { get; set; }

public string FirstName { get; set; }

public string PatronymicName { get; set; }

}

}

2.5. Configuration.cs

using System;

namespace LOM

{

public static class Configuration

{

public static string RESTUrl = "http://lomapi.isp.regruhosting.ru/api/";

public static string RESTToken = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIyIiwianRpIjoiNGMwZGRkYjI3OGNhODIyMzdlYjFkMjc2M2EyMmNmYzZjZmJkOWVmYzBmNzQwMGUwYzdiMmExMzcwODI5MGEwN2Y1MjY0ZGExMzI5M2YxMWIiLCJpYXQiOjE1ODg2MDI3NDEsIm5iZiI6MTU4ODYwMjc0MSwiZXhwIjoxNjIwMTM4NzQxLCJzdWIiOiIxIiwic2NvcGVzIjpbXX0.V87VBpYbpE4uT5fdWhIiaOSCieFy5vdnFy4oMPkYI3jEL5SA1kR2Z46ZQVc93gYr0c0rl37y4LnOLuARp64zRSF7Z\_w099gNZNNVJQYPyLt1o7FawgCH74nopY2-EFUXqOkUfX7yZOVgA34awrNEXNPqWc-pLgn\_7XxCy0ySO-iQrXld5Nz1yOb3pagpVSebsML25l9hhahzc7fflo66J2bCwH-FQXSbDMJ0VqOK51OlCpagYK0X3ZsT\_8BzaI6UdIgDoFtAQIdSsjqdcEddjwe\_AI-ofK3oTwzKmLhyu63vxS8uvEjjcduprAMxp0diUP3V5QSNYHbfYnF66e-\_GzSsXJx7TH7hzE0QSimQpDp9CCmAFktbRx36EnHBv3wTNhjV3athJ4DlRU6JfwfKPjagpfDK9IIkX98AGN4ScKO6fe5QR-tydns9kjtcreb4lti5l8jF6yMvbzpVoZQ23w7v-R1V9XD92BoSE2i9hynSRnlAW2wWNI3cZIZbucRcLhbE5p9Aqa0Vs3CDBt-7jDOebPlbvZvUjkHvmEwL1P9WFEhgJLiafZS4036nP6GEus6nNrBjQMSHNK\_K\_f8nlnm4CqexJynLaORhcZsWvBl1VaH5V0SMPLA2dp2M-vZkrvAWgpwt9-H-V8OeJ2g-4io7EG6O36eaKMApKfJwV-g";

public static string ScudConnectionString = "Driver={Gemini InterBase ODBC Driver 2.0};" +

"Server=localhost;" +

"DataBase=localhost:C:\\Program Files (x86)\\Cardsoft\\DuoSE\\access\\ShieldPro.gdb;" +

"Uid=sysdba;" +

"Pwd=temp;";

public static string AdminPanelUrl = "http://lomadmin.isp.regruhosting.ru/admin";

public static string FileRotation = "14";

public static string[] TimeGenerationPinCode = new string[] {"12:00", "13:00"};

public static string[] TimeUpdateDatabase = new string[] { "12:00", "13:00" };

public static string DefaultDirectoryPath = "C:\\ProgramData\\LOM";

public static string DefaultDirectoryPathToIni = "C:\\ProgramData\\LOM\\LOM.ini";

public static string DefaultDirectoryPathToLog = "C:\\ProgramData\\LOM\\Lom\_" + DateTime.Now.ToString("yyyy-MM-dd") + ".log";

public static string ParentOrg = "1";

public static void SetLogFileNewName()

{

DefaultDirectoryPathToLog = "C:\\ProgramData\\LOM\\Lom\_" + DateTime.Now.ToString("yyyy-MM-dd") + ".log";

}

}

}

2.6. Command.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LOM

{

public abstract class Command

{

public Command() { }

public abstract string CommandName { get; set; }

public abstract Task Run();

}

}

2.7. HelpCommand.cs

using System;

using System.Reflection;

using System.Threading.Tasks;

using IziLog;

using IziLog.Records;

namespace LOM

{

class HelpCommand : Command

{

public override string CommandName { get; set; } = "/help";

public async override Task Run()

{

Logger.Log(new InfoRecord("Запуск LOM.exe в режиме /help"));

Console.WriteLine("/help - вызов справки\n/silent - запуск программы в штатном режиме\n/pin - генерация пин-кодв\n/load - Выгрузка базы данных\n/stop - остановка LOM\nВерсия - " + Assembly.GetExecutingAssembly().GetName().Version.ToString() + " 2020");

}

}

}

2.8. LoadCommand.cs

using System.Threading.Tasks;

using IziLog;

using IziLog.Records;

namespace LOM

{

class LoadCommand : Command

{

public override string CommandName { get; set; } = "/load";

public async override Task Run()

{

//Task.WaitAll(DatabaseUpdater.iterationExecute());

Logger.Log(new InfoRecord("Запуск LOM.exe в режиме /load"));

await DatabaseUpdater.iterationExecute();

}

}

}

2.8. LoadCommand.cs

using System.Threading.Tasks;

using IziLog;

using IziLog.Records;

namespace LOM

{

class LoadCommand : Command

{

public override string CommandName { get; set; } = "/load";

public async override Task Run()

{

//Task.WaitAll(DatabaseUpdater.iterationExecute());

Logger.Log(new InfoRecord("Запуск LOM.exe в режиме /load"));

await DatabaseUpdater.iterationExecute();

}

}

}

2.9. PinCommand.cs

using System.Threading.Tasks;

using IziLog;

using IziLog.Records;

namespace LOM

{

class PinCommand : Command

{

public override string CommandName { get; set; } = "/pin";

public async override Task Run()

{

Logger.Log(new InfoRecord("Запуск LOM.exe в режиме /pin"));

await PinCodeGenerator.IterationExecute();

}

}

}

2.10. SilentCommand.cs

using System;

using System.Diagnostics;

using System.Runtime.InteropServices;

using System.Threading;

using System.Threading.Tasks;

using IziLog;

using IziLog.Records;

namespace LOM

{

class SilentCommand : Command

{

#region Для функции скрытия консольного окна

const int SW\_HIDE = 0;

const int SW\_SHOW = 5;

const int SW\_Min = 2;

const int SW\_Max = 3;

const int SW\_Norm = 4;

[DllImport("kernel32.dll")]

static extern IntPtr GetConsoleWindow();

[DllImport("user32.dll")]

static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

#endregion

private static bool PinCodeAlreadyGenerated = false;

private static bool DatabaseAlreadyUpdated = false;

private static int SecondsLeftToUpdateConfiguration = 60;

public override string CommandName { get; set; } = "/silent";

public async override Task Run()

{

try

{

if (IsAlreadyRunned())

{

Logger.Log(new InfoRecord("Запуск LOM.exe в режиме /silent"));

HideConsole();

Main();

}

}

catch(Exception ex)

{

Console.WriteLine("Run() - " + ex.Message);

}

}

private void HideConsole()

{

try

{

var handle = GetConsoleWindow();

//скрыть консоль

ShowWindow(handle, SW\_HIDE);

Logger.Log(new InfoRecord("Скрытие консоли"));

}

catch (Exception ex)

{

Console.WriteLine("HideConsole() - " + ex.Message);

}

}

private bool IsAlreadyRunned()

{

try

{

int CountProcess = 0;

Process[] ProcessList = Process.GetProcesses();

foreach (Process ProcessObject in ProcessList)

{

if (ProcessObject.ProcessName == "LOM") CountProcess++;

}

if (CountProcess == 1) return true;

throw new Exception("Не возможно запустить ещё один экземпляр программы");

}

catch (Exception ex)

{

Console.WriteLine("IsAlreadyRunned() - " + ex.Message);

return false;

}

}

private async void Main()

{

try

{

while (true)

{

SecondsLeftToUpdateConfiguration--;

if (SecondsLeftToUpdateConfiguration == 0)

{

Configuration.SetLogFileNewName();

Program.GetConfigurationFromFile();

SecondsLeftToUpdateConfiguration = 60;

}

await GenerateAndUpdateAsync();

string CashTime = DateTime.Now.ToString("H:mm");

Thread.Sleep(1000);

if(DateTime.Now.ToString("H:mm") != CashTime)

{

DatabaseAlreadyUpdated = false;

PinCodeAlreadyGenerated = false;

}

}

}

catch (Exception ex)

{

Console.WriteLine("Main() - " + ex.Message);

}

}

private async Task GenerateAndUpdateAsync()

{

try

{

//Logger.Log(new InfoRecord("IsTimeToGeneratePincode() = " + IsTimeToGeneratePincode() + " IsPinCodeAlreadyGenerated() = " + IsPinCodeAlreadyGenerated()));

//Logger.Log(new InfoRecord("IsTimeToUpdateDatabase() = " + IsTimeToUpdateDatabase() + "IsDatabaseAlreadyUpdated() = " + IsDatabaseAlreadyUpdated()));

if (IsTimeToGeneratePincode() && !IsPinCodeAlreadyGenerated())

{

PinCodeAlreadyGenerated = true;

//await PinCodeGenerator.IterationExecute();

Task.WaitAll(PinCodeGenerator.IterationExecute());

}

if (IsTimeToUpdateDatabase() && !IsDatabaseAlreadyUpdated())

{

DatabaseAlreadyUpdated = true;

await DatabaseUpdater.iterationExecute();

}

}

catch (Exception ex)

{

Console.WriteLine("GenerateAndUpdateAync() - " + ex.Message);

}

}

private bool IsTimeToGeneratePincode()

{

try

{

foreach (string Time in Configuration.TimeGenerationPinCode)

{

if (Time == DateTime.Now.ToString("H:mm")) return true;

}

PinCodeAlreadyGenerated = false;

return false;

}

catch(Exception ex)

{

Console.WriteLine("IsTimeToGeneratePincode() - " + ex.Message);

return false;

}

}

private bool IsTimeToUpdateDatabase()

{

try

{

foreach (string Time in Configuration.TimeUpdateDatabase)

{

if (Time == DateTime.Now.ToString("H:mm")) return true;

}

DatabaseAlreadyUpdated = false;

return false;

}

catch(Exception ex)

{

Console.WriteLine("IsTimeToUpdataDatabase() - " + ex.Message);

return false;

}

}

private bool IsPinCodeAlreadyGenerated()

{

return PinCodeAlreadyGenerated;

}

private bool IsDatabaseAlreadyUpdated()

{

return DatabaseAlreadyUpdated;

}

}

}

2.11. DatabaseUpdater.cs

using System;

using System.Collections.Generic;

using System.Data.Odbc;

using System.Net.Http;

using System.Threading.Tasks;

using Newtonsoft.Json.Linq;

using IziLog;

using IziLog.Records;

using System.Net.Http.Headers;

namespace LOM

{

class DatabaseUpdater

{

private static string RestMethodToGetPinCodes = "pin-code";

private static string RestMethodToGetPeopleId = "people";

public static async Task iterationExecute()

{

try

{

Logger.Log(new InfoRecord("Старт обновления базы данных СКУД"));

string JsonContent = await GetLastPinCodesInJSON();

List<Pincode> PincodeList = JSONConvertToObjectList(JsonContent);

DeleteOldPeople();

InsertNewData(PincodeList);

}

catch (Exception ex)

{

Logger.Log(new ErrorRecord(ex.Message));

}

}

private static async Task<string> GetLastPinCodesInJSON()

{

string JsonContent;

using (HttpClient Client = new HttpClient())

{

Client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

HttpResponseMessage response = await Client.GetAsync(Configuration.RESTUrl + RestMethodToGetPinCodes);

JsonContent = await response.Content.ReadAsStringAsync();

}

Logger.Log(new InfoRecord("Был получен JSON файл с последними пин-кодами: " + JsonContent));

return JsonContent;

}

private static List<Pincode> JSONConvertToObjectList(string JsonContent)

{

JArray JArray = JArray.Parse(JsonContent);

return JArray.ToObject<List<Pincode>>();

}

private static void DeleteOldPeople()

{

OdbcCommand Command = new OdbcCommand("DELETE FROM people p WHERE p.note LIKE 'LOM\_%'");

ExecuteQuery(Command);

Logger.Log(new InfoRecord("Удаление старых PEOPLE из базы данных СКУД"));

}

private static async void InsertNewData(List<Pincode> PincodeList)

{

foreach (Pincode Pin in PincodeList)

{

int PeopleId = await InsertNewPeople(Pin, Pin.id\_People);

InsertNewCard(Pin, PeopleId);

}

}

private static async Task<int> InsertNewPeople(Pincode Pincode, int PeopleId)

{

People CurrentPeople = await SelectPeopleFromRest(PeopleId);

OdbcCommand Command = new OdbcCommand("INSERT INTO people (ID\_DB,ID\_ORG,SURNAME,NAME,PATRONYMIC,DATEBIRTH,PLACELIFE,PLACEREG,PHONEHOME,PHONECELLULAR,PHONEWORK,NUMDOC,DATEDOC,PLACEDOC,PHOTO,WORKSTART,WORKEND,\"ACTIVE\",FLAG,LOGIN,PSWD,ID\_DEVGROUP,ID\_ORGCTRL,PEPTYPE,POST,PLACEBIRTH,SOUND,ID\_PLAN,PRESENT,NOTE,ID\_AREA,SYSNOTE)VALUES(1, " + Configuration.ParentOrg + ", '" + CurrentPeople.FirstName + "', ' " + CurrentPeople.SurName +"', '" + CurrentPeople.PatronymicName + "', '19-AUG-2008', NULL, NULL, NULL, NULL, NULL, 'X', '19-AUG-2008', 'X', NULL, '8:00:00', '17:00:00', 1, 57343, 'ACRH" + Pincode.PinCode + "', '3332', 1, 1, 1, NULL, 'X', NULL, NULL, 0, 'LOM\_" + Pincode.id\_People + "', 0, NULL);");

ExecuteQuery(Command);

Logger.Log(new InfoRecord("Вставлен People " + CurrentPeople.FirstName + " " + CurrentPeople.SurName));

return GetPeopleId("LOM\_" + Pincode.id\_People);

}

private static void InsertNewCard(Pincode Pincode, int PeopleId)

{

// Pincode.PinCode = ConvertPincode(Pincode.PinCode);

OdbcCommand Command = new OdbcCommand("INSERT INTO CARD (ID\_CARD,ID\_DB,ID\_PEP,ID\_ACCESSNAME,TIMESTART,STATUS,\"ACTIVE\", NOTE) " +

"VALUES('" + ConvertPincode(Pincode.PinCode) + "', 1, " + PeopleId + ", 1, '" + Pincode.DateTimeGen + "', 0, 1,'" + Pincode.PinCode + "'); ");

ExecuteQuery(Command);

Logger.Log(new InfoRecord("Вставлен Pin-code " + Pincode.PinCode));

}

private static async Task<People> SelectPeopleFromRest(int PeopleId)

{

string JsonContent;

using (HttpClient Client = new HttpClient())

{

Client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

HttpResponseMessage response = await Client.GetAsync(Configuration.RESTUrl + RestMethodToGetPeopleId + "/" + PeopleId);

JsonContent = await response.Content.ReadAsStringAsync();

}

//JArray JArray = JArray.Parse(JsonContent);

JObject JObject = JObject.Parse(JsonContent);

return JObject.ToObject<People>();

}

private static int GetPeopleId(string LomId)

{

OdbcCommand Command = new OdbcCommand("SELECT id\_pep FROM people p WHERE p.note = '" + LomId + "'");

object ScudId;

using (OdbcConnection Connection = new OdbcConnection(Configuration.ScudConnectionString))

{

Command.Connection = Connection;

Connection.Open();

ScudId = Command.ExecuteScalar();

}

return (int)ScudId;

}

private static void ExecuteQuery(OdbcCommand Command)

{

using (OdbcConnection Connection = new OdbcConnection(Configuration.ScudConnectionString))

{

Command.Connection = Connection;

Connection.Open();

Command.ExecuteNonQuery();

}

}

private static string ConvertPincode(string Pincode)

{

string result = "";

byte[] HexString = new byte[Pincode.Length];

for(int i = 0; i < Pincode.Length; i++) HexString[i] = Convert.ToByte(int.Parse(Pincode[i].ToString()));

Array.Reverse(HexString);

foreach (byte symbol in HexString)

{

string cash = Convert.ToString(symbol, 2).PadLeft(4, '0');

char[] inputarray = cash.ToCharArray();

Array.Reverse(inputarray);

string output = new string(inputarray);

string hex = Convert.ToInt32(output, 2).ToString("X");

result += hex;

}

return result + "00001A";

}

}

}

2.12. PinCode.cs

namespace LOM

{

public class Pincode

{

public int id { get; set; }

public int id\_People { get; set; }

public string PinCode { get; set; }

public string DateTimeGen { get; set; }

public string EndTime { get; set; }

}

}

2.13. Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LomConfig

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

2.14. Form1.cs

using System.IO;

using System.Text;

using System.Reflection;

using System.Windows.Forms;

using IniParser;

using IniParser.Model;

using IziLog;

using IziLog.Records;

using System.Data.Odbc;

using System;

using System.Diagnostics;

using System.Net.Http;

namespace LomConfig

{

public partial class MainForm : Form

{

private static string RestMethodToGetPeopleId = "peoples";

public static bool ProgramStarted = false;

public MainForm()

{

InitializeComponent();

VersionYearStatus.Text = Assembly.GetExecutingAssembly().GetName().Version.ToString() + " 2020";

IziLog.Configuration.PathToLogFile = Configuration.DefaultDirectoryPathToLog;

IziLog.Configuration.FileRotation = Convert.ToInt32(Configuration.FileRotation);

Configuration.SetLogFileNewName();

LoggerServer.Start();

if (IsFirstLaunch()) FirstStartConfiguration();

else GetConfigurationFromFile();

}

#region Methods

private bool IsFirstLaunch()

{

if (!Directory.Exists(Configuration.DefaultDirectoryPath)) return true;

return false;

}

private void FirstStartConfiguration()

{

Directory.CreateDirectory(Configuration.DefaultDirectoryPath);

using (FileStream fstream = new FileStream(Configuration.DefaultDirectoryPathToIni, FileMode.OpenOrCreate))

{

string contains = "[Main]\n" +

"UrlREST = " + Configuration.RESTUrl + "\n" +

"RESTToken = " + Configuration.RESTToken + "\n" +

"FileRotation = " + Configuration.FileRotation + "\n" +

"ScudConnectionString = " + Configuration.ScudConnectionString + "\n" +

"AdminPanelUrl = " + Configuration.AdminPanelUrl + "\n" +

"TimeGenerationPincode = " + FromArrayToString(Configuration.TimeGenerationPinCode) + "\n" +

"TimeUpdateDatabase = " + FromArrayToString(Configuration.TimeUpdateDatabase) + "\n" +

"ParentOrg = " + Configuration.ParentOrg;

byte[] array = Encoding.Default.GetBytes(contains);

fstream.Write(array, 0, array.Length);

}

Logger.Log(new InfoRecord("Процесс создания домашней директории и файла конфигурации, завершен успешно"));

GetConfigurationFromFile();

}

public void GetConfigurationFromFile()

{

try

{

Logger.Log(new InfoRecord("Считывание конфигурации, приложением LomConfig..."));

FileIniDataParser IniParser = new FileIniDataParser();

IniData IniData = IniParser.ReadFile(Configuration.DefaultDirectoryPathToIni);

Configuration.RESTUrl = IniData["Main"]["UrlREST"];

Configuration.RESTToken = IniData["Main"]["RESTToken"];

Configuration.FileRotation = IniData["Main"]["FileRotation"];

Configuration.ScudConnectionString = IniData["Main"]["ScudConnectionString"];

Configuration.AdminPanelUrl = IniData["Main"]["AdminPanelUrl"];

Configuration.TimeGenerationPinCode = FromStringToArray(IniData["Main"]["TimeGenerationPincode"]);

Configuration.TimeUpdateDatabase = FromStringToArray(IniData["Main"]["TimeUpdateDatabase"]);

Configuration.ParentOrg = IniData["Main"]["ParentOrg"];

Logger.Log(new InfoRecord("Считывание конфигурации, приложением LomConfig завершенно, прошло успешно"));

InsertConfigurationToTheField();

}

catch(Exception ex)

{

MessageBox.Show("В результате чтения конфигурационного файла, произошла ошибка. Удалите файл конфигурации и повторите попытку!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void InsertConfigurationToTheField()

{

UrlRestTbx.Text = Configuration.RESTUrl;

CkydConnTxb.Text = Configuration.ScudConnectionString;

RESTAPITokenTbx.Text = Configuration.RESTToken;

FileRotationUpD.Value = Convert.ToInt32(Configuration.FileRotation);

UrlAdminPanelTbx.Text = Configuration.AdminPanelUrl;

foreach (string time in Configuration.TimeGenerationPinCode) PinGenTimeTableLbx.Items.Add(time);

foreach (string time in Configuration.TimeUpdateDatabase) DBUpdateTimeTableLbx.Items.Add(time);

FillOrganizationTree();

SelectCurrentOrganizationNode(OrganizationTree.Nodes);

ProgramStarted = true;

}

///Вдруг попросят вернуть выпадающий список

private static string[] GetOrganizationList()

{

object StringsCount;

OdbcCommand Command = new OdbcCommand("SELECT COUNT(\*) FROM ORGANIZATION");

using (OdbcConnection Connection = new OdbcConnection(Configuration.ScudConnectionString))

{

Command.Connection = Connection;

Connection.Open();

StringsCount = Command.ExecuteScalar();

}

string[] items = new string[Convert.ToInt32(StringsCount)];

Command = new OdbcCommand("SELECT \* FROM ORGANIZATION");

OdbcDataReader reader;

using (OdbcConnection Connection = new OdbcConnection(Configuration.ScudConnectionString))

{

Command.Connection = Connection;

Connection.Open();

reader = Command.ExecuteReader();

if(reader.HasRows)

{

int i = 0;

items = new string[Convert.ToInt32(StringsCount)];

while (reader.Read())

{

items[i] += reader.GetValue(0) + " - " + reader.GetValue(2);

i++;

}

}

reader.Close();

}

return items;

}

private void FillOrganizationTree()

{

OdbcCommand Command = new OdbcCommand("SELECT \* FROM ORGANIZATION WHERE ID\_PARENT = 1");

OdbcDataReader reader;

using (OdbcConnection Connection = new OdbcConnection(Configuration.ScudConnectionString))

{

Command.Connection = Connection;

Connection.Open();

reader = Command.ExecuteReader();

TreeNode OrganizationNode;

if (reader.HasRows)

{

while (reader.Read())

{

OrganizationNode = new TreeNode(reader.GetValue(0) + " - " + reader.GetValue(2));

OrganizationTree.Nodes.Add(OrganizationNode);

if(Convert.ToInt32(reader.GetValue(0)) != 1) AddChildrenOrganizationNode(OrganizationNode, Convert.ToInt32(reader.GetValue(0)));

}

}

}

}

private void AddChildrenOrganizationNode(TreeNode ParentNode, int ParentId)

{

OdbcCommand Command = new OdbcCommand("SELECT \* FROM ORGANIZATION WHERE ID\_PARENT = " + ParentId);

OdbcDataReader reader;

using (OdbcConnection Connection = new OdbcConnection(Configuration.ScudConnectionString))

{

Command.Connection = Connection;

Connection.Open();

reader = Command.ExecuteReader();

TreeNode OrganizationNode;

if (reader.HasRows)

{

while (reader.Read())

{

OrganizationNode = new TreeNode(reader.GetValue(0) + " - " + reader.GetValue(2));

ParentNode.Nodes.Add(OrganizationNode);

AddChildrenOrganizationNode(OrganizationNode, Convert.ToInt32(reader.GetValue(0)));

}

}

}

}

private void SelectCurrentOrganizationNode(TreeNodeCollection Nodes)

{

foreach(TreeNode OrgNode in Nodes)

{

int Index = OrgNode.Text.IndexOf(" ");

if (OrgNode.Text.Substring(0, Index) == Configuration.ParentOrg)

{

OrganizationTree.SelectedNode = OrgNode;

OrganizationTree.SelectedNode.BackColor = System.Drawing.Color.LightBlue;

break;

}

else SelectCurrentOrganizationNode(OrgNode.Nodes);

}

}

private static string FromArrayToString(string[] TimeTable)

{

string FinalString = "";

foreach (string Time in TimeTable) FinalString += Time + " ";

return FinalString;

}

private static string[] FromStringToArray(string Times)

{

return Times.Split(' ');

}

private bool IsChangesExist()

{

bool ChangeExisting = false;

if (UrlRestTbx.Text != Configuration.RESTUrl) ChangeExisting = true;

if (CkydConnTxb.Text != Configuration.ScudConnectionString) ChangeExisting = true;

if (RESTAPITokenTbx.Text != Configuration.RESTToken) ChangeExisting = true;

if (UrlAdminPanelTbx.Text != Configuration.AdminPanelUrl) ChangeExisting = true;

if (FileRotationUpD.Value.ToString() != Configuration.FileRotation) ChangeExisting = true;

if(OrganizationTree.SelectedNode != null)

{

string NodeOrgStr = OrganizationTree.SelectedNode.Text;

int Index = NodeOrgStr.IndexOf(" ");

if (NodeOrgStr.Substring(0, Index) != Configuration.ParentOrg) ChangeExisting = true;

}

string[] cash = new string[PinGenTimeTableLbx.Items.Count];

PinGenTimeTableLbx.Items.CopyTo(cash, 0);

string total = FromArrayToString(cash);

if(total != FromArrayToString(Configuration.TimeGenerationPinCode)) ChangeExisting = true;

cash = new string[DBUpdateTimeTableLbx.Items.Count];

DBUpdateTimeTableLbx.Items.CopyTo(cash, 0);

total = FromArrayToString(cash);

if (total != FromArrayToString(Configuration.TimeUpdateDatabase)) ChangeExisting = true;

return ChangeExisting;

}

public void CheckChanges()

{

if (ProgramStarted && IsChangesExist()) SaveSettingStatus.Text = "Есть несохраненные изменения";

else SaveSettingStatus.Text = "Изменения сохранены";

SaveConfBtn.Enabled = IsChangesExist();

}

public void SaveProcess()

{

try

{

FileIniDataParser IniParser = new FileIniDataParser();

IniData IniData = new IniData();

Configuration.RESTUrl = UrlRestTbx.Text;

Configuration.ScudConnectionString = CkydConnTxb.Text;

Configuration.RESTToken = RESTAPITokenTbx.Text;

Configuration.FileRotation = FileRotationUpD.Value.ToString();

Configuration.AdminPanelUrl = UrlAdminPanelTbx.Text;

string[] cash = new string[PinGenTimeTableLbx.Items.Count];

PinGenTimeTableLbx.Items.CopyTo(cash, 0);

Configuration.TimeGenerationPinCode = cash;

cash = new string[DBUpdateTimeTableLbx.Items.Count];

DBUpdateTimeTableLbx.Items.CopyTo(cash, 0);

Configuration.TimeUpdateDatabase = cash;

string NodeOrgStr = OrganizationTree.SelectedNode.Text;

int Index = NodeOrgStr.IndexOf(" ");

Configuration.ParentOrg = NodeOrgStr.Substring(0, Index);

IniData["Main"]["UrlREST"] = Configuration.RESTUrl;

IniData["Main"]["RESTToken"] = Configuration.RESTToken;

IniData["Main"]["FileRotation"] = Configuration.FileRotation;

IniData["Main"]["ScudConnectionString"] = Configuration.ScudConnectionString;

IniData["Main"]["AdminPanelUrl"] = Configuration.AdminPanelUrl;

IniData["Main"]["TimeGenerationPincode"] = FromArrayToString(Configuration.TimeGenerationPinCode);

IniData["Main"]["TimeUpdateDatabase"] = FromArrayToString(Configuration.TimeUpdateDatabase);

IniData["Main"]["ParentOrg"] = Configuration.ParentOrg;

IniParser.WriteFile(Configuration.DefaultDirectoryPathToIni, IniData);

Logger.Log(new InfoRecord("Ручное изменение конфигурации, прошло успешно"));

CheckChanges();

MessageBox.Show("Сохранено успешно!");

}

catch (Exception ex)

{

Logger.Log(new ErrorRecord(ex.Message));

MessageBox.Show(ex.Message);

}

finally

{

Logger.Log(new WarningRecord("Были использованы следующие настройки:\nUrlRest = " + Configuration.RESTUrl + "\n" +

"RESTTokem = " + Configuration.RESTToken + "\n" +

"FileRotation = " + Configuration.FileRotation + "\n" +

"ScudConnectionString = " + Configuration.ScudConnectionString + "\n" +

"AdminPanelUrl = " + Configuration.AdminPanelUrl + "\n" +

"TimeGenerationPincode = " + FromArrayToString(Configuration.TimeGenerationPinCode) + "\n" +

"TimeUpdateDatabase = " + FromArrayToString(Configuration.TimeUpdateDatabase) + "\n" +

"ParentOrg = " + Configuration.ParentOrg));

}

}

#endregion

#region Events

private void AddPinCodeGenTimeBtn\_Click(object sender, EventArgs e)

{

try

{

DateTime Time = Convert.ToDateTime(PinGenMTbx.Text);

PinGenTimeTableLbx.Items.Add(Time.ToString("H:mm"));

CheckChanges();

}

catch(Exception ex)

{

MessageBox.Show("Неверный формат данных!");

}

}

private void DeletePinCodeGenTimeBtn\_Click(object sender, EventArgs e)

{

PinGenTimeTableLbx.Items.Remove(PinGenTimeTableLbx.SelectedItem);

CheckChanges();

}

private void AddDBUpdateTimeBtn\_Click(object sender, EventArgs e)

{

try

{

DateTime Time = Convert.ToDateTime(DBUpdaterMTbx.Text);

DBUpdateTimeTableLbx.Items.Add(Time.ToString("H:mm"));

CheckChanges();

}

catch(Exception ex)

{

MessageBox.Show("Неверный формат данных!");

}

}

private void DeleteDBUpdateTimeBtn\_Click(object sender, EventArgs e)

{

DBUpdateTimeTableLbx.Items.Remove(DBUpdateTimeTableLbx.SelectedItem);

CheckChanges();

}

private void SaveConfBtn\_Click(object sender, EventArgs e)

{

SaveProcess();

}

private void CheckConnectionTbx\_Click(object sender, EventArgs e)

{

try

{

OdbcCommand Command = new OdbcCommand("SELECT \* FROM people");

object ScudId;

using (OdbcConnection Connection = new OdbcConnection(Configuration.ScudConnectionString))

{

Command.Connection = Connection;

Connection.Open();

OdbcDataReader OdbcReader = Command.ExecuteReader();

}

Logger.Log(new InfoRecord("Проверка подключения к базе данных СКУД прошло успешно"));

MessageBox.Show("Статус: ОК\nОписание: Подключние успешно установлено!", "Проверка соединения со СКУД");

}

catch (Exception ex)

{

Logger.Log(new ErrorRecord("Проверка подключения к базе данных СКУД прошло с ошибкой " + ex.Message));

MessageBox.Show("Статус: Ошибка\nОписание: Произошёл сбой в подключение! " + ex.Message, "Проверка соединения со СКУД");

}

}

private async void CheckConnectionRestBtn\_Click(object sender, EventArgs e)

{

try

{

string JsonContent;

using (HttpClient Client = new HttpClient())

{

HttpResponseMessage response = await Client.GetAsync(Configuration.RESTUrl + RestMethodToGetPeopleId);

JsonContent = await response.Content.ReadAsStringAsync();

}

Logger.Log(new InfoRecord("Проверка подключения c REST прошло успешно"));

MessageBox.Show("Статус: ОК\nОписание: Подключние успешно установлено!", "Проверка соединения с REST");

}

catch(Exception ex)

{

Logger.Log(new ErrorRecord("Проверка подключения c REST прошло с ошибкой " + ex.Message));

MessageBox.Show("Статус: Ошибка\nОписание: Произошёл сбой в подключение! " + ex.Message, "Проверка соединения с Rest");

}

}

private void OpenFolderBtn\_Click(object sender, EventArgs e)

{

Process.Start("explorer.exe", Configuration.DefaultDirectoryPath);

}

private void OpenAdminPanelBtn\_Click(object sender, EventArgs e)

{

Process.Start(Configuration.AdminPanelUrl);

}

private void HelpMenuBtn\_Click(object sender, EventArgs e)

{

MessageBox.Show("/help - вызов справки\n/silent - запуск программы в штатном режиме\n/pin - генерация пин-кодв\n/load - Выгрузка базы данных\n/stop - остановка LOM", "Помощь", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void AboutMenuBtn\_Click(object sender, EventArgs e)

{

MessageBox.Show("Разработчик: Бушин Алексей Юрьевич\nООО \"Арстек\" 2020\n", "О программе", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void UrlRestTbx\_TextChanged(object sender, EventArgs e)

{

CheckChanges();

}

private void MainForm\_FormClosing(object sender, FormClosingEventArgs e)

{

if (IsChangesExist())

{

DialogResult Result = MessageBox.Show("Имеются несохраненные данные, сохранить перед выходом?", "Есть несохраненные данные", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (Result.Equals(DialogResult.Yes)) SaveProcess();

}

}

private void OrganizationTree\_AfterSelect(object sender, TreeViewEventArgs e)

{

CheckChanges();

}

#endregion

}

}

2.15. Configuration.cs

using System;

namespace LomConfig

{

public static class Configuration

{

public static string RESTUrl = "http://lomapi.isp.regruhosting.ru//api/";

public static string RESTToken = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIxIiwianRpIjoiM2RlOWIxZTM5ZTZkOTI3N2I4NjM1YmVlZTE2N2JmYWMxNzU2NGE3Nzg3NzQ4N2JlZjY4ZTllOGFlZTU1Yzg5Njk4ZDJkYzM4OGEyZWUwMmEiLCJpYXQiOjE1ODg2MDE0OTUsIm5iZiI6MTU4ODYwMTQ5NSwiZXhwIjoxNjIwMTM3NDk1LCJzdWIiOiIxIiwic2NvcGVzIjpbXX0.OMtVubD9kN--ak4uXlc5h82erCZD3Pd8lJ440PWgiW5e5oW2sCibN33xr5KBGoR\_VeZc55yboxqvEcu5J8X8ny4gvi3ML8O3Gkzwh3S9pDOyGbDU2tTsOSSOL3d9g96QoDJHY7rkqnQWSVtwFUw0Svhiy-xUkvpJDuMy3acy-gRMbjMlwzIzEOEBx1spn9T6uxQrM1BWuk3cW\_qZC\_cNObLj4Ylyd\_riKwn4loPPVyQsTKguTfqif9ub45TMewcoo2qzGRR1AhRJWXCBGeP\_s4BwH6OGnGqJkHOHCK0aeHgFeHteqGy\_FVkyNa5luuSJFGpU0wk1mLVLD9Yh3ty632NDM83N-nqTFqlxUdlt\_LR8oND4lCPaoVFnDFwZH-QvEz-r\_NgsFEX6KjGubL3euiP0aXhkRpQFmBLIw6AQAdntqfoeub9BiEbH-t\_aZy7UWjUOQxp5XEdZLS\_cR2h2uic5goGAbsgNGGmZjrHQfrcHp9BKDnpMG7v\_YynQAzLwXKVSFoVGC\_mnMwiBb0SR5UJGp8Bvfy2H6r0vh-j\_p094ISyzhWTf5Ah0SBDNfIcAuVqsMJtYhWWGcd5x-BkLIaBDLzL\_I71VizuK-7BAslKK2VMbBSY8dRVDjbyLCarFDQS\_ZLqxWEfx0cheLtwoDKnQfuPKMswMgM4ljjH2d8M";

public static string ScudConnectionString = "Driver={Gemini InterBase ODBC Driver 2.0};" +

"Server=localhost;" +

"DataBase=localhost:C:\\Program Files (x86)\\Cardsoft\\DuoSE\\access\\ShieldPro.gdb;" +

"Uid=sysdba;" +

"Pwd=temp;";

public static string AdminPanelUrl = "http://lomadmin.isp.regruhosting.ru/admin";

public static string FileRotation = "14";

public static string[] TimeGenerationPinCode = new string[] {"12:00", "13:00"};

public static string[] TimeUpdateDatabase = new string[] { "12:00", "13:00" };

public static string DefaultDirectoryPath = "C:\\ProgramData\\LOM";

public static string DefaultDirectoryPathToIni = "C:\\ProgramData\\LOM\\LOM.ini";

public static string DefaultDirectoryPathToLog = "C:\\ProgramData\\LOM\\Lom\_" + DateTime.Now.ToString("yyyy-MM-dd") + ".log";

public static string ParentOrg = "1";

public static void SetLogFileNewName()

{

DefaultDirectoryPathToLog = "C:\\ProgramData\\LOM\\Lom\_" + DateTime.Now.ToString("yyyy-MM-dd") + ".log";

}

}

}

2.16. App.xaml.cs

using System;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace LomMobile

{

public partial class App : Application

{

public App()

{

InitializeComponent();

MainPage = new NavigationPage(new AuthorizationPage());

}

protected override void OnStart()

{

}

protected override void OnSleep()

{

}

protected override void OnResume()

{

}

}

}

2.17. AuthorizationPage.xaml.cs

using System;

using System.Collections.Generic;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Threading;

using System.Threading.Tasks;

using LomMobile.System;

using Newtonsoft.Json.Linq;

using Xamarin.Essentials;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace LomMobile

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class AuthorizationPage : ContentPage

{

public AuthorizationPage()

{

InitializeComponent();

StartCheckInternet();

StartCheckInternetStatus();

}

private async void StartCheckInternet()

{

await InternetChecker.Start();

}

private async void StartCheckInternetStatus()

{

await Task.Run(() =>

{

while (true)

{

Thread.Sleep(100);

Device.BeginInvokeOnMainThread(() =>

{

if (!InternetChecker.IsInternetConnected && !InternetChecker.PageShowed)

{

Navigation.PushModalAsync(new NoInternetPage());

InternetChecker.PageShowed = true;

LoginBtn.IsEnabled = false;

ToRegistrationBtn.IsEnabled = false;

}

else if (InternetChecker.IsInternetConnected && InternetChecker.PageShowed)

{

Navigation.PopModalAsync();

InternetChecker.PageShowed = false;

LoginBtn.IsEnabled = true;

ToRegistrationBtn.IsEnabled = true;

}

});

}

});

}

private bool IsValidate()

{

bool Password = true, Login = true;

if (LoginEntr.Text == null || LoginEntr.Text == "")

{

LoginErrorLbl.Text = "Введите логин";

Password = false;

}

if (PasswordEntr.Text == null || PasswordEntr.Text == "")

{

PasswordErrorLbl.Text = "Введите пароль";

Login = false;

}

if (Password && Login) return true;

else return false;

}

private async Task<bool> IsAuthorized()

{

try

{

User User = await GetUserAsync();

if (User.id != 0)

{

UserInfo.Id = User.id;

UserInfo.FirstName = User.FirstName;

UserInfo.SurName = User.SurName;

UserInfo.PatronymicName = User.PatronymicName;

return true;

}

else return false;

}

catch (Exception ex)

{

return false;

}

}

private async Task<User> GetUserAsync()

{

string JsonContent;

using (HttpClient Client = new HttpClient())

{

HttpRequestMessage Request = new HttpRequestMessage(HttpMethod.Post, Configuration.RESTUrl + "login");

Request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

Request.Content = GenerateJsonForLogin();

HttpResponseMessage Response = await Client.SendAsync(Request);

JsonContent = await Response.Content.ReadAsStringAsync();

}

JObject JObject = JObject.Parse(JsonContent);

return JObject.ToObject<User>();

}

private FormUrlEncodedContent GenerateJsonForLogin()

{

var keyValues = new List<KeyValuePair<string, string>>();

keyValues.Add(new KeyValuePair<string, string>("Login", LoginEntr.Text));

keyValues.Add(new KeyValuePair<string, string>("Password", PasswordEntr.Text));

return new FormUrlEncodedContent(keyValues);

}

private async Task lmao()

{

await Task.Run(() =>

{

Thread.Sleep(2000);

Device.BeginInvokeOnMainThread(() =>

{

LoginErrorLbl.Text = "Вроде работает!";

});

});

}

private async void LoginBtn\_Clicked(Object sender, EventArgs e)

{

UserInfo.FirstName = "";

UserInfo.Id = 0;

UserInfo.PatronymicName = "";

UserInfo.PinCode = "";

UserInfo.SurName = "";

LoginErrorLbl.Text = "";

PasswordErrorLbl.Text = "";

LoginResultLbl.Text = "";

bool IsValid = IsValidate();

bool IsAuth = await IsAuthorized();

if (IsValid && IsAuth) await Navigation.PushAsync(new MainPage());

else if (IsValid) LoginResultLbl.Text = "Неверный логин или пароль";

}

private async void ToRegistrationBtn\_Clicked(Object sender, EventArgs e)

{

await Navigation.PushModalAsync(new RegistrationPage());

}

}

}

2.18. AuthorizationPage.xaml

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:d="http://xamarin.com/schemas/2014/forms/design"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

x:Class="LomMobile.AuthorizationPage"

Title="LomMobile">

<ContentPage.Content>

<StackLayout>

<Grid VerticalOptions="CenterAndExpand"

HorizontalOptions="CenterAndExpand">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Label Grid.Row="0"

Text="LOM"

FontSize="33"

HorizontalOptions="CenterAndExpand"/>

<Label Grid.Row="1"

Text="Авторизация"

FontSize="30"

HorizontalOptions="CenterAndExpand"/>

<Label Grid.Row="2"

x:Name="LoginErrorLbl"

TextColor="Red"/>

<Entry x:Name="LoginEntr"

Grid.Row="3"

Placeholder="Логин" />

<Label Grid.Row="4"

x:Name="PasswordErrorLbl"

TextColor="Red"/>

<Entry x:Name="PasswordEntr"

Grid.Row="5"

Placeholder="Пароль"

IsPassword="True"/>

<Button x:Name="LoginBtn"

Text="Войти"

Grid.Row="6"

Clicked="LoginBtn\_Clicked"/>

<Button x:Name="ToRegistrationBtn"

Text="Регистрация"

Grid.Row="7"

Clicked="ToRegistrationBtn\_Clicked"/>

<Label Grid.Row="8"

x:Name="LoginResultLbl"

TextColor="red"/>

</Grid>

</StackLayout>

</ContentPage.Content>

</ContentPage>

2.19. RegistrationPage.xaml.cs

using System;

using System.Threading;

using System.Collections.Generic;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Threading.Tasks;

using Newtonsoft.Json.Linq;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace LomMobile

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class RegistrationPage : ContentPage

{

public RegistrationPage()

{

InitializeComponent();

StartCheckInternetStatus();

}

private async void StartCheckInternetStatus()

{

await Task.Run(() =>

{

while (true)

{

Thread.Sleep(100);

Device.BeginInvokeOnMainThread(() =>

{

if (!InternetChecker.IsInternetConnected && !InternetChecker.PageShowed) RegBtn.IsEnabled = false;

else if (InternetChecker.IsInternetConnected && InternetChecker.PageShowed) RegBtn.IsEnabled = true;

});

}

});

}

private async void RegBtn\_Clicked(Object sender, EventArgs e)

{

SurNameErrorLbl.Text = "";

FirstNameErrorLbl.Text = "";

PatronymicNameErrorLbl.Text = "";

LoginErrorLbl.Text = "";

PasswordErrorLbl.Text = "";

RepeatPasswordErrorLbl.Text = "";

bool IsValid = IsValidate();

bool IsRegistred = await IsRegistered();

if (IsValid && IsRegistred) await Navigation.PopModalAsync();

}

private bool IsValidate()

{

bool SurName = true, FirstName = true, PatronymicName = true, Login = true, Password = true, RepeatPassword = true;

if(SurNameEntr.Text == null || SurNameEntr.Text == "")

{

SurNameErrorLbl.Text = "Введите фамилию";

SurName = false;

}

else if (SurNameEntr.Text == null || SurNameEntr.Text.Length < 3)

{

SurNameErrorLbl.Text = "Фамилия должна быть длинее 3-ёх символов";

SurName = false;

}

if (FirstNameEntr.Text == null || FirstNameEntr.Text == "")

{

FirstNameErrorLbl.Text = "Введите имя";

FirstName = false;

}

else if (FirstNameEntr.Text == null || FirstNameEntr.Text.Length < 3)

{

FirstNameErrorLbl.Text = "Имя должно быть длинее 2-ух символов";

FirstName = false;

}

if(LoginEntr.Text == null || LoginEntr.Text == "")

{

LoginErrorLbl.Text = "Введите логин";

Login = false;

}

else if (LoginEntr.Text == null || LoginEntr.Text.Length < 6)

{

LoginErrorLbl.Text = "Логин должен быть длинее 6-ти символов";

Login = false;

}

if (PasswordEntr.Text == null || PasswordEntr.Text == "")

{

PasswordErrorLbl.Text = "Введите пароль";

Password = false;

}

else if (PasswordEntr.Text == null || PasswordEntr.Text.Length < 6)

{

PasswordErrorLbl.Text = "Пароль должен быть длинее 6-ти символов";

Password = false;

}

if (RepeatPasswordEntr.Text == null || PasswordEntr.Text != RepeatPasswordEntr.Text)

{

RepeatPasswordErrorLbl.Text = "Пароли не совпадают";

RepeatPassword = false;

}

if (SurName && FirstName && PatronymicName && Login && Password && RepeatPassword) return true;

else return false;

}

private async Task<bool> IsRegistered()

{

try

{

User User = await GetUserAsync();

if (User.id != 0) return true;

else return false;

}

catch (Exception ex)

{

return false;

}

}

private async Task<User> GetUserAsync()

{

string JsonContent;

using (HttpClient Client = new HttpClient())

{

HttpRequestMessage Request = new HttpRequestMessage(HttpMethod.Post, Configuration.RESTUrl + "people");

Request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

Request.Content = GenerateJsonForLogin();

HttpResponseMessage Response = await Client.SendAsync(Request);

JsonContent = await Response.Content.ReadAsStringAsync();

}

JObject JObject = JObject.Parse(JsonContent);

return JObject.ToObject<User>();

}

private FormUrlEncodedContent GenerateJsonForLogin()

{

var keyValues = new List<KeyValuePair<string, string>>();

keyValues.Add(new KeyValuePair<string, string>("SurName", SurNameEntr.Text));

keyValues.Add(new KeyValuePair<string, string>("FirstName", FirstNameEntr.Text));

keyValues.Add(new KeyValuePair<string, string>("PatronymicName", PatronymicNameEntr.Text));

keyValues.Add(new KeyValuePair<string, string>("Login", LoginEntr.Text));

keyValues.Add(new KeyValuePair<string, string>("Password", PasswordEntr.Text));

return new FormUrlEncodedContent(keyValues);

}

private async void BackBtn\_Clicked(Object sender, EventArgs e)

{

await Navigation.PopModalAsync();

}

}

}

2.20. RegistrationPage.xaml

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:d="http://xamarin.com/schemas/2014/forms/design"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

x:Class="LomMobile.RegistrationPage"

Title="LomMobile">

<ContentPage.Content>

<StackLayout>

<ScrollView>

<Grid VerticalOptions="CenterAndExpand">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Label Grid.Row="0"

Text="Регистрация"

FontSize="30"

HorizontalOptions="CenterAndExpand"/>

<Label Grid.Row="1"

x:Name="SurNameErrorLbl"

TextColor="Red"/>

<Entry x:Name="SurNameEntr"

Grid.Row="2"

Placeholder="\* Фамилия"/>

<Label Grid.Row="3"

x:Name="FirstNameErrorLbl"

TextColor="Red"/>

<Entry x:Name="FirstNameEntr"

Grid.Row="4"

Placeholder="\* Имя"/>

<Label Grid.Row="5"

x:Name="PatronymicNameErrorLbl"

TextColor="Red"/>

<Entry x:Name="PatronymicNameEntr"

Grid.Row="6"

Placeholder="Отчество"/>

<Label Grid.Row="7"

x:Name="LoginErrorLbl"

TextColor="Red"/>

<Entry x:Name="LoginEntr"

Grid.Row="8"

Placeholder="\* Логин"/>

<Label Grid.Row="9"

x:Name="PasswordErrorLbl"

TextColor="Red"/>

<Entry x:Name="PasswordEntr"

Grid.Row="10"

Placeholder="\* Пароль"

IsPassword="True"/>

<Label Grid.Row="11"

x:Name="RepeatPasswordErrorLbl"

TextColor="Red"/>

<Entry x:Name="RepeatPasswordEntr"

Grid.Row="12"

Placeholder="\* Повторите пароль"

IsPassword="True"/>

<Button x:Name="RegBtn"

Text="Зарегестрироваться"

Grid.Row="13"

Clicked="RegBtn\_Clicked"/>

<Button x:Name="BackBtn"

Text="Назад"

Grid.Row="14"

Clicked="BackBtn\_Clicked"/>

</Grid>

</ScrollView>

</StackLayout>

</ContentPage.Content>

</ContentPage>

2.21. MainPage.xaml.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Newtonsoft.Json.Linq;

using LomMobile.System;

namespace LomMobile

{

// Learn more about making custom code visible in the Xamarin.Forms previewer

// by visiting https://aka.ms/xamarinforms-previewer

[DesignTimeVisible(false)]

public partial class MainPage : ContentPage

{

public MainPage()

{

InitializeComponent();

FIOLbl.Text = UserInfo.FirstName + " " + UserInfo.SurName;

SetPinCode();

}

private async void SetPinCode()

{

try

{

Pincode Pincode = await GetPinCode(UserInfo.Id);

Device.BeginInvokeOnMainThread(() =>

{

ActivePinCodeLbl.Text = Pincode.PinCode;

});

}

catch(Exception ex)

{

Device.BeginInvokeOnMainThread(() =>

{

ActivePinCodeLbl.Text = "Пин-код отсутствует";

});

}

}

private async Task<Pincode> GetPinCode(int UserId)

{

string JsonContent;

using (HttpClient Client = new HttpClient())

{

Client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

HttpResponseMessage response = await Client.GetAsync(Configuration.RESTUrl + "pin-code/" + UserId.ToString());

JsonContent = await response.Content.ReadAsStringAsync();

}

JObject JObject = JObject.Parse(JsonContent);

return JObject.ToObject<Pincode>();

}

private void ReFreshBtn\_Clicked(Object sender, EventArgs e)

{

SetPinCode();

}

private async void GenerateBtn\_Clicked(Object sender, EventArgs e)

{

try

{

using (HttpClient Client = new HttpClient())

{

Client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", Configuration.RESTToken);

HttpRequestMessage Request = new HttpRequestMessage(HttpMethod.Post, Configuration.RESTUrl + "events-handling");

Request.Content = GenerateJsonOfEventHandling(UserInfo.Id);

HttpResponseMessage Response = await Client.SendAsync(Request);

}

}

catch

{

}

finally

{

SetPinCode();

}

}

private static FormUrlEncodedContent GenerateJsonOfEventHandling(int Id)

{

int status = 110;

var keyValues = new List<KeyValuePair<string, string>>();

keyValues.Add(new KeyValuePair<string, string>("id\_People", Id.ToString()));

keyValues.Add(new KeyValuePair<string, string>("id\_EventType", status.ToString()));

keyValues.Add(new KeyValuePair<string, string>("HumanOrder", "Нет"));

return new FormUrlEncodedContent(keyValues);

}

}

}

2.22. MainPage.xaml

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:d="http://xamarin.com/schemas/2014/forms/design"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

x:Class="LomMobile.MainPage"

Title="Профиль">

<StackLayout>

<Grid VerticalOptions="CenterAndExpand"

HorizontalOptions="CenterAndExpand">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Label Grid.Row="1"

x:Name="FIOLbl"

FontSize="30"

HorizontalOptions="CenterAndExpand"/>

<Label Grid.Row="2"

Text="Активный пин-код"

FontSize="30"

HorizontalOptions="CenterAndExpand"/>

<Label Grid.Row="3"

x:Name="ActivePinCodeLbl"

FontSize="30"

HorizontalOptions="CenterAndExpand"/>

<Button Grid.Row="4"

x:Name="ReFreshBtn"

Text="Обновить"

Clicked="ReFreshBtn\_Clicked"/>

<Button Grid.Row="5"

x:Name="GenerateBtn"

Text="Сгенерировать"

Clicked="GenerateBtn\_Clicked"/>

</Grid>

</StackLayout>

</ContentPage>

2.23. NoInternetPage.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace LomMobile

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class NoInternetPage : ContentPage

{

public NoInternetPage()

{

InitializeComponent();

}

}

}

2.24. NoInternetPage.xaml

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:d="http://xamarin.com/schemas/2014/forms/design"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

x:Class="LomMobile.NoInternetPage">

<ContentPage.Content>

<StackLayout>

<Image Source="NoInternet.png"

VerticalOptions="CenterAndExpand"

HorizontalOptions="CenterAndExpand"/>

</StackLayout>

</ContentPage.Content>

</ContentPage>

2.25. Configuration.cs

namespace LomMobile

{

public static class Configuration

{

public static string RESTUrl = "http://lomapi.isp.regruhosting.ru/api/";

public static string RESTToken = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIxIiwianRpIjoiM2RlOWIxZTM5ZTZkOTI3N2I4NjM1YmVlZTE2N2JmYWMxNzU2NGE3Nzg3NzQ4N2JlZjY4ZTllOGFlZTU1Yzg5Njk4ZDJkYzM4OGEyZWUwMmEiLCJpYXQiOjE1ODg2MDE0OTUsIm5iZiI6MTU4ODYwMTQ5NSwiZXhwIjoxNjIwMTM3NDk1LCJzdWIiOiIxIiwic2NvcGVzIjpbXX0.OMtVubD9kN--ak4uXlc5h82erCZD3Pd8lJ440PWgiW5e5oW2sCibN33xr5KBGoR\_VeZc55yboxqvEcu5J8X8ny4gvi3ML8O3Gkzwh3S9pDOyGbDU2tTsOSSOL3d9g96QoDJHY7rkqnQWSVtwFUw0Svhiy-xUkvpJDuMy3acy-gRMbjMlwzIzEOEBx1spn9T6uxQrM1BWuk3cW\_qZC\_cNObLj4Ylyd\_riKwn4loPPVyQsTKguTfqif9ub45TMewcoo2qzGRR1AhRJWXCBGeP\_s4BwH6OGnGqJkHOHCK0aeHgFeHteqGy\_FVkyNa5luuSJFGpU0wk1mLVLD9Yh3ty632NDM83N-nqTFqlxUdlt\_LR8oND4lCPaoVFnDFwZH-QvEz-r\_NgsFEX6KjGubL3euiP0aXhkRpQFmBLIw6AQAdntqfoeub9BiEbH-t\_aZy7UWjUOQxp5XEdZLS\_cR2h2uic5goGAbsgNGGmZjrHQfrcHp9BKDnpMG7v\_YynQAzLwXKVSFoVGC\_mnMwiBb0SR5UJGp8Bvfy2H6r0vh-j\_p094ISyzhWTf5Ah0SBDNfIcAuVqsMJtYhWWGcd5x-BkLIaBDLzL\_I71VizuK-7BAslKK2VMbBSY8dRVDjbyLCarFDQS\_ZLqxWEfx0cheLtwoDKnQfuPKMswMgM4ljjH2d8M";

}

}

2.26. InternetChecker.cs

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using Xamarin.Essentials;

namespace LomMobile

{

public static class InternetChecker

{

public static bool IsInternetConnected;

public static bool PageShowed;

public async static Task Start()

{

await Task.Run(() =>

{

while (true)

{

Check();

Thread.Sleep(100);

}

});

}

public static void Check()

{

var current = Connectivity.NetworkAccess;

if (current == NetworkAccess.Internet) IsInternetConnected = true;

else IsInternetConnected = false;

}

}

}

2.27. PinCode.cs

namespace LomMobile

{

class Pincode

{

public int id { get; set; }

public int id\_People { get; set; }

public string PinCode { get; set; }

public int PinCodeNumberUses { get; set; }

public int id\_PinCodeType { get; set; }

public string PinCodeAccess { get; set; }

public string DateTimeGen { get; set; }

public string EndTime { get; set; }

}

}

2.28. User,cs

namespace LomMobile

{

class User

{

public int id { get; set; }

public string SurName { get; set; }

public string FirstName { get; set; }

public string PatronymicName { get; set; }

public string Login { get; set; }

}

}

2.29. UserInfo.cs

namespace LomMobile.System

{

public static class UserInfo

{

public static int Id;

public static string SurName;

public static string FirstName;

public static string PatronymicName;

public static string PinCode;

}

}

2.30. AccessList.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class AccessList extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_AccessList';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.31. EventResult.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class EventResult extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_EventResult';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.32. EventsLogs.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class EventsLogs extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_EventsLogs';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.33. Organization.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Organization extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_Organizations';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.34. People.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class People extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_People';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.35. PeopleAccessInterim.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class People extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_People';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.36. PeopleContact.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class PeopleContact extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_PeopleContact';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.37. PeopleOrganization.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class PeopleOrganizations extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_PeopleOrganizations';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.38. PinCode.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Pincode extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_PinCode';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.39. AccessGrid.php

<?php

namespace App\Admin;

use Encore\Admin\Grid;

class AccessGrid extends Grid

{

/\*\*

\* Render create button for grid.

\*

\* @return string

\*/

public function renderCreateButton()

{

return (new AddRoleButton($this))->render();

}

/\*\*

\* Get create url.

\*

\* @return string

\*/

public function getCreateUrl()

{

$queryString = '';

if ($constraints = $this->model()->getConstraints()) {

$queryString = http\_build\_query($constraints);

}

return sprintf('%s/add-access%s',

$this->resource(),

$queryString ? ('?'.$queryString) : ''

);

}

}

?>

2.40. AddOrganizationButton.php

<?php

namespace App\Admin;

use Encore\Admin\Grid\Tools\CreateButton;

class AddOrganizationButton extends CreateButton

{

/\*\*

\* Render CreateButton.

\*

\* @return string

\*/

public function render()

{

if (!$this->grid->showCreateBtn()) {

return '';

}

$new = trans('admin.new');

return <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$this->grid->getCreateUrl()}" class="btn btn-sm btn-success" title="{$new}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$new}</span>

</a>

</div>

EOT;

}

}

?>

2.41. AddRoleButton.php

<?php

namespace App\Admin;

use Encore\Admin\Grid\Tools\CreateButton;

class AddRoleButton extends CreateButton

{

/\*\*

\* Render CreateButton.

\*

\* @return string

\*/

public function render()

{

if (!$this->grid->showCreateBtn()) {

return '';

}

$new = trans('admin.new');

return <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$this->grid->getCreateUrl()}" class="btn btn-sm btn-success" title="{$new}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$new}</span>

</a>

</div>

EOT;

}

}

?>

2.42. OrganizationGrid.php

<?php

namespace App\Admin;

use Encore\Admin\Grid;

class OrganizationGrid extends Grid

{

/\*\*

\* Render create button for grid.

\*

\* @return string

\*/

public function renderCreateButton()

{

return (new AddOrganizationButton($this))->render();

}

/\*\*

\* Get create url.

\*

\* @return string

\*/

public function getCreateUrl()

{

$queryString = '';

if ($constraints = $this->model()->getConstraints()) {

$queryString = http\_build\_query($constraints);

}

return sprintf('%s/add-organization%s',

$this->resource(),

$queryString ? ('?'.$queryString) : ''

);

}

}

?>

2.43. AccessListController.php

<?php

namespace App\Admin\Controllers;

use App\AccessList;

use Encore\Admin\Controllers\AdminController;

use Encore\Admin\Form;

use Encore\Admin\Grid;

use Encore\Admin\Show;

class AccessListController extends CustomAdminController

{

/\*\*

\* Title for current resource.

\*

\* @var string

\*/

protected $title = 'AccessList';

/\*\*

\* Make a grid builder.

\*

\* @return Grid

\*/

protected function grid()

{

$grid = new Grid(new AccessList());

$grid->disableExport();

$grid->disableFilter();

$grid->column('id', \_\_('Id'));

$grid->column('Access', \_\_(trans('base.Access')));

$grid->column('Comment', \_\_(trans('base.Content')));

return $grid;

}

/\*\*

\* Make a show builder.

\*

\* @param mixed $id

\* @return Show

\*/

protected function detail($id)

{

$show = new Show(AccessList::findOrFail($id));

$show->field('id', \_\_('Id'));

$show->field('Access', \_\_(trans('base.Access')));

$show->field('Comment', \_\_(trans('base.Content')));

return $show;

}

/\*\*

\* Make a form builder.

\*

\* @return Form

\*/

protected function form()

{

$form = new Form(new AccessList());

$form->text('Access', \_\_(trans('base.Access')))->rules('required');

$form->text('Comment', \_\_(trans('base.Content')))->rules('required');

return $form;

}

}

2.44. CustomAdminController.php

<?php

namespace App\Admin\Controllers;

use Encore\Admin\Controllers\AdminController;

class CustomAdminController extends AdminController

{

/\*\*

\* Get content title.

\*

\* @return string

\*/

protected function title()

{

return trans('base.'. $this->title);

}

}

?>

2.45. EventsLogsController.php

<?php

namespace App\Admin\Controllers;

use App\EventsLogs;

use App\People;

use App\EventResult;

use Encore\Admin\Controllers\AdminController;

use Encore\Admin\Form;

use Encore\Admin\Grid;

use Encore\Admin\Show;

class EventsLogsController extends CustomAdminController

{

/\*\*

\* Title for current resource.

\*

\* @var string

\*/

protected $title = 'EventsLogs';

/\*\*

\* Make a grid builder.

\*

\* @return Grid

\*/

protected function grid()

{

$grid = new Grid(new EventsLogs());

$grid->filter(function($filter)

{

$filter->disableIdFilter();

$filter->date('DateTimeEvent', \_\_(trans('base.DateTimeEvent')));

});

$grid->disableExport();

$grid->disableCreation();

$grid->option('show\_actions', false);

$grid->option('show\_row\_selector', false);

$grid->column(trans('base.SFP'))->display(function()

{

$current\_people = People::where('id', $this->id\_People)->first();

return $current\_people->SurName." ".$current\_people->FirstName." ".$current\_people->PatronymicName;

});

$grid->column(trans('base.Result'))->display(function()

{

$event\_result = EventResult::where('id', $this->id\_EventResult)->first();

return trans('base.' . $event\_result->ResultName);

});

$grid->column('Comments', \_\_(trans('base.Content')));

$grid->column('DateTimeEvent', \_\_(trans('base.DateTimeEvent')));

return $grid;

}

/\*\*

\* Make a show builder.

\*

\* @param mixed $id

\* @return Show

\*/

protected function detail($id)

{

$show = new Show(EventsLogs::findOrFail($id));

$show->field('id', \_\_('Id'));

$show->field('id\_People', \_\_('Id People'));

$show->field('id\_EventResult', \_\_('Id EventResult'));

$show->field('Comments', \_\_('Comments'));

$show->field('DateTimeEvent', \_\_('DateTimeEvent'));

return $show;

}

/\*\*

\* Make a form builder.

\*

\* @return Form

\*/

protected function form()

{

$form = new Form(new EventsLogs());

$form->number('id\_People', \_\_('Id People'));

$form->number('id\_EventResult', \_\_('Id EventResult'));

$form->text('Comments', \_\_('Comments'));

$form->datetime('DateTimeEvent', \_\_('DateTimeEvent'))->default(date('Y-m-d H:i:s'));

return $form;

}

}

2.46 NotificationController.php

<?php

namespace App\Admin\Controllers;

use App\People;

use App\EventsLogs;

use Encore\Admin\Controllers\AdminController;

use Encore\Admin\Form;

use Encore\Admin\Grid;

use Encore\Admin\Show;

class NotificationController extends CustomAdminController

{

/\*\*

\* Title for current resource.

\*

\* @var string

\*/

protected $title = 'Notifications';

/\*\*

\* Make a grid builder.

\*

\* @return Grid

\*/

protected function grid()

{

$grid = new Grid(new EventsLogs());

$grid->disableExport();

$grid->filter(function($filter)

{

$filter->disableIdFilter();

$filter->date('DateTimeEvent', \_\_(trans('base.DateTimeEvent')));

});

$grid->disableExport();

$grid->disableCreation();

$grid->option('show\_actions', false);

$grid->option('show\_row\_selector', false);

$grid->model()->where('id\_EventType', '=', 4);

$grid->column(trans('base.SFP'))->display(function()

{

$current\_people = People::where('id', $this->id\_People)->first();

return $current\_people->SurName." ".$current\_people->FirstName." ".$current\_people->PatronymicName;

});

$grid->column('Comments', \_\_(trans('base.Content')));

$grid->column('DateTimeEvent', \_\_(trans('base.DateTimeEvent')));

return $grid;

}

/\*\*

\* Make a show builder.

\*

\* @param mixed $id

\* @return Show

\*/

protected function detail($id)

{

$show = new Show(EventsLogs::findOrFail($id));

$show->field('id', \_\_('Id'));

$show->field('id\_People', \_\_('Id People'));

$show->field('id\_EventType', \_\_('Id EventType'));

$show->field('id\_EventResult', \_\_('Id EventResult'));

$show->field('Comments', \_\_('Comments'));

$show->field('HumanOrder', \_\_('HumanOrder'));

$show->field('DateTimeEvent', \_\_('DateTimeEvent'));

return $show;

}

/\*\*

\* Make a form builder.

\*

\* @return Form

\*/

protected function form()

{

$form = new Form(new EventsLogs());

$form->number('id\_People', \_\_('Id People'));

$form->number('id\_EventType', \_\_('Id EventType'));

$form->number('id\_EventResult', \_\_('Id EventResult'));

$form->text('Comments', \_\_('Comments'));

$form->text('HumanOrder', \_\_('HumanOrder'))->default('Нет');

$form->datetime('DateTimeEvent', \_\_('DateTimeEvent'))->default(date('Y-m-d H:i:s'));

return $form;

}

}

2.47. OrganizationController.php

<?php

namespace App\Admin\Controllers;

use App\Organization;

use Encore\Admin\Controllers\AdminController;

use Encore\Admin\Form;

use Encore\Admin\Grid;

use Encore\Admin\Show;

class OrganizationController extends CustomAdminController

{

/\*\*

\* Title for current resource.

\*

\* @var string

\*/

protected $title = 'Organizations';

/\*\*

\* Make a grid builder.

\*

\* @return Grid

\*/

protected function grid()

{

$grid = new Grid(new Organization());

$grid->disableExport();

$grid->filter(function($filter)

{

$filter->disableIdFilter();

$filter->like('Name', \_\_(trans('base.Name')));

});

$grid->column('id', \_\_('Id'));

$grid->column('Name', \_\_(trans('base.Name')));

$grid->column(\_\_(trans('base.Parent')))->display(function()

{

$organization = Organization::where('id', $this->Parent)->first();

if($organization != null) return $organization->Name;

else return "";

});

return $grid;

}

/\*\*

\* Make a show builder.

\*

\* @param mixed $id

\* @return Show

\*/

protected function detail($id)

{

$show = new Show(Organization::findOrFail($id));

$show->field('id', \_\_('Id'));

$show->field('Name', \_\_(trans('base.Name')));

$show->field(trans('base.Parent'))->as(function()

{

$organization = Organization::where('id', $this->Parent)->first();

if($organization != null) return $organization->Name;

else return "";

});

return $show;

}

/\*\*

\* Make a form builder.

\*

\* @return Form

\*/

protected function form()

{

$form = new Form(new Organization());

$form->text('Name', \_\_(trans('base.Name')))->rules('required');

$form->select('Parent', \_\_(trans('base.Parent')))->options(function ($id) {

$organization = Organization::all();

$options = [];

foreach($organization as $org) $options += [$org->id => $org->Name];

return $options;

});

return $form;

}

}

2.48. PeopleController.php

<?php

namespace App\Admin\Controllers;

use App\People;

use App\PeopleContact;

use App\Pincode;

use App\PeopleAccessInterim;

use App\AccessList;

use App\Organization;

use App\PeopleOrganizations;

use App\Admin\AccessGrid;

use App\Admin\OrganizationGrid;

use Encore\Admin\Form;

use Encore\Admin\Grid;

use Encore\Admin\Show;

use Encore\Admin\Layout\Content;

class PeopleController extends CustomAdminController

{

/\*\*

\* Title for current resource.

\*

\* @var string

\*/

protected $title = "Peoples";

/\*\*

\* Make a grid builder.

\*

\* @return Grid

\*/

protected function grid()

{

$grid = new Grid(new People());

$grid->disableExport();

$grid->filter(function($filter){

$filter->disableIdFilter();

$filter->like('SurName',\_\_(trans('base.SurName')));

$filter->like('FirstName', \_\_(trans('base.FirstName')));

$filter->like('PatronymicName', \_\_(trans('base.PatronymicName')));

$filter->like('Login', \_\_(trans('base.Login')));

$filter->date('DateTimeCreate', \_\_(trans('base.DateTimeCreate')));

});

$grid->column('id', \_\_('Id'));

$grid->column('SurName', \_\_(trans('base.SurName')));

$grid->column('FirstName', \_\_(trans('base.FirstName')));

$grid->column('PatronymicName', \_\_(trans('base.PatronymicName')));

$grid->column('Login', \_\_(trans('base.Login')));

$grid->column('Password', \_\_(trans('base.Password')));

$grid->column('DateTimeCreate', \_\_(trans('base.DateTimeCreate')));

$grid->column('ActivityStatus', \_\_(trans('base.ActivityStatus')));

$grid->column(trans('base.ActualPincode'))->display(function()

{

$active\_pincode = Pincode::where('id\_People', $this->id)->orderBy('id', 'desc')->first();

if($active\_pincode != null) return $active\_pincode->PinCode;

else return "";

});

$grid->column(trans('base.PinCodeGenerationTime'))->display(function()

{

$active\_pincode = Pincode::where('id\_People', $this->id)->orderBy('id', 'desc')->first();

if($active\_pincode != null) return $active\_pincode->DateTimeGen;

else return "";

});

$grid->column(trans('base.PinCodeEndTime'))->display(function()

{

$active\_pincode = Pincode::where('id\_People', $this->id)->orderBy('id', 'desc')->first();

if($active\_pincode != null) return $active\_pincode->EndTime;

else return "";

});

$grid->column(trans('base.Phonenumber'))->display(function()

{

$actual\_phone = PeopleContact::where('id\_People', $this->id)->where('id\_TypeContact', 1)->orderBy('id', 'desc')->first();

if($actual\_phone != null) return $actual\_phone->Contact;

else return "";

});

$grid->column(trans('base.Email'))->display(function()

{

$actual\_email = PeopleContact::where('id\_People', $this->id)->where('id\_TypeContact', 2)->orderBy('id', 'desc')->first();

if($actual\_email != null) return $actual\_email->Contact;

else return "";

});

$grid->column(trans('base.Organization'))->display(function()

{

$organizations = PeopleOrganizations::where('id\_People', $this->id)->get();

$result = "";

foreach($organizations as $org)

{

$OrgInfo = Organization::where('id', $org->id\_Organizations)->first();

$result .= $OrgInfo->Name . "\n";

}

return $result;

});

return $grid;

}

/\*\*

\* Make a show builder.

\*

\* @param mixed $id

\* @return Show

\*/

protected function detail($id)

{

$show = new Show(People::findOrFail($id));

$show->field('id', \_\_('Id'));

$show->field('SurName', \_\_('SurName'));

$show->field('FirstName', \_\_('FirstName'));

$show->field('PatronymicName', \_\_('PatronymicName'));

$show->field('Login', \_\_('Login'));

$show->field('Password', \_\_('Password'));

$show->field('DateTimeCreate', \_\_('DateTimeCreate'));

$show->field('ActivityStatus', \_\_('ActivityStatus'));

return $show->render() . $this->renderAccessList(true, $id)->render() . $this->renderOrganizationList(true, $id)->render();

}

/\*\*

\* Make a form builder.

\*

\* @return Form

\*/

protected function form()

{

$form = new Form(new People());

$form->text('SurName', \_\_(trans('base.SurName')))->rules('required|min:2');

$form->text('FirstName', \_\_(trans('base.FirstName')))->rules('required|min:2');

$form->text('PatronymicName', \_\_(trans('base.PatronymicName')));

$form->text('Login', \_\_(trans('base.Login')))->rules('required|min:6');

$form->password('Password', \_\_(trans('base.Password')))->rules('required|min:6');

return $form;

}

protected function renderAccessList(bool $enableCreation, $id\_people)

{

$grid = new AccessGrid(new PeopleAccessInterim());

$grid->disableExport();

$grid->model()->where('id\_People', '=', $id\_people);

$grid->option('show\_actions', false);

$grid->option('show\_row\_selector', false);

if(!$enableCreation) $grid->disableCreation();

$grid->column(trans('base.Access'))->display(function()

{

$access = AccessList::where('id', $this->id\_Access)->first();

if($access != null) return $access->Comment;

else return "Его не существует!";

});

$grid->column(trans('base.Option'))->display(function() use ($id\_people)

{

$delete = trans('admin.delete');

$url = url('admin/peoples/'. $id\_people .'/delete-access/'. $this->id);

return <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$url}" class="btn btn-sm btn-danger" title="{$delete}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$delete}</span>

</a>

</div>

EOT;

});

return $grid;

}

protected function roleList($id\_people, Content $content)

{

$grid = new Grid(new AccessList());

$grid->disableCreation();

$grid->disableExport();

$grid->option('show\_actions', false);

$grid->option('show\_row\_selector', false);

$grid->column('Comment', \_\_(trans('base.Comment')));

$grid->column(trans('base.Option'))->display(function() use ($id\_people)

{

$new = trans('admin.new');

$url = url('admin/peoples/'. $id\_people .'/add-access/'. $this->id);

return <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$url}" class="btn btn-sm btn-success" title="{$new}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$new}</span>

</a>

</div>

EOT;

});

$cancel = trans('admin.cancel');

$url = url('/admin/peoples/'. $id\_people);

$cancelButton = <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$url}" class="btn btn-sm btn-danger" title="{$cancel}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$cancel}</span>

</a>

</div>

EOT;

return $content

->title(trans('base.AddAccess'))

->body($grid->render() . $cancelButton);

}

protected function addAccess($id\_people, $id\_access)

{

$PeopleAccessInterim = new PeopleAccessInterim();

$PeopleAccessInterim->id\_People = $id\_people;

$PeopleAccessInterim->id\_Access = $id\_access;

$PeopleAccessInterim->save();

return redirect('admin/peoples/'. $id\_people);

}

protected function deleteAccess($id\_people, $id\_accessInterim)

{

$PeopleAccessInterim = PeopleAccessInterim::find($id\_accessInterim);

$PeopleAccessInterim->delete();

return redirect('admin/peoples/'. $id\_people);

}

protected function renderOrganizationList(bool $enableCreation, $id\_people)

{

$grid = new OrganizationGrid(new PeopleOrganizations());

$grid->disableExport();

$grid->model()->where('id\_People', '=', $id\_people);

$grid->option('show\_actions', false);

$grid->option('show\_row\_selector', false);

if(!$enableCreation) $grid->disableCreation();

$grid->column(trans('base.Organization'))->display(function()

{

$organization = Organization::where('id', $this->id\_Organizations)->first();

if($organization != null) return $organization->Name;

else return "Его не существует!";

});

$grid->column(trans('base.Option'))->display(function() use ($id\_people)

{

$delete = trans('admin.delete');

$url = url('admin/peoples/'. $id\_people .'/delete-organization/'. $this->id);

return <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$url}" class="btn btn-sm btn-danger" title="{$delete}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$delete}</span>

</a>

</div>

EOT;

});

return $grid;

}

protected function organizationList($id\_people, Content $content)

{

$grid = new Grid(new Organization());

$grid->disableCreation();

$grid->disableExport();

$grid->option('show\_actions', false);

$grid->option('show\_row\_selector', false);

$grid->column('Name', \_\_(trans('base.Name')));

$grid->column(trans('base.Option'))->display(function() use ($id\_people)

{

$new = trans('admin.new');

$url = url('admin/peoples/'. $id\_people .'/add-organization/'. $this->id);

return <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$url}" class="btn btn-sm btn-success" title="{$new}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$new}</span>

</a>

</div>

EOT;

});

$cancel = trans('admin.cancel');

$url = url('/admin/peoples/'. $id\_people);

$cancelButton = <<<EOT

<div class="btn-group pull-right grid-create-btn" style="margin-right: 10px">

<a href="{$url}" class="btn btn-sm btn-danger" title="{$cancel}">

<i class="fa fa-plus"></i><span class="hidden-xs">&nbsp;&nbsp;{$cancel}</span>

</a>

</div>

EOT;

return $content

->title(trans('base.AddOrganization'))

->body($grid->render() . $cancelButton);

}

protected function addOrganization($id\_people, $id\_organization)

{

$PeopleOrganizations = new PeopleOrganizations();

$PeopleOrganizations->id\_People = $id\_people;

$PeopleOrganizations->id\_Organizations = $id\_organization;

$PeopleOrganizations->save();

return redirect('admin/peoples/'. $id\_people);

}

protected function deleteOrganization($id\_people, $id\_peopleOrganization)

{

$PeopleOrganizations = PeopleOrganizations::find($id\_peopleOrganization);

$PeopleOrganizations->delete();

return redirect('admin/peoples/'. $id\_people);

}

}

2.49. ToolController.php

<?php

namespace App\Admin\Controllers;

use Encore\Admin\Controllers\AdminController;

use Encore\Admin\Layout\Content;

use Encore\Admin\Form;

use Encore\Admin\Grid;

use Encore\Admin\Show;

use GuzzleHttp\Client;

class ToolController extends AdminController

{

public function index(Content $content)

{

$generatePinCodeUrl = url('admin/tools/generate-pincode');

return $content

->title(trans('base.Tools'))

->body(view('tools', ['generatePinCodeUrl' => $generatePinCodeUrl,

'launch' => trans('base.Launch'),

'pinCodeGeneration' => trans('base.PinCodeGeneration')]));

}

public function generatePinCode(Content $content)

{

try

{

$client = new Client(['base\_uri' => 'http://lomapi.isp.regruhosting.ru/api/']);

$PeoplesJson = $client->request('GET', 'people', [

'headers' => [

'Authorization' => 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIyIiwianRpIjoiNGMwZGRkYjI3OGNhODIyMzdlYjFkMjc2M2EyMmNmYzZjZmJkOWVmYzBmNzQwMGUwYzdiMmExMzcwODI5MGEwN2Y1MjY0ZGExMzI5M2YxMWIiLCJpYXQiOjE1ODg2MDI3NDEsIm5iZiI6MTU4ODYwMjc0MSwiZXhwIjoxNjIwMTM4NzQxLCJzdWIiOiIxIiwic2NvcGVzIjpbXX0.V87VBpYbpE4uT5fdWhIiaOSCieFy5vdnFy4oMPkYI3jEL5SA1kR2Z46ZQVc93gYr0c0rl37y4LnOLuARp64zRSF7Z\_w099gNZNNVJQYPyLt1o7FawgCH74nopY2-EFUXqOkUfX7yZOVgA34awrNEXNPqWc-pLgn\_7XxCy0ySO-iQrXld5Nz1yOb3pagpVSebsML25l9hhahzc7fflo66J2bCwH-FQXSbDMJ0VqOK51OlCpagYK0X3ZsT\_8BzaI6UdIgDoFtAQIdSsjqdcEddjwe\_AI-ofK3oTwzKmLhyu63vxS8uvEjjcduprAMxp0diUP3V5QSNYHbfYnF66e-\_GzSsXJx7TH7hzE0QSimQpDp9CCmAFktbRx36EnHBv3wTNhjV3athJ4DlRU6JfwfKPjagpfDK9IIkX98AGN4ScKO6fe5QR-tydns9kjtcreb4lti5l8jF6yMvbzpVoZQ23w7v-R1V9XD92BoSE2i9hynSRnlAW2wWNI3cZIZbucRcLhbE5p9Aqa0Vs3CDBt-7jDOebPlbvZvUjkHvmEwL1P9WFEhgJLiafZS4036nP6GEus6nNrBjQMSHNK\_K\_f8nlnm4CqexJynLaORhcZsWvBl1VaH5V0SMPLA2dp2M-vZkrvAWgpwt9-H-V8OeJ2g-4io7EG6O36eaKMApKfJwV-g'

],

]);

$Peoples = json\_decode($PeoplesJson->getBody());

foreach($Peoples as $People)

{

$body = "id\_People=". $People->id ."&id\_EventType=110&HumanOrder=%u041D%u0435%u0442";

$responce = $client->request('POST', 'events-handling', [

'headers' => [

'Content-Type' => 'application/x-www-form-urlencoded',

'Authorization' => 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIyIiwianRpIjoiNGMwZGRkYjI3OGNhODIyMzdlYjFkMjc2M2EyMmNmYzZjZmJkOWVmYzBmNzQwMGUwYzdiMmExMzcwODI5MGEwN2Y1MjY0ZGExMzI5M2YxMWIiLCJpYXQiOjE1ODg2MDI3NDEsIm5iZiI6MTU4ODYwMjc0MSwiZXhwIjoxNjIwMTM4NzQxLCJzdWIiOiIxIiwic2NvcGVzIjpbXX0.V87VBpYbpE4uT5fdWhIiaOSCieFy5vdnFy4oMPkYI3jEL5SA1kR2Z46ZQVc93gYr0c0rl37y4LnOLuARp64zRSF7Z\_w099gNZNNVJQYPyLt1o7FawgCH74nopY2-EFUXqOkUfX7yZOVgA34awrNEXNPqWc-pLgn\_7XxCy0ySO-iQrXld5Nz1yOb3pagpVSebsML25l9hhahzc7fflo66J2bCwH-FQXSbDMJ0VqOK51OlCpagYK0X3ZsT\_8BzaI6UdIgDoFtAQIdSsjqdcEddjwe\_AI-ofK3oTwzKmLhyu63vxS8uvEjjcduprAMxp0diUP3V5QSNYHbfYnF66e-\_GzSsXJx7TH7hzE0QSimQpDp9CCmAFktbRx36EnHBv3wTNhjV3athJ4DlRU6JfwfKPjagpfDK9IIkX98AGN4ScKO6fe5QR-tydns9kjtcreb4lti5l8jF6yMvbzpVoZQ23w7v-R1V9XD92BoSE2i9hynSRnlAW2wWNI3cZIZbucRcLhbE5p9Aqa0Vs3CDBt-7jDOebPlbvZvUjkHvmEwL1P9WFEhgJLiafZS4036nP6GEus6nNrBjQMSHNK\_K\_f8nlnm4CqexJynLaORhcZsWvBl1VaH5V0SMPLA2dp2M-vZkrvAWgpwt9-H-V8OeJ2g-4io7EG6O36eaKMApKfJwV-g'

],

'body' => $body

]);

}

admin\_success('Успех!', 'Пин-коды были сгенерированы успешно!');

}

catch (Exception $e)

{

admin\_error('Провал!', 'Что-то пошло не так =( '. $e->getMessage());

}

return redirect(url('admin/tools'));

}

}

2.50. BaseController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller as Controller;

class BaseController extends Controller

{

/\*\*

\* success response method.

\*

\* @return \Illuminate\Http\Response

\*/

public function sendResponse($result, $message)

{

return response()->json($result, 200);

}

/\*\*

\* return error response.

\*

\* @return \Illuminate\Http\Response

\*/

public function sendError($error, $errorMessages = [], $code = 404)

{

$response = [

'success' => false,

'message' => $error,

];

if(!empty($errorMessages)){

$response['data'] = $errorMessages;

}

return response()->json($response, $code);

}

}

2.51. EventsHandlingController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\BaseController as BaseController;

use App\EventsHandling;

use Validator;

class EventsHandlingController extends BaseController

{

public function index()

{

$eventsHandling = EventsHandling::all();

return $this->sendResponse($eventsHandling->toArray(), 'EventsHandlng retrieved successfully.');

}

public function store(Request $request)

{

$input = $request->all();

$eventsHandling = EventsHandling::create($input);

return $this->sendResponse($eventsHandling->toArray(), 'EventsHandling created successfully.');

}

}

2.52. LoginController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\BaseController as BaseController;

use App\People;

class LoginController extends BaseController

{

public function login(Request $request)

{

$input = $request->all();

$login = $input['Login'];

$password = $input['Password'];

$people = People::where([

['Login', '=', $login],

['Password', '=', $password]

])->first();

if($people != null)

{

return $this->sendResponse($people->toArray(), 'Succesfull authorization');

}

else

{

return $this->sendError('Authorization Fail');

}

}

}

2.53. PeopleController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\BaseController as BaseController;

use App\People;

class PeopleController extends BaseController

{

public function index()

{

$people = People::all();

return $this->sendResponse($people->toArray(), 'EventsHandlng retrieved successfully.');

}

public function show($id)

{

$people = People::find($id);

return $this->sendResponse($people->toArray(), 'People retrieved successfully.');

}

public function store(Request $request)

{

$input = $request->all();

try

{

$people = People::create($input);

return $this->sendResponse($people->toArray(), 'Success');

}

catch(Exceception $e)

{

return $this->sendError($e, [], 500);

}

}

}

2.54. PinCodeController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\BaseController as BaseController;

use App\PinCode;

class PinCodeController extends BaseController

{

public function index()

{

$pinCode = PinCode::all();

return $this->sendResponse($pinCode->toArray(), 'EventsHandlng retrieved successfully.');

}

public function show($id)

{

$pinCode = PinCode::where('id\_People', $id)->first();

return $this->sendResponse($pinCode->toArray(), 'People retrieved successfully.');

}

}

2.55. RegisterController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\BaseController as BaseController;

use App\User;

use Illuminate\Support\Facades\Auth;

use Validator;

class RegisterController extends BaseController

{

/\*\*

\* Register api

\*

\* @return \Illuminate\Http\Response

\*/

public function register(Request $request)

{

$validator = Validator::make($request->all(), [

'name' => 'required',

'email' => 'required|email',

'password' => 'required',

'c\_password' => 'required|same:password',

]);

if($validator->fails()){

return $this->sendError('Validation Error.', $validator->errors());

}

$input = $request->all();

$input['password'] = bcrypt($input['password']);

$user = User::create($input);

$success['token'] = $user->createToken('MyApp')->accessToken;

$success['name'] = $user->name;

return $this->sendResponse($success, 'User register successfully.');

}

}

2.56. EventsHandling.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class EventsHandling extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_EventsHandling';

/\*\*

\* @var bool

\*/

public $timestamps = false;

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array

\*/

protected $fillable = [

'id\_People', 'id\_EventType', 'HumanOrder',

];

}

2.57. People.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class People extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_People';

/\*\*

\* @var bool

\*/

public $timestamps = false;

/\*\*

\* The attributes that should be hidden for arrays.

\*

\* @var array

\*/

protected $hidden = [

'Password', 'DateTimeCreate', 'ActivityStatus',

];

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array

\*/

protected $fillable = [

'SurName', 'FirstName', 'PatronymicName', 'Login', 'Password',

];

}

2.58. PinCode.php

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class PinCode extends Model

{

/\*\*

\* Таблица, связанная с моделью.

\*

\* @var string

\*/

protected $table = 'Lom\_PinCode';

/\*\*

\* @var bool

\*/

public $timestamps = false;

}

2.59. User.php

<?php

namespace App;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

use Laravel\Passport\HasApiTokens;

class User extends Authenticatable implements MustVerifyEmail

{

use HasApiTokens, Notifiable;

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array

\*/

protected $fillable = [

'name', 'email', 'password',

];

/\*\*

\* The attributes that should be hidden for arrays.

\*

\* @var array

\*/

protected $hidden = [

'password', 'remember\_token',

];

/\*\*

\* The attributes that should be cast to native types.

\*

\* @var array

\*/

protected $casts = [

'email\_verified\_at' => 'datetime',

# ПРИЛОЖЕНИЕ Б. СКРИПТ БАЗЫ ДАННЫХ

АННОТАЦИЯ

В приложении скрипт базы данных, продемонстрирован скрипт базы данных.

1. Скрипт базы данных

-- phpMyAdmin SQL Dump

-- version 5.0.2

-- https://www.phpmyadmin.net/

--

-- Хост: localhost

-- Время создания: Май 14 2020 г., 13:25

-- Версия сервера: 5.7.23-24

-- Версия PHP: 7.1.32

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- База данных: `u0828948\_lom`

--

CREATE DATABASE IF NOT EXISTS `u0828948\_lom` DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;

USE `u0828948\_lom`;

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

--

-- Структура таблицы `admin\_menu`

--

CREATE TABLE `admin\_menu` (

`id` int(10) UNSIGNED NOT NULL,

`parent\_id` int(11) NOT NULL DEFAULT '0',

`order` int(11) NOT NULL DEFAULT '0',

`title` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`icon` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`uri` varchar(50) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`permission` varchar(255) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Структура таблицы `admin\_operation\_log`

--

CREATE TABLE `admin\_operation\_log` (

`id` int(10) UNSIGNED NOT NULL,

`user\_id` int(11) NOT NULL,

`path` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`method` varchar(10) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`ip` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`input` text COLLATE utf8mb4\_unicode\_ci NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Структура таблицы `admin\_permissions`

--

CREATE TABLE `admin\_permissions` (

`id` int(10) UNSIGNED NOT NULL,

`name` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`slug` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`http\_method` varchar(255) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`http\_path` text COLLATE utf8mb4\_unicode\_ci,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `admin\_permissions`

--

INSERT INTO `admin\_permissions` (`id`, `name`, `slug`, `http\_method`, `http\_path`, `created\_at`, `updated\_at`) VALUES

(1, 'All permission', '\*', '', '\*', NULL, NULL),

(2, 'Dashboard', 'dashboard', 'GET', '/', NULL, NULL),

(3, 'Login', 'auth.login', '', '/auth/login\r\n/auth/logout', NULL, NULL),

(4, 'User setting', 'auth.setting', 'GET,PUT', '/auth/setting', NULL, NULL),

(5, 'Auth management', 'auth.management', '', '/auth/roles\r\n/auth/permissions\r\n/auth/menu\r\n/auth/logs', NULL, NULL);

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

--

-- Структура таблицы `admin\_roles`

--

CREATE TABLE `admin\_roles` (

`id` int(10) UNSIGNED NOT NULL,

`name` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`slug` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `admin\_roles`

--

INSERT INTO `admin\_roles` (`id`, `name`, `slug`, `created\_at`, `updated\_at`) VALUES

(1, 'Administrator', 'administrator', '2020-03-05 08:59:41', '2020-03-05 08:59:41');

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

--

-- Структура таблицы `admin\_role\_menu`

--

CREATE TABLE `admin\_role\_menu` (

`role\_id` int(11) NOT NULL,

`menu\_id` int(11) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `admin\_role\_menu`

--

INSERT INTO `admin\_role\_menu` (`role\_id`, `menu\_id`, `created\_at`, `updated\_at`) VALUES

(1, 2, NULL, NULL),

(1, 9, NULL, NULL);

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

--

-- Структура таблицы `admin\_role\_permissions`

--

CREATE TABLE `admin\_role\_permissions` (

`role\_id` int(11) NOT NULL,

`permission\_id` int(11) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `admin\_role\_permissions`

--

INSERT INTO `admin\_role\_permissions` (`role\_id`, `permission\_id`, `created\_at`, `updated\_at`) VALUES

(1, 1, NULL, NULL);

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

--

-- Структура таблицы `admin\_role\_users`

--

CREATE TABLE `admin\_role\_users` (

`role\_id` int(11) NOT NULL,

`user\_id` int(11) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `admin\_role\_users`

--

INSERT INTO `admin\_role\_users` (`role\_id`, `user\_id`, `created\_at`, `updated\_at`) VALUES

(1, 1, NULL, NULL),

(1, 2, NULL, NULL);

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

--

-- Структура таблицы `admin\_users`

--

CREATE TABLE `admin\_users` (

`id` int(10) UNSIGNED NOT NULL,

`username` varchar(190) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`password` varchar(60) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`name` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`avatar` varchar(255) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`remember\_token` varchar(100) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `admin\_users`

--

INSERT INTO `admin\_users` (`id`, `username`, `password`, `name`, `avatar`, `remember\_token`, `created\_at`, `updated\_at`) VALUES

(1, 'admin', '$2y$10$CWWWJneIt4gQGG9nBU70BeQu81hCFV1SwzJgRTlJ8GqVB6FLsZXXu', 'Administrator', NULL, 'nvqGXsdV6C6OwspE6xbVS3qTFceRogQ71axSz2yOkC2wjtbtyNHzMTCKTMXK', '2020-03-05 08:59:41', '2020-03-05 08:59:41'),

(2, 'Alexey', '$2y$10$XkcnFX.dnERNExN.fDPLpesmylNQ/vj9VPUKnfMTUqPbEMpdh7WBa', 'Bushin', NULL, NULL, '2020-03-05 09:44:05', '2020-03-05 09:44:05');

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

--

-- Структура таблицы `admin\_user\_permissions`

--

CREATE TABLE `admin\_user\_permissions` (

`user\_id` int(11) NOT NULL,

`permission\_id` int(11) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `admin\_user\_permissions`

--

INSERT INTO `admin\_user\_permissions` (`user\_id`, `permission\_id`, `created\_at`, `updated\_at`) VALUES

(2, 1, NULL, NULL);

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

--

-- Структура таблицы `Lom\_AccessList`

--

CREATE TABLE `Lom\_AccessList` (

`id` int(11) NOT NULL,

`Access` varchar(254) NOT NULL COMMENT 'Указывается доступ, который будет привязан к определенному человеку в таблице «PeopleAccessInterim»',

`Comment` varchar(254) NOT NULL COMMENT 'Краткое описание доступа.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_AccessList`

--

INSERT INTO `Lom\_AccessList` (`id`, `Access`, `Comment`) VALUES

(1, 'AllAccess', 'Полный Доступ'),

(2, 'A', 'Доступ А'),

(3, 'B', 'Доступ В'),

(4, 'C', 'Доступ С'),

(5, 'D', 'Доступ D'),

(6, 'Taxi/Visitors', 'Доступ для Гостей/Такси');

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

--

-- Структура таблицы `Lom\_ConfigTable`

--

CREATE TABLE `Lom\_ConfigTable` (

`id` int(11) NOT NULL,

`Value` varchar(254) NOT NULL COMMENT 'Указывается в секундах. Данное значение используется триггером для выполнения необходимых условий или задач.',

`Comment` varchar(254) NOT NULL COMMENT 'Описывается, для чего используется данное значение.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_ConfigTable`

--

INSERT INTO `Lom\_ConfigTable` (`id`, `Value`, `Comment`) VALUES

(1, '1.2.0.20', 'Версия Базы Данных'),

(2, '2160000', 'Срок хранения логов (Указывается в секундах).'),

(10, '4', 'Длина Пин-Кода (Кол-во символов, Макс 10).'),

(11, '1', 'Время, для повторной генерации Пин-Кода (Указывается в секундах).'),

(110, '86400', 'Срок жизни Пин-Кода \"СКУД\" (Указывается в секундах).'),

(111, '1', 'Количество проходов по одному пин-коду для \"CKYD\"'),

(120, '120', 'Срок жизни Пин-Кода \"ГРЗ\" (Указывается в секундах).'),

(121, '2', 'Количество проходов по одному пин-коду для \"ГРЗ\"'),

(130, '180', 'Срок жизни Пин-Кода \"Такси/Посетитель\" (Указывается в секундах).'),

(131, '3', 'Количество проходов по одному пин-коду для \"Такси/Посетитель\"'),

(140, '240', 'Срок жизни Пин-Кода \"Разовый\" (Указывается в секундах).'),

(141, '4', 'Количество проходов по одному пин-коду для \"Разовый\"');

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

--

-- Структура таблицы `Lom\_EventResult`

--

CREATE TABLE `Lom\_EventResult` (

`id` int(11) NOT NULL,

`ResultName` varchar(254) NOT NULL COMMENT 'Указываются возможные варианты причин выполнения или невыполнения определенного события.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_EventResult`

--

INSERT INTO `Lom\_EventResult` (`id`, `ResultName`) VALUES

(1, 'Success'),

(2, 'Error');

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

--

-- Структура таблицы `Lom\_EventsHandling`

--

CREATE TABLE `Lom\_EventsHandling` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Указывается пользователь, который вызвал событие.',

`id\_EventType` int(11) NOT NULL COMMENT 'Указывается какое событие вызвал пользователь в приложении.',

`HumanOrder` varchar(254) NOT NULL DEFAULT 'Нет' COMMENT 'Используется, если пользователь заказал ГРЗ в приложении.',

`DateTimeEvent` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT 'Дата/время вызова события.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_EventsHandling`

--

-- Триггеры `Lom\_EventsHandling`

--

DELIMITER $$

CREATE TRIGGER `Lom\_EventsHandlingTr` AFTER INSERT ON `Lom\_EventsHandling` FOR EACH ROW BEGIN

-- BEGIN

-- ConfigTable Open

-- Обозначение (Переменных) id каждого класса Пин-кода Open

SET@PinCodeGenTime\_ID = 11; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_CKYD\_ID = 110; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_GRZ\_ID = 120; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_Taxi\_Visitor\_ID = 130; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_OneOff\_ID = 140;-- Lom\_ConfigTable и Lom\_EventType

-- Обозначение (Переменных) id каждого класса Пин-кода Close

SET@PinCodeGenTime = (SELECT Lom\_ConfigTable.Value FROM Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeGenTime\_ID); -- Переменная хранит в себе время, через которое можно повторно запросить генерацию Пин-Кода.

-- ConfigTable Close

-- Автоматическое удаление логов по времени указывается в ConfigTable Open

-- SET@TimeForDelete\_ID = 2; -- Указывается ID времени в таблице "ConfigTable"

-- SET@TimeForDelete\_Value = (SELECT Lom\_ConfigTable.Value FROM Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @TimeForDelete\_ID);

-- DELETE FROM Lom\_EventsLogs WHERE Lom\_EventsLogs.DateTimeEvent <= DATE\_SUB(NOW(), INTERVAL @TimeForDelete\_Value SECOND); -- Удаляет все записи из таблицы "EventsLogs", которые были созданы определенное количество дней назад.

-- DELETE FROM Lom\_OrderCar WHERE Lom\_OrderCar.DateTimeOrder <= DATE\_SUB(NOW(), INTERVAL @TimeForDelete\_Value SECOND); -- Удаляет все записи из таблицы "OrderCar", которые были созданы определенное количество дней назад.

-- -- Автоматическое удаление логов по времени указывается в ConfigTable Close

-- Начало IF SELECT People.ActivityStatus Open

-- Условие проверяет активность пользователя, который вызвал событие Open

IF ((SELECT Lom\_People.ActivityStatus FROM Lom\_People WHERE Lom\_People.id = new.id\_People) = 1) THEN -- 1 активен, 0 не активен

-- Проверка наличия прав доступа у человека Open

IF EXISTS (SELECT Lom\_AccessList.Access FROM Lom\_AccessList INNER JOIN Lom\_PeopleAccessInterim WHERE Lom\_AccessList.id = Lom\_PeopleAccessInterim.id\_Access AND Lom\_PeopleAccessInterim.id\_People = new.id\_People) THEN

-- Генерация Пин-Кода CKYD Open

IF ((SELECT new.id\_EventType FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.id = new.id) = @PinCodeGen\_CKYD\_ID) THEN

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '1', -- Success

Lom\_EventsLogs.Comments = 'Генерация Пин-Кода CKYD',

Lom\_EventsLogs.HumanOrder = new.HumanOrder;

END IF;

-- Генерация Пин-Кода CKYD Close

-- Генерация Пин-Кода GRZ Open

IF ((SELECT new.id\_EventType FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.id = new.id) = @PinCodeGen\_GRZ\_ID) THEN

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '1', -- Success

Lom\_EventsLogs.Comments = 'Заказ ГРЗ',

Lom\_EventsLogs.HumanOrder = new.HumanOrder;

END IF;

-- Генерация Пин-Кода GRZ Close

-- Генерация Пин-Кода Taxi Open

IF ((SELECT new.id\_EventType FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.id = new.id) = @PinCodeGen\_Taxi\_Visitor\_ID) THEN

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '1', -- Success

Lom\_EventsLogs.Comments = 'Заказ Такси/Посетителя(ей)',

Lom\_EventsLogs.HumanOrder = new.HumanOrder;

END IF;

-- Генерация Пин-Кода Taxi Close

-- Генерация Пин-Кода OneOff Open

IF ((SELECT new.id\_EventType FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.id = new.id) = @PinCodeGen\_OneOff\_ID) THEN

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '1', -- Success

Lom\_EventsLogs.Comments = 'Генерация Пин-Кода OneOff',

Lom\_EventsLogs.HumanOrder = new.HumanOrder;

END IF;

-- Генерация Пин-Кода OneOff Close

ELSE

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '2', -- Error

Lom\_EventsLogs.Comments = 'У пользователя нет прав доступа',

Lom\_EventsLogs.HumanOrder = new.HumanOrder;

END IF;

-- Проверка наличия прав доступа у человека Close

-- Вход в аккаунт Open

IF ((SELECT new.id\_EventType FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.id = new.id) = '1') THEN -- 1 LogIn

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '1', -- 1 Success

Lom\_EventsLogs.Comments = 'Вошел в аккаунт';

END IF;

-- Вход в аккаунт Close

-- Выход из аккаунта Open

IF ((SELECT new.id\_EventType FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.id = new.id) = '2') THEN -- LogOut

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '1', -- Success

Lom\_EventsLogs.Comments = 'Вышел из аккаунта';

END IF;

-- Выход из аккаунта Close

ELSE -- Выполняется, если учетная запись пользователя не активна.

-- Регистрация Open

IF ((SELECT new.id\_EventType FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.id = new.id) = '4') THEN -- 4 Register

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '1', -- 1 Success

Lom\_EventsLogs.Comments = 'Регистрация нового пользователя';

-- Регистрация Close

ELSE

INSERT INTO Lom\_EventsLogs SET

Lom\_EventsLogs.id\_People = new.id\_People,

Lom\_EventsLogs.id\_EventType = new.id\_EventType,

Lom\_EventsLogs.id\_EventResult = '2', -- Error

Lom\_EventsLogs.Comments = 'Учетная запись не активна',

Lom\_EventsLogs.HumanOrder = new.HumanOrder;

END IF;

END IF;

-- Конец IF SELECT People.ActivityStatus Close

-- END

END

$$

DELIMITER ;

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

--

-- Структура таблицы `Lom\_EventsLogs`

--

CREATE TABLE `Lom\_EventsLogs` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Заполняется триггером: EventsHandlingTr из таблицы «EventsHandling», указывается пользователь, который вызвал событие.',

`id\_EventType` int(11) NOT NULL COMMENT 'Заполняется триггером: EventsHandlingTr из таблицы «EventsHandling», указывается какое событие вызвал пользователь в приложении.',

`id\_EventResult` int(11) NOT NULL COMMENT 'Заполняется триггером: EventsHandlingTr из таблицы «EventsHandling», указывается, будет ли выполнено или не выполнено определенное событие.',

`Comments` varchar(254) NOT NULL COMMENT 'EventsHandlingTr из таблицы «EventsHandling», и представляет собой краткое описание события.',

`HumanOrder` varchar(254) NOT NULL DEFAULT 'Нет' COMMENT 'Заполняется триггером: EventsHandlingTr из таблицы «EventsHandling», используется, если человек заказал ГРЗ в приложении.',

`DateTimeEvent` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT 'Дата/время вызова события.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_EventsLogs`

--

--

-- Триггеры `Lom\_EventsLogs`

--

DELIMITER $$

CREATE TRIGGER `Lom\_EventsLogsTr` BEFORE INSERT ON `Lom\_EventsLogs` FOR EACH ROW BEGIN

-- BEGIN

-- ConfigTable Open

-- Обозначение (Переменных) id каждого класса Пин-кода Open

SET@ConfigTablePin\_Length\_ID = 10; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_CKYD\_ID = 110; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_GRZ\_ID = 120; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_Taxi\_Visitor\_ID = 130; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeGen\_OneOff\_ID = 140;-- Lom\_ConfigTable и Lom\_EventType

-- Обозначение (Переменных) id каждого класса Пин-кода Close

-- Обозначение (Переменных) id каждого класса Пин-кода (Количество использований одного пин кода) Open

SET@PinCodeNumbUses\_CKYD\_ID = 111; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeNumbUses\_GRZ\_ID = 121; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeNumbUses\_Taxi\_Visitor\_ID = 131; -- Lom\_ConfigTable и Lom\_EventType

SET@PinCodeNumbUses\_OneOff\_ID = 141;-- Lom\_ConfigTable и Lom\_EventType

-- Обозначение (Переменных) id каждого класса Пин-кода (Количество использований одного пин кода) Close

-- Срок жизни Пин-кода Open

SET@LifePinCode\_CKYD = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeGen\_CKYD\_ID); -- CKYD

SET@LifePinCode\_GRZ = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeGen\_GRZ\_ID); -- GRZ

SET@LifePinCode\_Taxi\_Visitor = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeGen\_Taxi\_Visitor\_ID); -- Taxi\_Visitor

SET@LifePinCode\_OneOff = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeGen\_OneOff\_ID); -- OneOff

-- Срок жизни Пин-кода Close

-- Количество использований одного пин кода Open

SET@NumbUsesPinCode\_CKYD = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeNumbUses\_CKYD\_ID); -- CKYD

SET@NumbUsesPinCode\_GRZ = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeNumbUses\_GRZ\_ID); -- GRZ

SET@NumbUsesPinCode\_Taxi\_Visitor = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeNumbUses\_Taxi\_Visitor\_ID); -- Taxi\_Visitor

SET@NumbUsesPinCode\_OneOff = (SELECT Lom\_ConfigTable.Value from Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @PinCodeNumbUses\_OneOff\_ID); -- OneOff

-- Количество использований одного пин кода Close

-- Уровень достпуа для Гостей/Такси Open

SET@Access\_Taxi\_Visitor\_ID = 6; -- Lom\_AccessList

SET@Access\_Taxi\_Visitor = (SELECT Lom\_AccessList.Access from Lom\_AccessList WHERE Lom\_AccessList.id = @Access\_Taxi\_Visitor\_ID); -- Taxi\_Visitor

-- Уровень достпуа для Гостей/Такси Close

-- ConfigTable Close

-- Автоматическое удаление логов по времени указывается в ConfigTable Open

-- SET@TimeForDelete\_ID = 2; -- Указывается ID времени в таблице "ConfigTable"

-- SET@TimeForDelete\_Value = (SELECT Lom\_ConfigTable.Value FROM Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @TimeForDelete\_ID);

-- DELETE FROM Lom\_EventsHandling WHERE Lom\_EventsHandling.DateTimeEvent <= DATE\_SUB(NOW(), INTERVAL @TimeForDelete\_Value SECOND); -- Удаляет все записи из таблицы "EventsHandling", которые были созданы определенное количество дней назад.

-- Автоматическое удаление логов по времени указывается в ConfigTable Close

-- Удаление Пин-Кодов Open

IF (new.id\_EventType = @PinCodeGen\_CKYD\_ID) THEN -- CKYD

DELETE FROM Lom\_PinCode WHERE Lom\_PinCode.id\_People = new.id\_People AND Lom\_PinCode.id\_PinCodeType = @PinCodeGen\_CKYD\_ID;

END IF;

IF (new.id\_EventType = @PinCodeGen\_Taxi\_Visitor\_ID) THEN -- Taxi\_Visitor

DELETE FROM Lom\_PinCode WHERE Lom\_PinCode.id\_People = new.id\_People AND Lom\_PinCode.id\_PinCodeType = @PinCodeGen\_Taxi\_Visitor\_ID;

END IF;

IF (new.id\_EventType = @PinCodeGen\_OneOff\_ID) THEN -- OneOff

DELETE FROM Lom\_PinCode WHERE Lom\_PinCode.id\_People = new.id\_People AND Lom\_PinCode.id\_PinCodeType = @PinCodeGen\_OneOff\_ID;

END IF;

-- DELETE FROM Lom\_PinCode WHERE Lom\_PinCode.EndTime < NOW(); -- Удаление Пин-Кодов, у которых истех срок действия.

-- DELETE FROM Lom\_PinCode WHERE Lom\_PinCode.id\_People = new.id\_People;

-- Удаление Пин-Кодов Close

-- Генерация Пин-Кода Open

IF (((new.id\_EventType = @PinCodeGen\_CKYD\_ID) OR (new.id\_EventType = @PinCodeGen\_GRZ\_ID) OR (new.id\_EventType = @PinCodeGen\_Taxi\_Visitor\_ID) OR (new.id\_EventType = @PinCodeGen\_OneOff\_ID)) AND (new.id\_EventResult <> 2)) THEN

SET@PinCode = SUBSTRING((ROUND(RAND()\*(9999999999-1000000000)+1000000000)), 1,(SELECT Lom\_ConfigTable.Value FROM Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @ConfigTablePin\_Length\_ID)); -- Генерация Пин-Кода 10 симв, и возвращение определенного количества символов.

IF((new.id\_EventType = @PinCodeGen\_Taxi\_Visitor\_ID) AND (new.HumanOrder > 1)) THEN

SET@Num = new.HumanOrder;

ELSE

SET@Num = 1;

END IF;

WHILE @Num > 0 DO

-- Цикл необходим для поиска дублирующихся Пин-Кодов.

IF EXISTS (SELECT Lom\_PinCode.PinCode FROM Lom\_PinCode WHERE Lom\_PinCode.PinCode = @PinCode) THEN

-- Повторная генерация Пин-Кода 10 симв, и возвращение определенного количества символов (Количество символов указывается в таблице "ConfigTable").

SET@PinCode = SUBSTRING((ROUND(RAND()\*(9999999999-1000000000)+1000000000)), 1,(SELECT Lom\_ConfigTable.Value FROM Lom\_ConfigTable WHERE Lom\_ConfigTable.id = @ConfigTablePin\_Length\_ID));

ELSE -- Выполняется, если сгенерированный Пин-Код уникален.

-- Генерация Пин-Кода CKYD Open

IF (new.id\_EventType = @PinCodeGen\_CKYD\_ID) THEN

INSERT INTO Lom\_PinCode SET

Lom\_PinCode.id\_People = new.id\_People,

Lom\_PinCode.PinCode = @PinCode,

Lom\_PinCode.PinCodeNumberUses = @NumbUsesPinCode\_CKYD,

Lom\_PinCode.id\_PinCodeType = @PinCodeGen\_CKYD\_ID,

Lom\_PinCode.PinCodeAccess = (SELECT GROUP\_CONCAT(Lom\_AccessList.Access, '') FROM Lom\_AccessList INNER JOIN Lom\_PeopleAccessInterim WHERE Lom\_AccessList.id = Lom\_PeopleAccessInterim.id\_Access AND Lom\_PeopleAccessInterim.id\_People = new.id\_People),

Lom\_PinCode.EndTime = (SELECT DATE\_ADD(NOW(), INTERVAL @LifePinCode\_CKYD SECOND)); -- Указывается дата/время, когда у сгенерированного Пин-Кода истечет срок действия (срок дейтсвия Пин-Кода указывается в таблице "ConfigTable").

SET@Num = 0;

END IF;

-- Генерация Пин-Кода CKYD Close

-- Генерация Пин-Кода GRZ Open

IF (new.id\_EventType = @PinCodeGen\_GRZ\_ID) THEN

-- Заказ автомобиля

INSERT INTO Lom\_OrderCar SET

Lom\_OrderCar.id\_People = new.id\_People,

Lom\_OrderCar.CarNumber = new.HumanOrder;

-- Заказ автомобиля

-- Пин-Код

INSERT INTO Lom\_PinCode SET

Lom\_PinCode.id\_People = new.id\_People,

Lom\_PinCode.PinCode = @PinCode,

Lom\_PinCode.PinCodeNumberUses = @NumbUsesPinCode\_GRZ,

Lom\_PinCode.id\_PinCodeType = @PinCodeGen\_GRZ\_ID,

Lom\_PinCode.PinCodeAccess = (SELECT GROUP\_CONCAT(Lom\_AccessList.Access, '') FROM Lom\_AccessList INNER JOIN Lom\_PeopleAccessInterim WHERE Lom\_AccessList.id = Lom\_PeopleAccessInterim.id\_Access AND Lom\_PeopleAccessInterim.id\_People = new.id\_People),

Lom\_PinCode.EndTime = (SELECT DATE\_ADD(NOW(), INTERVAL @LifePinCode\_GRZ SECOND)); -- Указывается дата/время, когда у сгенерированного Пин-Кода истечет срок действия (срок дейтсвия Пин-Кода указывается в таблице "ConfigTable").

SET@Num = 0;

-- Пин-Код

END IF;

-- Генерация Пин-Кода GRZ Close

-- Генерация Пин-Кода Taxi\_Visitor Open

IF (new.id\_EventType = @PinCodeGen\_Taxi\_Visitor\_ID) THEN

INSERT INTO Lom\_PinCode SET

Lom\_PinCode.id\_People = new.id\_People,

Lom\_PinCode.PinCode = @PinCode,

Lom\_PinCode.PinCodeNumberUses = @NumbUsesPinCode\_Taxi\_Visitor,

Lom\_PinCode.id\_PinCodeType = @PinCodeGen\_Taxi\_Visitor\_ID,

Lom\_PinCode.PinCodeAccess = @Access\_Taxi\_Visitor,

Lom\_PinCode.EndTime = (SELECT DATE\_ADD(NOW(), INTERVAL @LifePinCode\_Taxi\_Visitor SECOND)); -- Указывается дата/время, когда у сгенерированного Пин-Кода истечет срок действия (срок дейтсвия Пин-Кода указывается в таблице "ConfigTable").

SET @Num = @Num - 1;

END IF;

-- Генерация Пин-Кода Taxi\_Visitor Close

-- Генерация Пин-Кода OneOff Open

IF (new.id\_EventType = @PinCodeGen\_OneOff\_ID) THEN

INSERT INTO Lom\_PinCode SET

Lom\_PinCode.id\_People = new.id\_People,

Lom\_PinCode.PinCode = @PinCode,

Lom\_PinCode.PinCodeNumberUses = @NumbUsesPinCode\_OneOff,

Lom\_PinCode.id\_PinCodeType = @PinCodeGen\_OneOff\_ID,

Lom\_PinCode.PinCodeAccess = (SELECT GROUP\_CONCAT(Lom\_AccessList.Access, '') FROM Lom\_AccessList INNER JOIN Lom\_PeopleAccessInterim WHERE Lom\_AccessList.id = Lom\_PeopleAccessInterim.id\_Access AND Lom\_PeopleAccessInterim.id\_People = new.id\_People),

Lom\_PinCode.EndTime = (SELECT DATE\_ADD(NOW(), INTERVAL @LifePinCode\_OneOff SECOND)); -- Указывается дата/время, когда у сгенерированного Пин-Кода истечет срок действия (срок дейтсвия Пин-Кода указывается в таблице "ConfigTable").

SET@Num = 0;

END IF;

-- Генерация Пин-Кода OneOff Close

END IF;

END WHILE;

END IF;

-- Генерация Пин-Кода Close

-- END

END

$$

DELIMITER ;

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

--

-- Структура таблицы `Lom\_EventType`

--

CREATE TABLE `Lom\_EventType` (

`id` int(11) NOT NULL,

`TypeName` varchar(254) NOT NULL COMMENT 'Указывается тип события, который может вызвать пользователь в приложении.',

`Comment` varchar(254) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_EventType`

--

INSERT INTO `Lom\_EventType` (`id`, `TypeName`, `Comment`) VALUES

(1, 'LogIn', 'Событие входа пользователя в приложение.'),

(2, 'LogOut', 'Событие выхода пользователя из приложения.'),

(4, 'Register', 'Событие регистрации нового пользователя.'),

(110, 'PinCodeGen\_CKYD', 'Событие генерации пин-кода СКУД.'),

(120, 'PinCodeGen\_GRZ', 'Событие генерации пин-кода ГРЗ.'),

(130, 'PinCodeGen\_Taxi\_Visitor', 'Событие генерации пин-кода для Такси/Посетителей.'),

(140, 'PinCodeGen\_OneOff', 'Событие генерации пин-кода Разового');

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

--

-- Структура таблицы `Lom\_IMEICode`

--

CREATE TABLE `Lom\_IMEICode` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Указывается пользователь, к которому привязан IMEI.',

`IMEI` varchar(254) NOT NULL COMMENT 'Указывается IMEI код телефона пользователя. IMEI необходим для предотвращения кражи связки логин/пароль, а также вход в аккаунт с другого устройства.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

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

--

-- Структура таблицы `Lom\_OrderCar`

--

CREATE TABLE `Lom\_OrderCar` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Заполняется триггером: EventsLogsTr из таблицы «EventsLogs», указывается человек, который заказал ГРЗ.',

`CarNumber` varchar(254) NOT NULL COMMENT 'Заполняется триггером: EventsLogsTr из таблицы «EventsLogs», указывается номер автомобиля.',

`DateTimeOrder` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT 'Дата/время заказа ГРЗ.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

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

--

-- Структура таблицы `Lom\_Organizations`

--

CREATE TABLE `Lom\_Organizations` (

`id` int(11) NOT NULL,

`Name` varchar(254) NOT NULL COMMENT 'Указывается название организации, к которой относятся работники',

`Parent` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_Organizations`

--

INSERT INTO `Lom\_Organizations` (`id`, `Name`, `Parent`) VALUES

(3, 'ООО Артсек', NULL),

(4, 'ООО ХлебЗавод', 3);

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

--

-- Структура таблицы `Lom\_People`

--

CREATE TABLE `Lom\_People` (

`id` int(11) NOT NULL,

`SurName` varchar(254) NOT NULL COMMENT 'Фамилия пользователя.',

`FirstName` varchar(254) NOT NULL COMMENT 'Имя пользователя.',

`PatronymicName` varchar(254) DEFAULT NULL COMMENT 'Отчество пользователя.',

`Login` varchar(1000) NOT NULL COMMENT 'Логин пользователя.',

`Password` varchar(1000) NOT NULL COMMENT 'Пароль пользователя.',

`DateTimeCreate` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT 'Дата/время создания аккаунта.',

`ActivityStatus` varchar(254) NOT NULL DEFAULT '1' COMMENT 'Определяется активность аккаунта пользователя. Если аккаунт не будет активирован, тогда пользователь не сможет выполнять какие-либо действия в приложении.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_People`

--

INSERT INTO `Lom\_People` (`id`, `SurName`, `FirstName`, `PatronymicName`, `Login`, `Password`, `DateTimeCreate`, `ActivityStatus`) VALUES

(1, 'Meschersky', 'Apollo', 'Abramovich', 'app', 'app', '2020-03-19 08:34:35', '1'),

(2, 'Buldakov', 'Miroslav', 'Dobrynich', '2', '2', '2020-03-19 08:34:35', '1'),

(3, 'Ivanov', 'Ivan', 'Ivanovich', '3', '3', '2020-03-19 08:35:07', '1'),

(6, 'Юзеров', 'Юзер', 'Юзеривич', 'User', '12', '2020-03-24 13:36:51', '0'),

(7, 'Денисов', 'Андрей', 'Александрович', 'GAD', 'qwerty!1Q', '2020-03-24 14:30:41', '1'),

(8, 'Балашов', 'Михаил', 'Борисович', 'Bloha', 'Bloha87', '2020-03-24 14:31:05', '1'),

(9, 'Данюшкин', 'Владислав', 'Александрович', 'vda', 'staples', '2020-03-24 14:31:33', '1'),

(10, 'Pimakhov', 'Anton', NULL, 'mord-sit', '3141592765', '2020-03-24 14:31:55', '1'),

(11, 'Смолькин', 'Сергей', 'Александрович', 'ssm', 'qwerty', '2020-03-24 14:40:52', '1'),

(14, 'Бушин', 'Алексей', 'Юрьевич', 'Bushin', '123456', '2020-05-08 14:07:17', '1'),

(15, 'Абдурханян', 'Джабун', NULL, 'Jabuno', '123456', '2020-05-08 14:12:31', '1'),

(16, 'Бушин', 'Алексей', 'Юрьевич', 'Bushin2', '123456', '2020-05-10 00:21:06', '1'),

(17, 'йцу', 'йцу', 'йцу', 'qweqwe', '123qwe', '2020-05-13 22:04:39', '1');

--

-- Триггеры `Lom\_People`

--

DELIMITER $$

CREATE TRIGGER `Lom\_PeopleTr` AFTER INSERT ON `Lom\_People` FOR EACH ROW BEGIN

-- Обозначение (Переменных) id прав доступа Open

SET@Access\_ID = 2; -- Lom\_ConfigTable и Lom\_EventType

SET@Register\_ID = 4; -- Lom\_ConfigTable и Lom\_EventType

-- Обозначение (Переменных) id прав доступа Close

-- Событие регистрации нового пользователя Open

INSERT INTO Lom\_EventsHandling SET

Lom\_EventsHandling.id\_People = new.id,

Lom\_EventsHandling.id\_EventType = @Register\_ID;

-- Событие регистрации нового пользователя Close

-- Добавление прав доступа новому пользователю Open

INSERT INTO Lom\_PeopleAccessInterim SET

Lom\_PeopleAccessInterim.id\_People = new.id,

Lom\_PeopleAccessInterim.id\_Access = @Access\_ID;

-- Добавление прав доступа новому пользователю Close

END

$$

DELIMITER ;

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

--

-- Структура таблицы `Lom\_PeopleAccessInterim`

--

CREATE TABLE `Lom\_PeopleAccessInterim` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Указывается человек к которому будет привязан определенный доступ из таблицы «AccessList».',

`id\_Access` int(11) NOT NULL COMMENT 'Указывается уровень доступа из таблицы «AccessList».'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_PeopleAccessInterim`

--

INSERT INTO `Lom\_PeopleAccessInterim` (`id`, `id\_People`, `id\_Access`) VALUES

(1, 1, 2),

(2, 1, 3),

(3, 1, 4),

(7, 3, 5),

(8, 2, 1),

(10, 6, 2),

(11, 7, 2),

(12, 8, 2),

(13, 9, 2),

(14, 10, 2),

(15, 11, 2),

(19, 14, 2),

(20, 15, 2),

(21, 16, 2),

(22, 17, 2);

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

--

-- Структура таблицы `Lom\_PeopleContact`

--

CREATE TABLE `Lom\_PeopleContact` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Указывается пользователь, к которому будут привязаны контакты.',

`Contact` varchar(254) NOT NULL COMMENT 'Указывается контакты человека, например, номер телефона, почта.',

`id\_TypeContact` int(11) NOT NULL COMMENT 'Указывается тип контакта из таблицы «PeopleContactType».'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_PeopleContact`

--

INSERT INTO `Lom\_PeopleContact` (`id`, `id\_People`, `Contact`, `id\_TypeContact`) VALUES

(1, 7, '+79639270532', 1),

(2, 7, 'gsys@mail.ru', 2),

(3, 8, '89999876567', 1),

(4, 8, 'm.balashov@mail.ru', 2),

(5, 9, '89262045235', 1),

(6, 9, 'vda@baza-s.ru', 2),

(7, 10, '+79260966734', 1),

(8, 10, 'api@baza-s.ru', 2),

(9, 11, '89266100247', 1),

(10, 11, 'ssm@baza-s.ru', 2);

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

--

-- Структура таблицы `Lom\_PeopleContactType`

--

CREATE TABLE `Lom\_PeopleContactType` (

`id` int(11) NOT NULL,

`TypeName` varchar(254) NOT NULL COMMENT 'Указывается тип контакта, который будет использоваться в таблице «PeopleContact».'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_PeopleContactType`

--

INSERT INTO `Lom\_PeopleContactType` (`id`, `TypeName`) VALUES

(1, 'Телефон'),

(2, 'Почта');

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

--

-- Структура таблицы `Lom\_PeopleOrganizations`

--

CREATE TABLE `Lom\_PeopleOrganizations` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Указывается человек к которому привязывается организация, в которой работает.',

`id\_Organizations` int(11) NOT NULL COMMENT 'Указывается уровень организация, в которой работает человек.'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_PeopleOrganizations`

--

INSERT INTO `Lom\_PeopleOrganizations` (`id`, `id\_People`, `id\_Organizations`) VALUES

(4, 1, 3);

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

--

-- Структура таблицы `Lom\_PinCode`

--

CREATE TABLE `Lom\_PinCode` (

`id` int(11) NOT NULL,

`id\_People` int(11) NOT NULL COMMENT 'Заполняется триггером: EventsLogsTr из таблицы «EventsLogs», указывается человек, который заказал генерацию Пин-Кода.',

`PinCode` varchar(10) NOT NULL COMMENT 'Заполняется триггером «EventsLogsTr».',

`PinCodeNumberUses` int(11) NOT NULL COMMENT 'Количество проходов по одному пин-коду.',

`id\_PinCodeType` int(11) NOT NULL COMMENT 'Тип пин-кода, для чего создается.',

`PinCodeAccess` varchar(1000) NOT NULL COMMENT 'Права доступа человека, которые применяются к пин коду.',

`DateTimeGen` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT 'Дата/время генерации Пин-Кода.',

`EndTime` datetime NOT NULL COMMENT 'Дата окончания Пин-Кода вычисляется триггером «EventsLogsTr», срок жизни Пин-Кода определяется в таблице «ConfigTable».'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Lom\_PinCode`

--

INSERT INTO `Lom\_PinCode` (`id`, `id\_People`, `PinCode`, `PinCodeNumberUses`, `id\_PinCodeType`, `PinCodeAccess`, `DateTimeGen`, `EndTime`) VALUES

(1791, 2, '6447', 1, 110, 'AllAccess', '2020-05-10 00:16:56', '2020-05-11 00:16:56'),

(1792, 3, '6820', 1, 110, 'D', '2020-05-10 00:16:56', '2020-05-11 00:16:56'),

(1793, 7, '8495', 1, 110, 'A', '2020-05-10 00:16:57', '2020-05-11 00:16:57'),

(1794, 8, '4615', 1, 110, 'A', '2020-05-10 00:16:57', '2020-05-11 00:16:57'),

(1795, 9, '8609', 1, 110, 'A', '2020-05-10 00:16:57', '2020-05-11 00:16:57'),

(1796, 10, '6361', 1, 110, 'A', '2020-05-10 00:16:57', '2020-05-11 00:16:57'),

(1797, 11, '1584', 1, 110, 'A', '2020-05-10 00:16:57', '2020-05-11 00:16:57'),

(1798, 14, '1459', 1, 110, 'A', '2020-05-10 00:16:57', '2020-05-11 00:16:57'),

(1799, 15, '6573', 1, 110, 'A', '2020-05-10 00:16:57', '2020-05-11 00:16:57'),

(1804, 16, '7904', 1, 110, 'A', '2020-05-10 00:21:36', '2020-05-11 00:21:36'),

(1805, 1, '4803', 1, 110, 'A,B,C', '2020-05-10 00:22:03', '2020-05-11 00:22:03');

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

--

-- Структура таблицы `migrations`

--

CREATE TABLE `migrations` (

`id` int(10) UNSIGNED NOT NULL,

`migration` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`batch` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `migrations`

--

INSERT INTO `migrations` (`id`, `migration`, `batch`) VALUES

(1, '2014\_10\_12\_000000\_create\_users\_table', 1),

(2, '2014\_10\_12\_100000\_create\_password\_resets\_table', 1),

(3, '2016\_01\_04\_173148\_create\_admin\_tables', 1),

(4, '2016\_06\_01\_000001\_create\_oauth\_auth\_codes\_table', 2),

(5, '2016\_06\_01\_000002\_create\_oauth\_access\_tokens\_table', 2),

(6, '2016\_06\_01\_000003\_create\_oauth\_refresh\_tokens\_table', 2),

(7, '2016\_06\_01\_000004\_create\_oauth\_clients\_table', 2),

(8, '2016\_06\_01\_000005\_create\_oauth\_personal\_access\_clients\_table', 2),

(9, '2019\_08\_19\_000000\_create\_failed\_jobs\_table', 2);

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

--

-- Структура таблицы `oauth\_access\_tokens`

--

CREATE TABLE `oauth\_access\_tokens` (

`id` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`user\_id` bigint(20) UNSIGNED DEFAULT NULL,

`client\_id` bigint(20) UNSIGNED NOT NULL,

`name` varchar(255) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`scopes` text COLLATE utf8mb4\_unicode\_ci,

`revoked` tinyint(1) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL,

`expires\_at` datetime DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `oauth\_access\_tokens`

--

INSERT INTO `oauth\_access\_tokens` (`id`, `user\_id`, `client\_id`, `name`, `scopes`, `revoked`, `created\_at`, `updated\_at`, `expires\_at`) VALUES

('3de9b1e39e6d9277b8635beee167bfac17564a77877487bef68e9e8aee55c89698d2dc388a2ee02a', 1, 1, 'MyApp', '[]', 0, '2020-05-04 11:11:35', '2020-05-04 11:11:35', '2021-05-04 14:11:35'),

('4c0dddb278ca82237eb1d2763a22cfc6cfbd9efc0f7400e0c7b2a13708290a07f5264da13293f11b', 1, 2, NULL, '[]', 0, '2020-05-04 11:32:21', '2020-05-04 11:32:21', '2021-05-04 14:32:21'),

('5b90f4986e3108ef134808505b982dc8062f0922b9aa4e00651a73a81fffb18e3736d2c12a62ba71', 1, 1, 'MyApp', '[]', 0, '2020-05-04 13:04:24', '2020-05-04 13:04:24', '2021-05-04 16:04:24'),

('830f77c7bcc5407c2f64922f88104af371aaba7ac9a065e116621f6730f932c9411507620f6411a7', 1, 2, NULL, '[]', 0, '2020-05-04 13:08:28', '2020-05-04 13:08:28', '2021-05-04 16:08:28'),

('8ff303b9e5977a80f3122b34d0b3caacd499fb57dd626b8e067c1c1381f2748a2743361d067279a3', 1, 2, NULL, '[]', 0, '2020-05-04 11:17:53', '2020-05-04 11:17:53', '2021-05-04 14:17:53');

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

--

-- Структура таблицы `oauth\_auth\_codes`

--

CREATE TABLE `oauth\_auth\_codes` (

`id` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`user\_id` bigint(20) UNSIGNED NOT NULL,

`client\_id` bigint(20) UNSIGNED NOT NULL,

`scopes` text COLLATE utf8mb4\_unicode\_ci,

`revoked` tinyint(1) NOT NULL,

`expires\_at` datetime DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

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

--

-- Структура таблицы `oauth\_clients`

--

CREATE TABLE `oauth\_clients` (

`id` bigint(20) UNSIGNED NOT NULL,

`user\_id` bigint(20) UNSIGNED DEFAULT NULL,

`name` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`secret` varchar(100) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`redirect` text COLLATE utf8mb4\_unicode\_ci NOT NULL,

`personal\_access\_client` tinyint(1) NOT NULL,

`password\_client` tinyint(1) NOT NULL,

`revoked` tinyint(1) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `oauth\_clients`

--

INSERT INTO `oauth\_clients` (`id`, `user\_id`, `name`, `secret`, `redirect`, `personal\_access\_client`, `password\_client`, `revoked`, `created\_at`, `updated\_at`) VALUES

(1, NULL, 'Laravel Personal Access Client', 'v56rcIxQeOw29Sug7FuDAHOtlTPENixHJQP6TeOY', 'http://lomapi.isp.regruhosting.ru', 1, 0, 0, '2020-05-04 10:40:49', '2020-05-04 10:40:49'),

(2, NULL, 'Laravel Password Grant Client', 'KYvvMe0Fa7NaaW4pQxef3V3Ke3h2mKmYMSLXGJul', 'http://lomapi.isp.regruhosting.ru', 0, 1, 0, '2020-05-04 10:40:49', '2020-05-04 10:40:49');

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

--

-- Структура таблицы `oauth\_personal\_access\_clients`

--

CREATE TABLE `oauth\_personal\_access\_clients` (

`id` bigint(20) UNSIGNED NOT NULL,

`client\_id` bigint(20) UNSIGNED NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `oauth\_personal\_access\_clients`

--

INSERT INTO `oauth\_personal\_access\_clients` (`id`, `client\_id`, `created\_at`, `updated\_at`) VALUES

(1, 1, '2020-05-04 10:40:49', '2020-05-04 10:40:49');

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

--

-- Структура таблицы `oauth\_refresh\_tokens`

--

CREATE TABLE `oauth\_refresh\_tokens` (

`id` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`access\_token\_id` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`revoked` tinyint(1) NOT NULL,

`expires\_at` datetime DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `oauth\_refresh\_tokens`

--

INSERT INTO `oauth\_refresh\_tokens` (`id`, `access\_token\_id`, `revoked`, `expires\_at`) VALUES

('4418a1223935dd36f0e71f7afd1ff85a2ce11113b61dd520907bf6ed9e2001ba056d96bdd313e9f4', '830f77c7bcc5407c2f64922f88104af371aaba7ac9a065e116621f6730f932c9411507620f6411a7', 0, '2021-05-04 16:08:28'),

('c30064a001989519193f38d28cebf6f420e64c9459d5cd1b240fd7bcb0c4b0c1f5e2b1691ebbd765', '8ff303b9e5977a80f3122b34d0b3caacd499fb57dd626b8e067c1c1381f2748a2743361d067279a3', 0, '2021-05-04 14:17:53'),

('f6e4973d9e1a7becec2a838ade013afd5e5c671cf02ea0faef7af84d38cbc4ff37ad866ffcab05c2', '4c0dddb278ca82237eb1d2763a22cfc6cfbd9efc0f7400e0c7b2a13708290a07f5264da13293f11b', 0, '2021-05-04 14:32:21');

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

--

-- Структура таблицы `password\_resets`

--

CREATE TABLE `password\_resets` (

`email` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`token` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

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

--

-- Структура таблицы `users`

--

CREATE TABLE `users` (

`id` bigint(20) UNSIGNED NOT NULL,

`name` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`email` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`email\_verified\_at` timestamp NULL DEFAULT NULL,

`password` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`remember\_token` varchar(100) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `users`

--

INSERT INTO `users` (`id`, `name`, `email`, `email\_verified\_at`, `password`, `remember\_token`, `created\_at`, `updated\_at`) VALUES

(1, 'admin', 'admin@gmail.com', NULL, '$2y$10$AiOMFa9O2T.iySJoq2KzB.JeZJaThrqguddfyjBV5b6NkBq9ugKgy', NULL, '2020-05-04 13:04:24', '2020-05-04 13:04:24');

--

-- Индексы сохранённых таблиц

--

--

-- Индексы таблицы `admin\_menu`

--

ALTER TABLE `admin\_menu`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `admin\_operation\_log`

--

ALTER TABLE `admin\_operation\_log`

ADD PRIMARY KEY (`id`),

ADD KEY `admin\_operation\_log\_user\_id\_index` (`user\_id`);

--

-- Индексы таблицы `admin\_permissions`

--

ALTER TABLE `admin\_permissions`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `admin\_permissions\_name\_unique` (`name`),

ADD UNIQUE KEY `admin\_permissions\_slug\_unique` (`slug`);

--

-- Индексы таблицы `admin\_roles`

--

ALTER TABLE `admin\_roles`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `admin\_roles\_name\_unique` (`name`),

ADD UNIQUE KEY `admin\_roles\_slug\_unique` (`slug`);

--

-- Индексы таблицы `admin\_role\_menu`

--

ALTER TABLE `admin\_role\_menu`

ADD KEY `admin\_role\_menu\_role\_id\_menu\_id\_index` (`role\_id`,`menu\_id`);

--

-- Индексы таблицы `admin\_role\_permissions`

--

ALTER TABLE `admin\_role\_permissions`

ADD KEY `admin\_role\_permissions\_role\_id\_permission\_id\_index` (`role\_id`,`permission\_id`);

--

-- Индексы таблицы `admin\_role\_users`

--

ALTER TABLE `admin\_role\_users`

ADD KEY `admin\_role\_users\_role\_id\_user\_id\_index` (`role\_id`,`user\_id`);

--

-- Индексы таблицы `admin\_users`

--

ALTER TABLE `admin\_users`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `admin\_users\_username\_unique` (`username`);

--

-- Индексы таблицы `admin\_user\_permissions`

--

ALTER TABLE `admin\_user\_permissions`

ADD KEY `admin\_user\_permissions\_user\_id\_permission\_id\_index` (`user\_id`,`permission\_id`);

--

-- Индексы таблицы `Lom\_AccessList`

--

ALTER TABLE `Lom\_AccessList`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `AccessUNIQUE` (`Access`),

ADD KEY `AccessINDES` (`Access`);

--

-- Индексы таблицы `Lom\_ConfigTable`

--

ALTER TABLE `Lom\_ConfigTable`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `Lom\_EventResult`

--

ALTER TABLE `Lom\_EventResult`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `Lom\_EventsHandling`

--

ALTER TABLE `Lom\_EventsHandling`

ADD PRIMARY KEY (`id`),

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE,

ADD KEY `EventTypeINDEX` (`id\_EventType`) USING BTREE;

--

-- Индексы таблицы `Lom\_EventsLogs`

--

ALTER TABLE `Lom\_EventsLogs`

ADD PRIMARY KEY (`id`),

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE,

ADD KEY `EventTypeINDEX` (`id\_EventType`) USING BTREE,

ADD KEY `EventResultINDEX` (`id\_EventResult`) USING BTREE;

--

-- Индексы таблицы `Lom\_EventType`

--

ALTER TABLE `Lom\_EventType`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `TypeNameUNIQUE` (`TypeName`);

--

-- Индексы таблицы `Lom\_IMEICode`

--

ALTER TABLE `Lom\_IMEICode`

ADD PRIMARY KEY (`id`),

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE;

--

-- Индексы таблицы `Lom\_OrderCar`

--

ALTER TABLE `Lom\_OrderCar`

ADD PRIMARY KEY (`id`),

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE;

--

-- Индексы таблицы `Lom\_Organizations`

--

ALTER TABLE `Lom\_Organizations`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `OrgNameUNIQUE` (`Name`),

ADD KEY `OrgNameINDEX` (`Name`),

ADD KEY `Parent` (`Parent`);

--

-- Индексы таблицы `Lom\_People`

--

ALTER TABLE `Lom\_People`

ADD PRIMARY KEY (`id`),

ADD KEY `SurNameFirstNameINDEX` (`SurName`,`FirstName`),

ADD KEY `SurNameINDEX` (`SurName`);

--

-- Индексы таблицы `Lom\_PeopleAccessInterim`

--

ALTER TABLE `Lom\_PeopleAccessInterim`

ADD PRIMARY KEY (`id`),

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE,

ADD KEY `AccessINDEX` (`id\_Access`) USING BTREE;

--

-- Индексы таблицы `Lom\_PeopleContact`

--

ALTER TABLE `Lom\_PeopleContact`

ADD PRIMARY KEY (`id`),

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE,

ADD KEY `TypeContactINDEX` (`id\_TypeContact`) USING BTREE,

ADD KEY `ContactINDEX` (`Contact`);

--

-- Индексы таблицы `Lom\_PeopleContactType`

--

ALTER TABLE `Lom\_PeopleContactType`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `Lom\_PeopleOrganizations`

--

ALTER TABLE `Lom\_PeopleOrganizations`

ADD PRIMARY KEY (`id`),

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE,

ADD KEY `OrganizationsINDEX` (`id\_Organizations`) USING BTREE;

--

-- Индексы таблицы `Lom\_PinCode`

--

ALTER TABLE `Lom\_PinCode`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `PinCodeUNIQUE` (`PinCode`) USING BTREE,

ADD KEY `PeopleINDEX` (`id\_People`) USING BTREE,

ADD KEY `lom\_pincode\_ibfk\_2` (`id\_PinCodeType`);

--

-- Индексы таблицы `migrations`

--

ALTER TABLE `migrations`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `oauth\_access\_tokens`

--

ALTER TABLE `oauth\_access\_tokens`

ADD PRIMARY KEY (`id`),

ADD KEY `oauth\_access\_tokens\_user\_id\_index` (`user\_id`);

--

-- Индексы таблицы `oauth\_auth\_codes`

--

ALTER TABLE `oauth\_auth\_codes`

ADD PRIMARY KEY (`id`),

ADD KEY `oauth\_auth\_codes\_user\_id\_index` (`user\_id`);

--

-- Индексы таблицы `oauth\_clients`

--

ALTER TABLE `oauth\_clients`

ADD PRIMARY KEY (`id`),

ADD KEY `oauth\_clients\_user\_id\_index` (`user\_id`);

--

-- Индексы таблицы `oauth\_personal\_access\_clients`

--

ALTER TABLE `oauth\_personal\_access\_clients`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `oauth\_refresh\_tokens`

--

ALTER TABLE `oauth\_refresh\_tokens`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `password\_resets`

--

ALTER TABLE `password\_resets`

ADD KEY `password\_resets\_email\_index` (`email`);

--

-- Индексы таблицы `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `users\_email\_unique` (`email`);

--

-- AUTO\_INCREMENT для сохранённых таблиц

--

--

-- AUTO\_INCREMENT для таблицы `admin\_menu`

--

ALTER TABLE `admin\_menu`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=15;

--

-- AUTO\_INCREMENT для таблицы `admin\_operation\_log`

--

ALTER TABLE `admin\_operation\_log`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=586;

--

-- AUTO\_INCREMENT для таблицы `admin\_permissions`

--

ALTER TABLE `admin\_permissions`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;

--

-- AUTO\_INCREMENT для таблицы `admin\_roles`

--

ALTER TABLE `admin\_roles`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT для таблицы `admin\_users`

--

ALTER TABLE `admin\_users`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT для таблицы `Lom\_AccessList`

--

ALTER TABLE `Lom\_AccessList`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;

--

-- AUTO\_INCREMENT для таблицы `Lom\_ConfigTable`

--

ALTER TABLE `Lom\_ConfigTable`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=142;

--

-- AUTO\_INCREMENT для таблицы `Lom\_EventResult`

--

ALTER TABLE `Lom\_EventResult`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT для таблицы `Lom\_EventsHandling`

--

ALTER TABLE `Lom\_EventsHandling`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2096;

--

-- AUTO\_INCREMENT для таблицы `Lom\_EventsLogs`

--

ALTER TABLE `Lom\_EventsLogs`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2095;

--

-- AUTO\_INCREMENT для таблицы `Lom\_EventType`

--

ALTER TABLE `Lom\_EventType`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=141;

--

-- AUTO\_INCREMENT для таблицы `Lom\_IMEICode`

--

ALTER TABLE `Lom\_IMEICode`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT для таблицы `Lom\_OrderCar`

--

ALTER TABLE `Lom\_OrderCar`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT для таблицы `Lom\_Organizations`

--

ALTER TABLE `Lom\_Organizations`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT для таблицы `Lom\_People`

--

ALTER TABLE `Lom\_People`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=18;

--

-- AUTO\_INCREMENT для таблицы `Lom\_PeopleAccessInterim`

--

ALTER TABLE `Lom\_PeopleAccessInterim`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=23;

--

-- AUTO\_INCREMENT для таблицы `Lom\_PeopleContact`

--

ALTER TABLE `Lom\_PeopleContact`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=11;

--

-- AUTO\_INCREMENT для таблицы `Lom\_PeopleContactType`

--

ALTER TABLE `Lom\_PeopleContactType`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT для таблицы `Lom\_PeopleOrganizations`

--

ALTER TABLE `Lom\_PeopleOrganizations`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;

--

-- AUTO\_INCREMENT для таблицы `Lom\_PinCode`

--

ALTER TABLE `Lom\_PinCode`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=1806;

--

-- AUTO\_INCREMENT для таблицы `migrations`

--

ALTER TABLE `migrations`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=10;

--

-- AUTO\_INCREMENT для таблицы `oauth\_clients`

--

ALTER TABLE `oauth\_clients`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT для таблицы `oauth\_personal\_access\_clients`

--

ALTER TABLE `oauth\_personal\_access\_clients`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT для таблицы `users`

--

ALTER TABLE `users`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- Ограничения внешнего ключа сохраненных таблиц

--

--

-- Ограничения внешнего ключа таблицы `Lom\_EventsHandling`

--

ALTER TABLE `Lom\_EventsHandling`

ADD CONSTRAINT `lom\_eventshandling\_ibfk\_2` FOREIGN KEY (`id\_EventType`) REFERENCES `Lom\_EventType` (`id`),

ADD CONSTRAINT `lom\_eventshandling\_ibfk\_3` FOREIGN KEY (`id\_People`) REFERENCES `Lom\_People` (`id`);

--

-- Ограничения внешнего ключа таблицы `Lom\_EventsLogs`

--

ALTER TABLE `Lom\_EventsLogs`

ADD CONSTRAINT `lom\_eventslogs\_ibfk\_2` FOREIGN KEY (`id\_EventType`) REFERENCES `Lom\_EventType` (`id`),

ADD CONSTRAINT `lom\_eventslogs\_ibfk\_3` FOREIGN KEY (`id\_EventResult`) REFERENCES `Lom\_EventResult` (`id`),

ADD CONSTRAINT `lom\_eventslogs\_ibfk\_4` FOREIGN KEY (`id\_People`) REFERENCES `Lom\_People` (`id`);

--

-- Ограничения внешнего ключа таблицы `Lom\_OrderCar`

--

ALTER TABLE `Lom\_OrderCar`

ADD CONSTRAINT `lom\_ordercar\_ibfk\_1` FOREIGN KEY (`id\_People`) REFERENCES `Lom\_People` (`id`);

--

-- Ограничения внешнего ключа таблицы `Lom\_Organizations`

--

ALTER TABLE `Lom\_Organizations`

ADD CONSTRAINT `Lom\_Organizations\_ibfk\_1` FOREIGN KEY (`Parent`) REFERENCES `Lom\_Organizations` (`id`);

--

-- Ограничения внешнего ключа таблицы `Lom\_PeopleAccessInterim`

--

ALTER TABLE `Lom\_PeopleAccessInterim`

ADD CONSTRAINT `lom\_peopleaccessinterim\_ibfk\_1` FOREIGN KEY (`id\_People`) REFERENCES `Lom\_People` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `lom\_peopleaccessinterim\_ibfk\_2` FOREIGN KEY (`id\_Access`) REFERENCES `Lom\_AccessList` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Ограничения внешнего ключа таблицы `Lom\_PeopleContact`

--

ALTER TABLE `Lom\_PeopleContact`

ADD CONSTRAINT `lom\_peoplecontact\_ibfk\_1` FOREIGN KEY (`id\_People`) REFERENCES `Lom\_People` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `lom\_peoplecontact\_ibfk\_2` FOREIGN KEY (`id\_TypeContact`) REFERENCES `Lom\_PeopleContactType` (`id`);

--

-- Ограничения внешнего ключа таблицы `Lom\_PeopleOrganizations`

--

ALTER TABLE `Lom\_PeopleOrganizations`

ADD CONSTRAINT `lom\_peopleorganizations\_ibfk\_1` FOREIGN KEY (`id\_People`) REFERENCES `Lom\_People` (`id`),

ADD CONSTRAINT `lom\_peopleorganizations\_ibfk\_2` FOREIGN KEY (`id\_Organizations`) REFERENCES `Lom\_Organizations` (`id`);

--

-- Ограничения внешнего ключа таблицы `Lom\_PinCode`

--

ALTER TABLE `Lom\_PinCode`

ADD CONSTRAINT `lom\_pincode\_ibfk\_1` FOREIGN KEY (`id\_People`) REFERENCES `Lom\_People` (`id`),

ADD CONSTRAINT `lom\_pincode\_ibfk\_2` FOREIGN KEY (`id\_PinCodeType`) REFERENCES `Lom\_EventType` (`id`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

# ПРИЛОЖЕНИЕ В. СЦЕНАРИИ И РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ

АННОТАЦИЯ

В данном приложении указаны сценарии и результаты тестовых испытаний, выполненные по схеме тестирования.

В разделе схема тестирования, изображена схема тестирования.

В разделе сценарии испытаний и результаты, указаны сценарии и результаты.

1. Схема тестирования



Рисунок 1 - Схема тестирования

1. Сценарии испытаний и результаты

Таблица 1 - Описание информационных полей для тестирования

|  |  |
| --- | --- |
| Наименование | Описание |
| Наименование проекта | Наименование проекта проверено |
| Номер версии | Версия проекта (первый номер можно принять как 1.0) |
| Имя тестера | Имя тестера, который выполнял эти тесты |
| Даты тестирования | Даты когда проводили тестирование – это может быть один тест или несколько. Если тесты проводили через большие промежутки времени, дата тестирования может определятся отдельными тест кейсами |

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | Уникальный ID для каждого test case.Следуйте опредленной логике именования и нумерации. например ‘TC\_UI\_1′ указание на ‘пользовательский интерфейс test case #1′. |
| Приоритет тестирования *(Малый/Средний/высокий)* | Насколько важен каждый тест. Приоритет при испытании бизнес-правил или функционала может быть средним или высоким, в то время как незначительные формы пользовательского интерфейса могут быть с низким приоритетом. |
| Название тестирования/Имя | Название тестирования. Например, проверка формы авторизации с правильным логином и паролем. |
| Резюме испытания | Описание, чего нужно достигнуть при тестировании. |
| Шаги тестирования | Перечислите детально все шаги тестирования. Напишите в каком порядке должны быть выполнены эти шаги. Убедитесь что вы обеспечили настолько максимальную детализацию насколько можете. Нумерованный список – будет хорошей идей |
| Данные тестирования | Напишите тестовые данные используемые для этого тестирования. Таким образом актуальные данные, которые будут предложены будут использоваться для проведения тестирования. Например логин и пароль – для входа в систему. |
| Ожидаемый результат | Какой должен получится результат после выполнения теста? Опишите подробно ожидаемый результат включая любые сообщения и ошибки, которые должны быть выданы на экран. |
| Фактический результат | Какой фактический результат после выполнения теста? Опишите любое соответствующее поведение системы после выполнения тестирования. |
| Предпосылки | Любые предварительные действия, которые должны быть выполнены перед проведением тестирования. Перечислите предварительные условия, для успешного выполнения проекта |
| Постусловия | Какое состояние должно быть у системы после выполнения тестирования? |
| Статус *(Pass/Fail)* | Если фактический результат не соответствует ожидаемым результатам отметка, что тест провалился (fail). В противном случае как прошло (pass) |
| Комментарии | Используйте эту область для любых дополнительных записей или комментариев. Это область нужна для поддержки полей выше (например есть какие-то особые условия, которые не могут быть описаны ни в одном из полей или есть вопросы связанные с ожидаемыми или фактическими результатами) |

Test Case #1

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации вводимого времени в приложении для конфигурирования |
| Резюме испытания | Необходимо запущенное приложение для конфигурации. |
| Шаги тестирования | 1. Ввести время в поле ввода 2. Нажать кнопку добавить рядом |
| Данные тестирования | 21:00 |
| Ожидаемый результат | Время добавится в список |
| Фактический результат | Время добавится в список |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #2

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_2 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации вводимого времени в приложении для конфигурирования |
| Резюме испытания | Необходимо запущенное приложение для конфигурации. |
| Шаги тестирования | 1. Ввести время в поле ввода 2. Нажать кнопку добавить рядом |
| Данные тестирования | 21: |
| Ожидаемый результат | Время добавится в список |
| Фактический результат | Программа выдаст ошибку формата введённых данных |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #3

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_3 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации вводимого времени в приложении для конфигурирования |
| Резюме испытания | Необходимо запущенное приложение для конфигурации. |
| Шаги тестирования | 1. Ввести время в поле ввода 2. Нажать кнопку добавить рядом |
| Данные тестирования | 21:1 |
| Ожидаемый результат | Время добавится в список |
| Фактический результат | Время добавится в список |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #4

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_4 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации вводимого времени в приложении для конфигурирования |
| Резюме испытания | Необходимо запущенное приложение для конфигурации. |
| Шаги тестирования | 1. Ввести время в поле ввода 2. Нажать кнопку добавить рядом |
| Данные тестирования | : |
| Ожидаемый результат | Время добавится в список |
| Фактический результат | Программа выдаст ошибку формата введённых данных |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #5

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_5 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации вводимого времени в приложении для конфигурирования |
| Резюме испытания | Необходимо запущенное приложение для конфигурации. |
| Шаги тестирования | 1. Ввести время в поле ввода 2. Нажать кнопку добавить рядом |
| Данные тестирования | 19:00 |
| Ожидаемый результат | Время добавится в список |
| Фактический результат | Время добавится в список |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #6

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_6 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить коноль |
| Шаги тестирования | 1. Ввести наименование программы 2. Ввести ключ 3. Нажать Enter |
| Данные тестирования | /pin |
| Ожидаемый результат | Запуска программы |
| Фактический результат | Запуск программы в режиме pin |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #7

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_7 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить коноль |
| Шаги тестирования | 1. Ввести наименование программы 2. Ввести ключ 3. Нажать Enter |
| Данные тестирования | /load |
| Ожидаемый результат | Запуска программы |
| Фактический результат | Запуск программы в режиме load |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #8

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_8 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить коноль |
| Шаги тестирования | 1. Ввести наименование программы 2. Ввести ключ 3. Нажать Enter |
| Данные тестирования | /silent |
| Ожидаемый результат | Запуска программы |
| Фактический результат | Запуск программы в режиме silent |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #9

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_9 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить коноль |
| Шаги тестирования | 1. Ввести наименование программы 2. Ввести ключ 3. Нажать Enter |
| Данные тестирования | /stop |
| Ожидаемый результат | Остановка основного процесса |
| Фактический результат | Остановка основного процесса |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #10

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_10 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить коноль |
| Шаги тестирования | 1. Ввести наименование программы 2. Ввести ключ 3. Нажать Enter |
| Данные тестирования |  |
| Ожидаемый результат | Запуск программы |
| Фактический результат | Вывод сообщение об ошибке и все возможные ключи |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #11

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_11 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить мобильное приложение |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку авторизации |
| Данные тестирования | Login: Bushin  Password: 123456 |
| Ожидаемый результат | При верно введенных данных, пользователь авторизируется. |
| Фактический результат | При верно введенных данных, пользователь авторизируется. |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #12

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_12 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить мобильное приложение |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку авторизации |
| Данные тестирования | Login: Bushin  Password: |
| Ожидаемый результат | Выведет сообщение, о том что нужно заполнить поле пароля |
| Фактический результат | Выведет сообщение, о том что нужно заполнить поле пароля |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #13

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_13 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить мобильное приложение |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку авторизации |
| Данные тестирования | Login:  Password: 123456 |
| Ожидаемый результат | Выведет сообщение, о том что нужно заполнить поле логина |
| Фактический результат | Выведет сообщение, о том что нужно заполнить поле логина |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #14

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_14 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить мобильное приложение |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку авторизации |
| Данные тестирования | SurName: Bushin  FirstName: Alexey  PatronymicName: Yurievich  Login: Bushin  Password: 123456  RepeatPassword: 123456 |
| Ожидаемый результат | Успешно зарегистрирует пользователя |
| Фактический результат | Успешно зарегистрирует пользователя |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #15

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_15 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка обработка ключей |
| Резюме испытания | Необходимо запустить мобильное приложение |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку авторизации |
| Данные тестирования | SurName:  FirstName:  PatronymicName: Yurievich  Login: Bushin  Password: 123456  RepeatPassword: 123456 |
| Ожидаемый результат | Выведет сообщение, о том что необходимо заполнить поля |
| Фактический результат | Выведет сообщение, о том что необходимо заполнить поля |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #16

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_16 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации добавления прав доступа |
| Резюме испытания | Необходимо авторизоваться пользователю, и перейти к форме добавления прав доступа |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку добавления |
| Данные тестирования | Название доступа: Class D  Содержание: Права уровня D |
| Ожидаемый результат | Выведет сообщение, об успешном добавление |
| Фактический результат | Выведет сообщение, об успешном добавление |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #17

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_17 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации добавления прав доступа |
| Резюме испытания | Необходимо авторизоваться пользователю, и перейти к форме добавления прав доступа |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку добавления |
| Данные тестирования | Название доступа:  Содержание: Права уровня D |
| Ожидаемый результат | Выведет сообщение с ошибкой, об заполенние полей |
| Фактический результат | Выведет сообщение с ошибкой, об заполнение полей |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

Test Case #18

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_18 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка валидации добавления прав доступа |
| Резюме испытания | Необходимо авторизоваться пользователю, и перейти к форме добавления прав доступа |
| Шаги тестирования | 1. Заполнить поля 2. Нажать кнопку добавления |
| Данные тестирования | Название доступа: Class D  Содержание: |
| Ожидаемый результат | Выведет сообщение с ошибкой, об заполенние полей |
| Фактический результат | Выведет сообщение с ошибкой, об заполнение полей |
| Предпосылки | Отсутствуют |
| Постусловия | Отсутствуют |
| Статус (Pass/Fail) | Pass |
| Комментарии | Отсутствуют |

# ПРИЛОЖЕНИЕ Г. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

АННОТАЦИЯ

Данный комплекс программного обеспечения нацелен, на обновление и копировае информации из одной базы данных в другую.

В разделе условия работы описываются необходимые для работы технические средства и их характеристики.

В разделе описание работы приводятся все сценарии работы пользователя с формами веб-приложения.

1. Условия работы

Для того, чтобы приступить к работе необходимо иметь операционную систему Windows с установленным .NET Framework, подключение к сети интернет, а также мобильной телефон на базе Android.

1. Описание работы
   1. Консольное приложение

Запускается через консоль и имею следующие ключи:

1. /pin – выполняет генерацию пин-кодов;
2. /load – выполняет копирование базы данных;
3. /stop – останавливает рабоыт основного процесса приложения;
4. /silent – запускает вечный процесс, который выполняет действия выше по расписанию.
   1. Приложение по конфигурированию

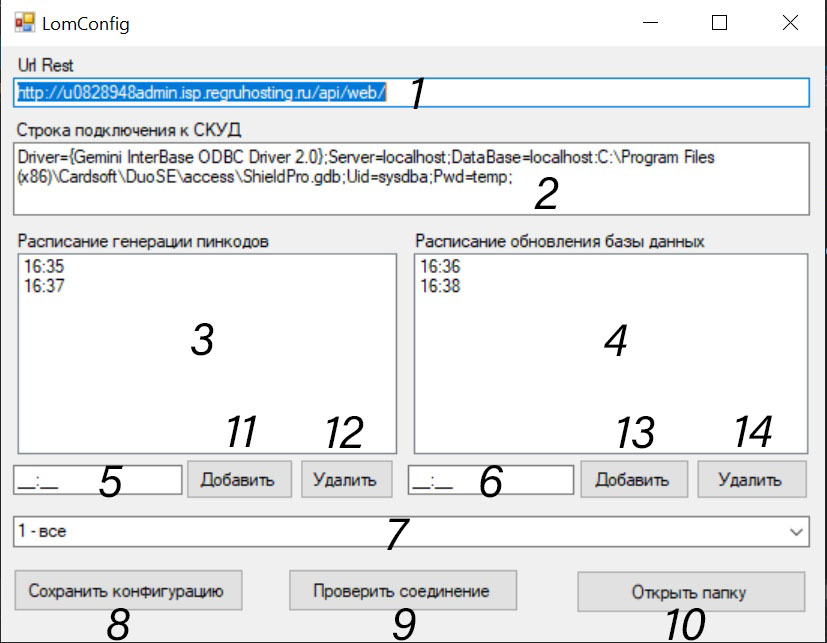


Рисунок 1 - Главное окно приложения конфигурации

1. Строка подключения к REST API;
2. Строка подключения к базе данных СКУД;
3. Расписание генерации пин-кодов;
4. Расписание обновления базы данных;
5. Поле ввода расписания генерации пин-кода;
6. Поле ввода расписания обновления базы данных;
7. Выпадающий список выбора организации;
8. Сохранения конфигурации;
9. Проверка соединения с базой данных СКУД;
10. Открытие директории с системными файлами;
11. Добавление расписания генерации пин-кодов;
12. Удаление выбранной временной отметки расписания;
13. Добавление расписания обновления базы данных;
14. Удаление выбранной временной отметки расписания;
    1. WEB-панель администратора

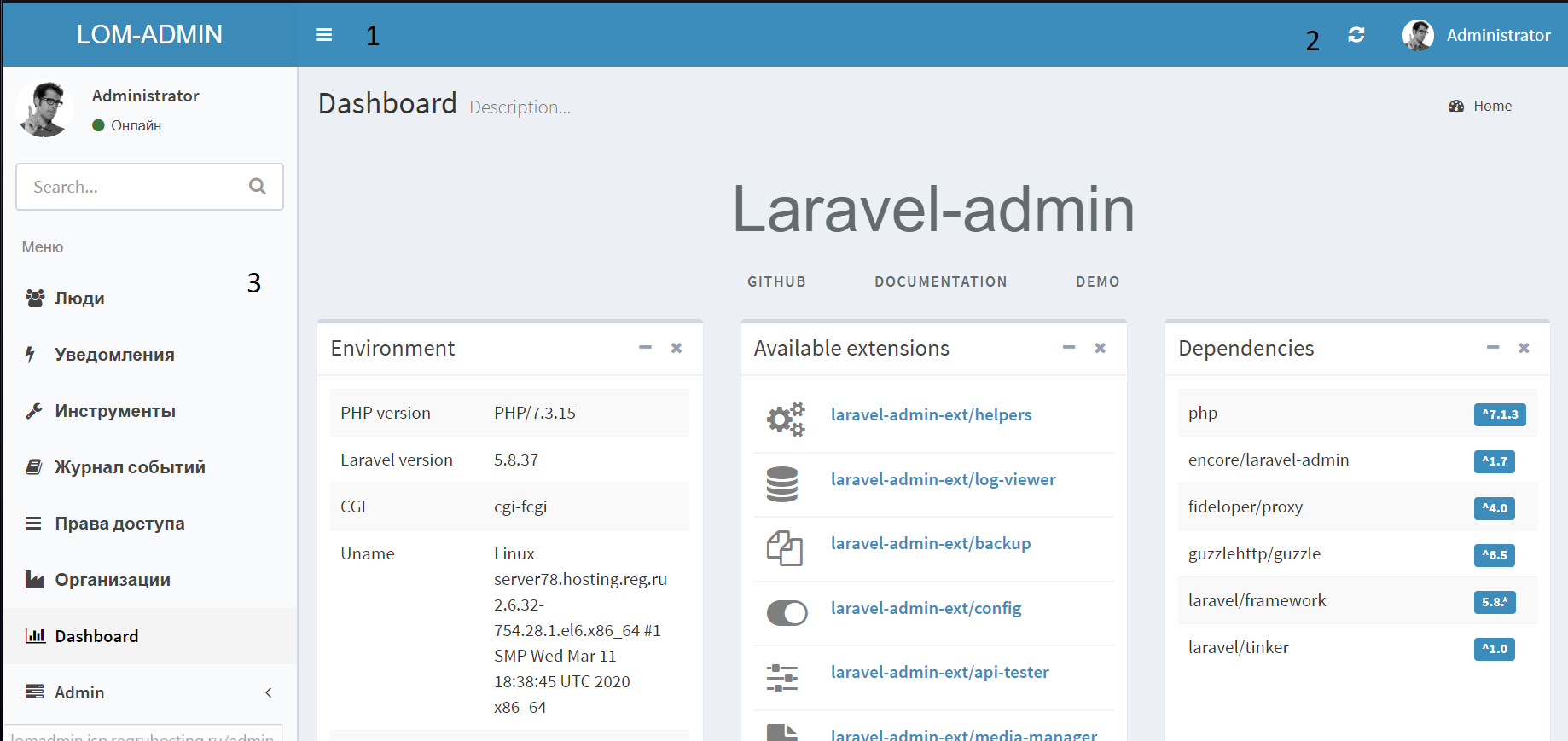


Рисунок 2 - Главная страница

1. Открытие/сокрытие навигационного меню
2. Информация о пользователе
3. Навигационное меню

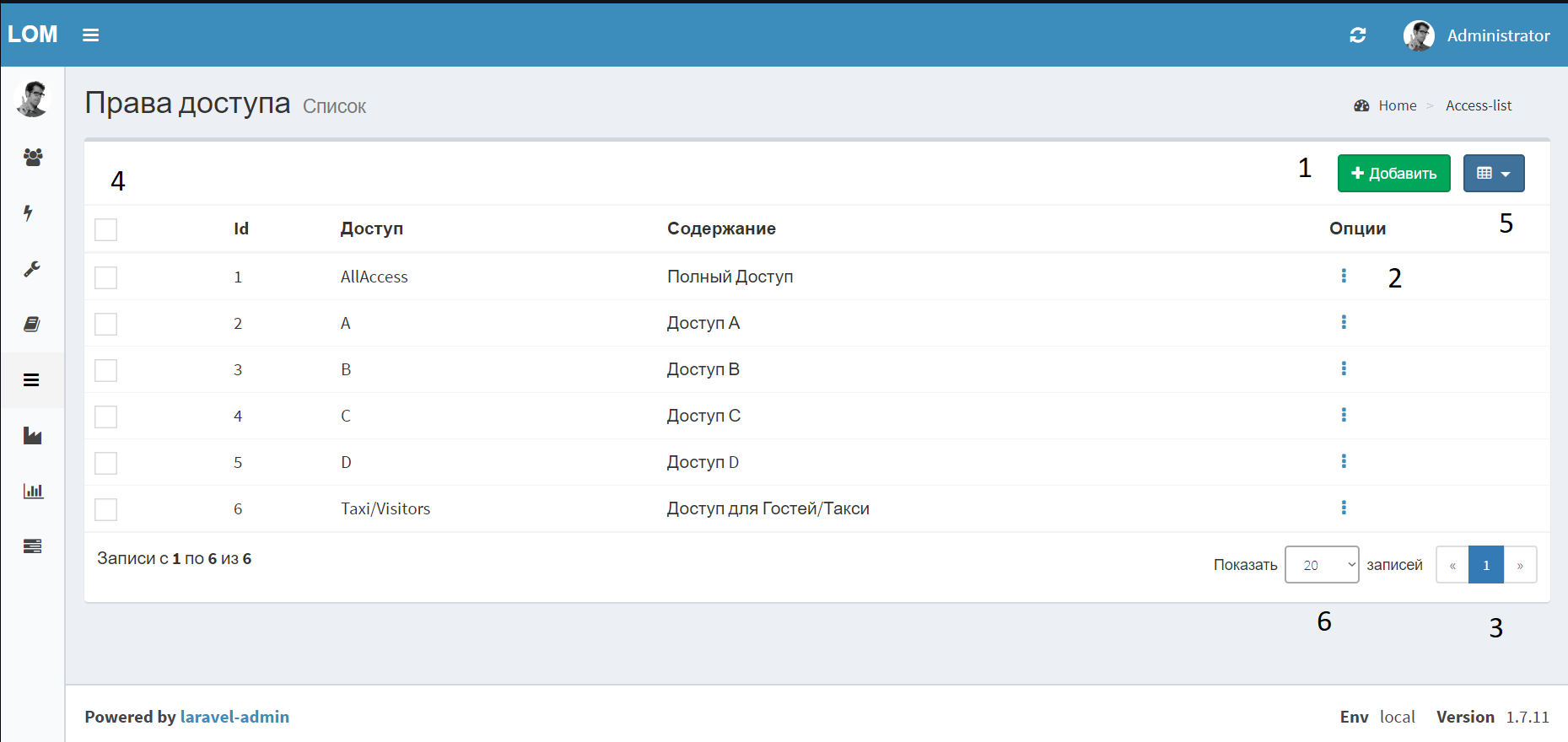


Рисунок 16 - Работа с таблицей

1. Добавление новой записи
2. Опции записи
3. Переход по страницам
4. Мульти-выбор записей
5. Выбор отображения определённых полей
6. Выбор кол-ва отображаемых записей
   1. Мобильное приложение конечного пользователя

Для экплуатации мобильного приложения необходимо иметь, активное подключение к сети интернет.

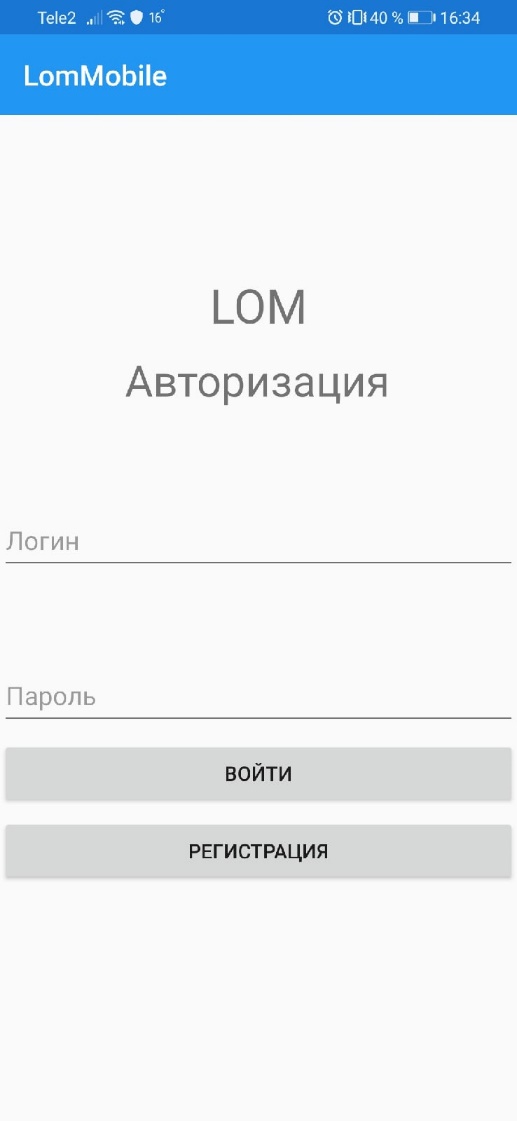


Рисунок 3 - Страница авторизации пользователя

1. Поле “Логин” обязательно для заполнения
2. Поле “Пароль” обязательно для заполнения
3. Кнопка “Авторизация”, отправка запроса на авторизацию
4. Кнопка “Регистрация” переход к странице с формой регистрации

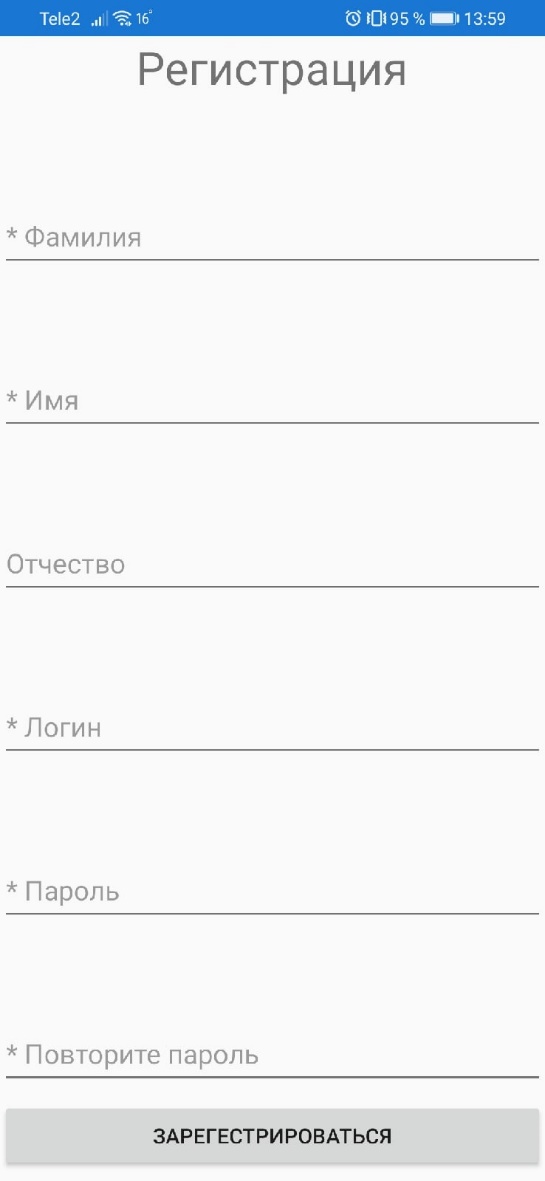


Рисунок 4 – Страница с формой регистрации пользователя

1. Поле “Фамилия”, должно содержать минимум 6 символов.
2. Поле “Имя” должно содержать минимум 2 символа.
3. Поле “Отчество” является не обязательным.
4. Поле “Логин” должно содержать минимум 6 символов.
5. Поле “Пароль” должно содержать минимум 6 символов.
6. Поле “Повторите пароль” должно совпадать с полем “Пароль”.



Рисунок 5 - Страница с информацией авторизированного пользователя

1. Вывод имени и фамилии с активным пин-кодом.
2. Кнопка “Обновить” для проверки существует ли новый пин-код.
3. Кнопка “Сгенерировать” для запроса на генерацию нового пин-кода.