Министерство образования и науки Республики Башкортостан Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

ОТЧЁТ

по учебной практике УП.04.01

по модулю ПМ.04 Сопровождение и обслуживание программного обеспечение компьютерных систем

Специальность

09.02.07 Информационные системы и программирование

Квалификация

Администратор баз данных

|  |  |
| --- | --- |
|  | Выполнил:  Студент группы 21БД-1  / И.И. Сахиуллин  подпись ФИО  « » 2024г.  Проверил:  / О.В.Фатхулова  подпись ФИО |

УФА - 2024 года

СОДЕРЖАНИЕ

лист

[1 Техническое задание 3](#_Toc183622776)

[1.1 Назначение разработки 3](#_Toc183622777)

[1.4 Стадии и этапы разработки 5](#_Toc183622778)

[1.5 Описание входной информации 6](#_Toc183622779)

[1.6 Описание выходного документа 7](#_Toc183622780)

[2 Разработка прототипа (скрины с фигмы) 9](#_Toc183622781)

[3 Структура проекта 15](#_Toc183622782)

[4 Описание запросов API 16](#_Toc183622783)

[5 Создание Docker контейнера 24](#_Toc183622784)

[6 Результат работы 27](#_Toc183622785)

[Заключение 30](#_Toc183622786)

ВВЕДЕНИЕ

Преимущества информационной системы: повышение эффективности и качества, снижение издержек, упрощение управления персоналом, централизованное хранение информации, улучшение планирования и взаимодействия между отделами. Информационная система хранит информацию о сотрудниках, отделах, этапах производства и технике.

Актуальность информационной системы заключается в том, что позволяет эффективно управлять всеми аспектами производства.

Целью работы является упрощение производства и улучшении эффективности производства.

Для достижения поставленной цели необходимо решить следующие задачи:

* изучить предметную область;
* спроектировать базу данных;
* разработать дизайн приложения;
* реализовать функцию авторизации сотрудников;
* реализовать функции для клиентов: просмотр данных производства, редактирование данных сотрудников;
* реализовать функции для начальника цеха: просмотр данных производства, добавление этапа производства, редактирование этапа производства, создание отчёта;
* разработать и протестировать веб-приложение.

1 Техническое задание

1.1 Назначение разработки

Система предназначена для упрощения работы сотрудников ГИБДД, посредством автоматизации процесса регистрации транспортных средств клиентов. Система должна быть обеспечена легким и интуитивно понятным интерфейсом, возможностями безопасной работы с заявлениями для сотрудников, управлениями пользователями системы для администраторов, простым отслеживанием текущих поданных заявлений и историй, легким процессом подачи регистрации транспортных средств для клиентов.

1.2 Требования к программе

Аппаратное обеспечение Сервера:

Сервер:

* Процессор: Intel Xeon E-2334 (или аналогичный AMD EPYC);
* Оперативная память: 32 GB DDR4 ECC REG;
* Накопитель: NVMe SSD 1 TB;
* Сетевая карта: 1 Gigabit Ethernet.

Рабочие станции:

* Процессор: Intel Core i5/i7/i9 (аналоги AMD Ryzen 5/7/9);
* Оперативная память: 8 GB DDR4;
* Накопитель: SSD 256 GB;
* Сетевая карта: Gigabit Ethernet.

Программное обеспечение:

Операционная система:

* Для сервера: Linux (например, Ubuntu Server);
* Рабочие станции: Windows 11 редакции Professional;

Прикладное программное обеспечение для рабочих станций:

* Офисный пакет: Microsoft Office;
* Браузер: Edge, Google Chrome;
* Антивирусное ПО: Kaspersky Anti-Virus.

1.3 Требования к функциональным характеристикам

Система «AutoReg» должна обеспечить выполнение функций:

Администратор:

* Просмотр данных о своем аккаунте;
* Поиск пользователей по определенным критериям;
* Добавление и удаление пользователей.

Сотрудник:

* Просмотр данных о своем аккаунте;
* Поиск заявлений по определенным критериям;
* Изменения статуса заявлений, в зависимости от степени успешности данной заявки.

Клиент:

* Просмотр данных о своем аккаунте;
* Просмотр текущих и архивных заявлений;
* Поиск заявлений текущих и архивных заявлений по определенным критериям;
* Управление своими текущими заявлениями;
* Возможность подачи новой заявки.

1.4 Стадии и этапы разработки

После утверждения технического задания организация-разработчик непосредственно приступает к разработке программного обеспечения.

Таблица 1.4.1 – Стадии и этапы разработки

|  |  |  |
| --- | --- | --- |
| Название этапа | Срок сдачи | Отчетность |
| 2 | 3 | 4 |
| Анализ требований | 8.10.2024 | Анализ требований |
| Написание технического задания | 9.10.2024-10.10.2024 | Техническое задание |
| Написание описания предметной области | 10.10.2024 | Описание предметной области |
| Написание входной/выходной информации | 11.10.2024 | Входная/Выходная информации |
| Написание контрольного примера | 12.10.2024-13.10.2024 | Контрольный пример |
| Проектирование ПО | 14.10.2024-21.10.2024 |  |
| Разработка прототипа ПО | 14.10.2024- 15.10.2024 | Прототип ПО |
| Разработка UML-диаграммы | 16.10.2024 | UML-диаграмма |
| Создание алгоритма разрабатываемого ПО | 17.10.2024 | Алгоритм разработки ПО |
| Создание структуры базы данных | 18.10.2024 | Структура базы данных |
| Написание контроллеров+HTTP-запросов | 19.10.2024-21.10.2024 | Контроллеры+HTTP-запросы |
| Разработка ПО | 22.10.2024-26.10.2024 | Структура ПО |

Продолжение таблицы 1.4.1.

|  |  |  |
| --- | --- | --- |
| Создание базы данных | 22.10.2024-23.10.2024 | База данных |
| Создание API | 24.10.2024-26.10.2024 | API |
| Тестирование ПО, API | 27.10.2024-29.10.2024 |  |
| Написание модульного тестирования | 27.10.2024 | Протокол модульного тестирования для веб-сервиса |
| Написание функционального тестирования | 28.10.2024 | Чек-лист тестирования веб-сервиса |
| Юзабилити | 29.10.2024 | Юзабилити |
| Сопровождение ПО | 30.10.2024-01.10.2024 |  |
| Сопровождение руководства пользователя | 30.10.2024-01.10.2024 | Руководство пользователя |

1.5 Описание входной информации

Входной информацией для системы будут являться данные о пользователях, о транспортных средствах, о заявках, об отделениях ГИБДД.

Информация о пользователях содержит следующие параметры:

* Фамилия и Имя пользователя;
* Паспортные данные;
* Контактные данные;

Информация о транспортных средствах содержит следующие параметры:

* Характеристики транспорта;

Информация о заявках содержит следующие параметры:

* Данные о клиенте;
* Данные о транспорте клиента;
* Информация о месте проведения (отделения);
* Информация о обрабатывающим заявку сотруднике;
* Информация о дате подачи, назначения и закрытия заявки;
* Информация о статусе и результате заявления;

Информация об отделениях ГИБДД содержит следующие параметры:

* Наименование отделения;
* Адрес отделения;
* Время работы;
* Данные о руководителе отделения;
* Номер телефона руководителя;

1.6 Описание выходного документа

Выходным документом будет являться отчет об успешности рассмотренного заявления. Описание документа представлен в таблице 1.2.

Таблица 1.6.1 – Описание выходного документа.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование документа | Периодичность выдачи документа | Кол-во экз. | Куда передаются |
| Отчет об успешности | После рассмотрения | 1 | Клиенту |

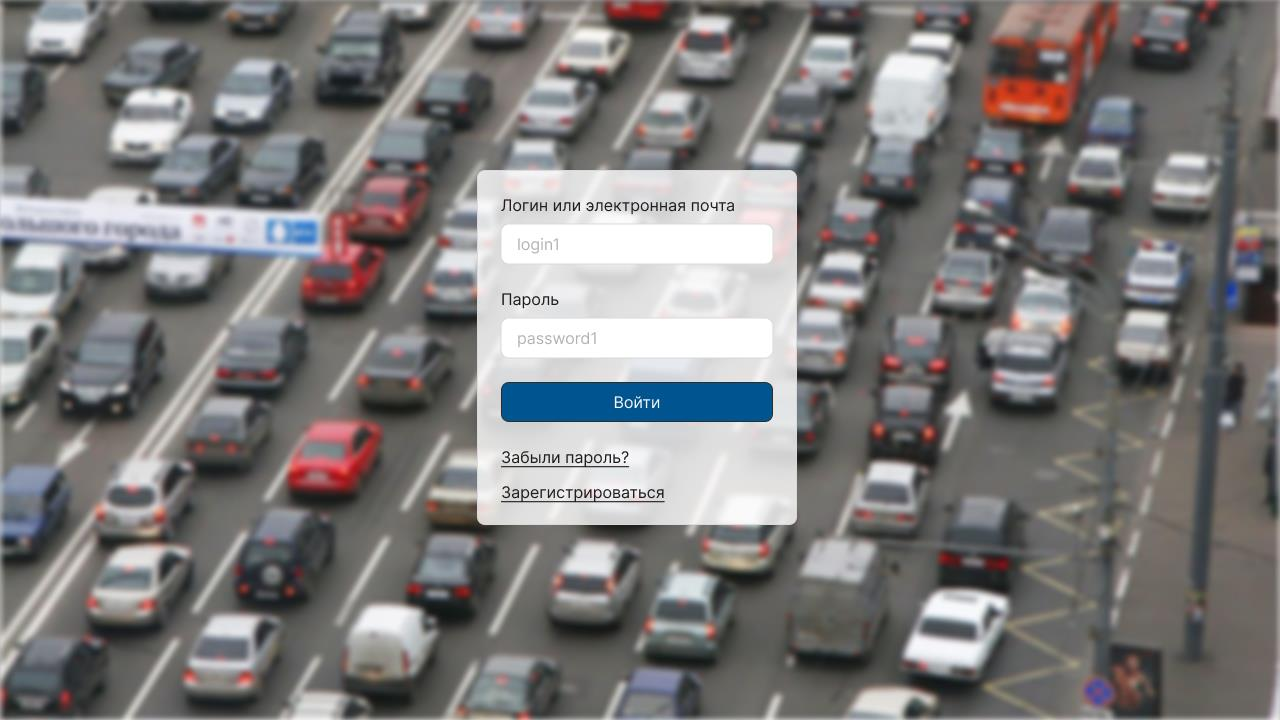
Образец выходного документа

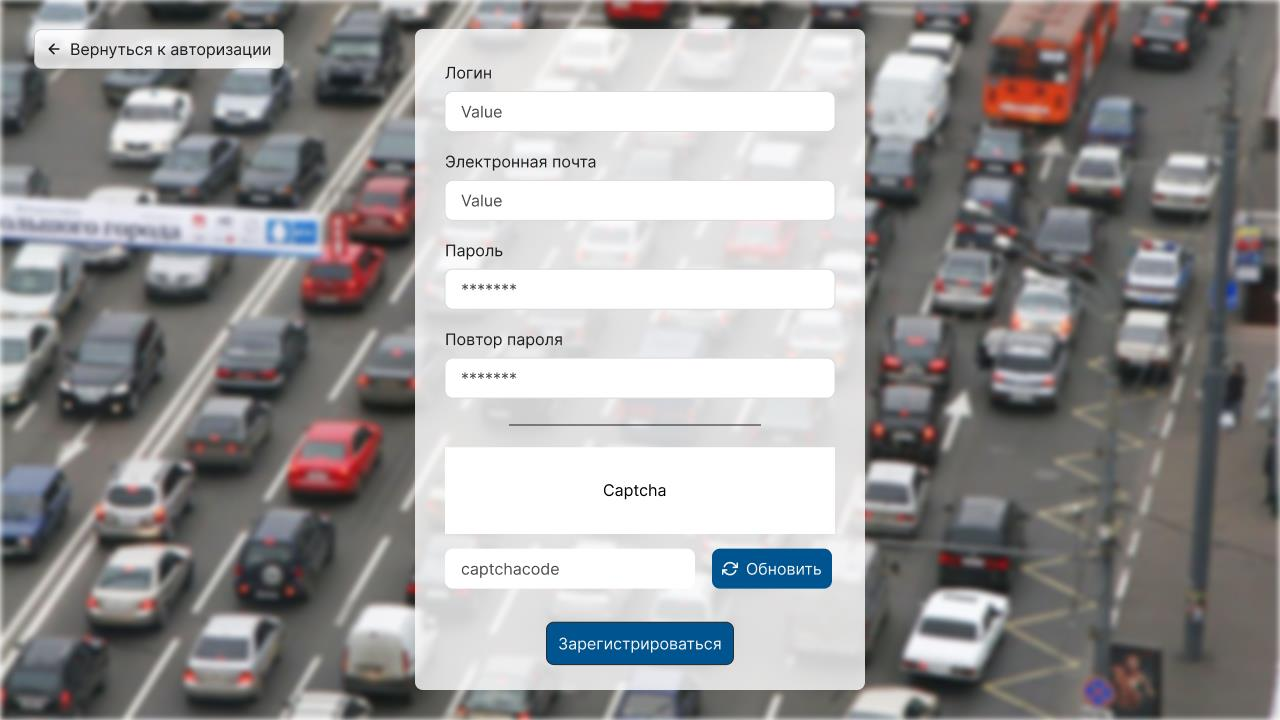


Рисунок 1.6.1 – Образец выходного документа «Отчет об отклонении заявления»

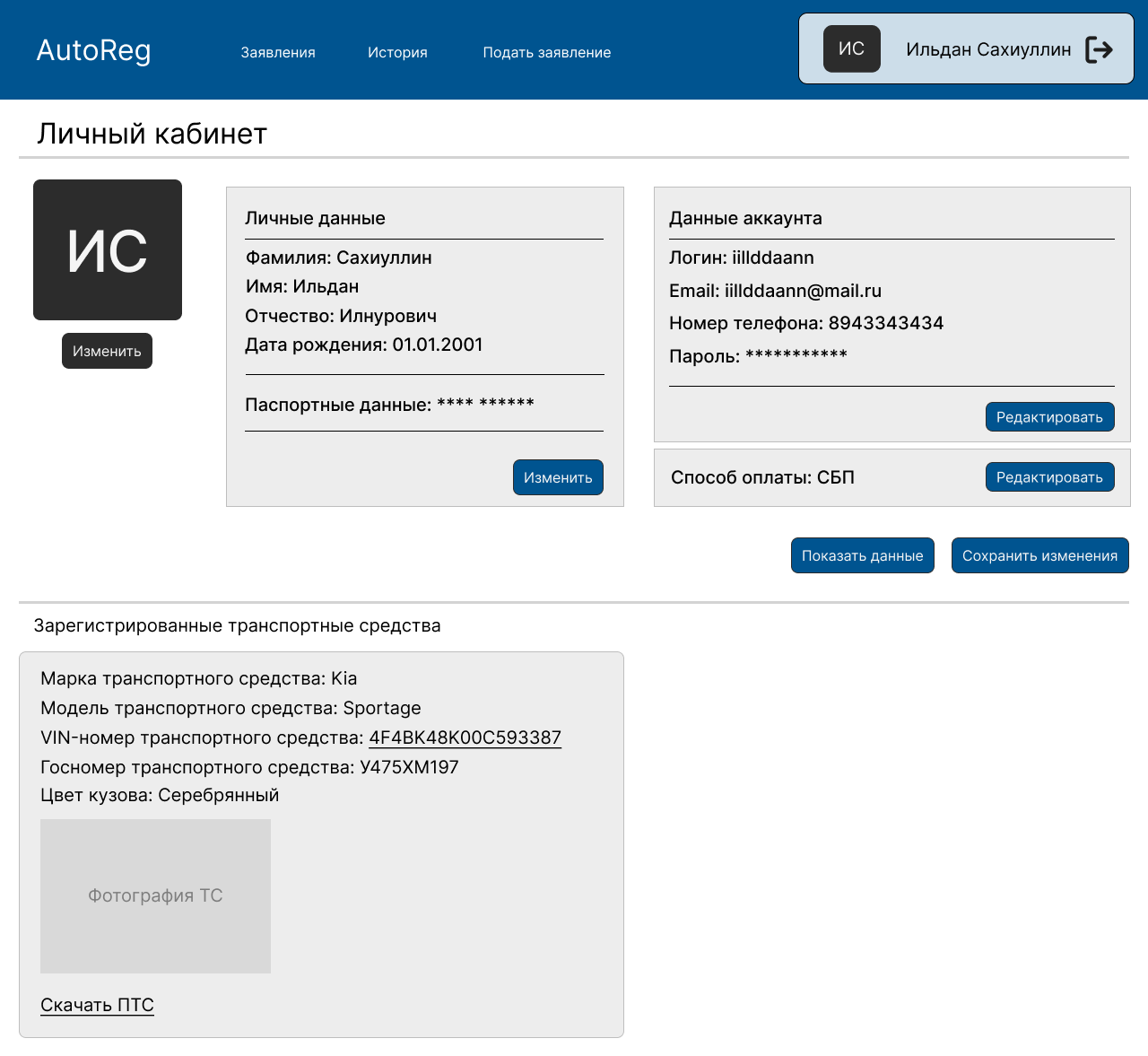
2 Разработка прототипа (скрины с фигмы)

На рисунках 2.1 и 2.2 показана возможность авторизоваться и зарегистрироваться в системе.

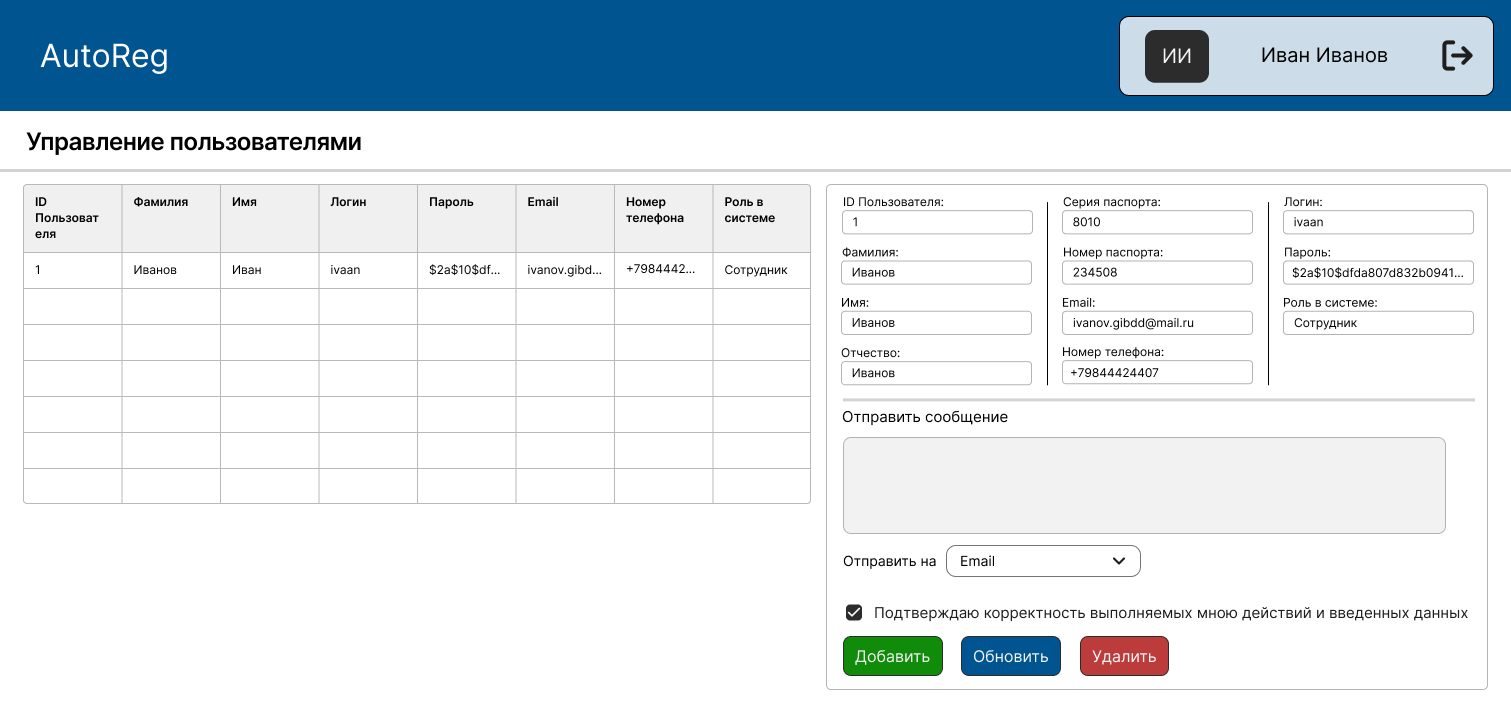
Рисунок 2.1 – Окно входа

Рисунок 2.2 – Окно регистрации

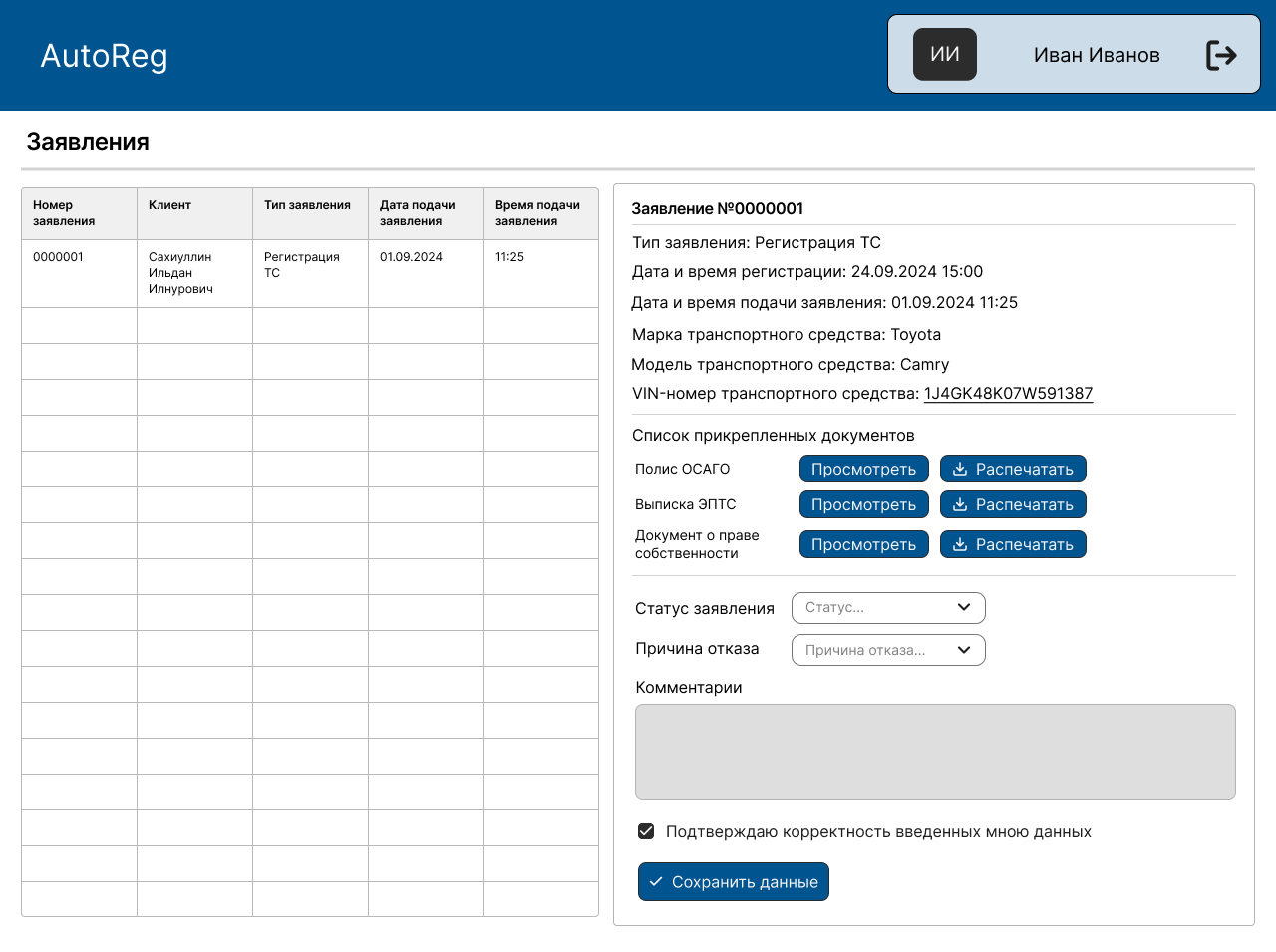
На рисунке 2.3 показан возможность просмотра данных в «личном кабинете».

Рисунок 2.3 – Окно профиля

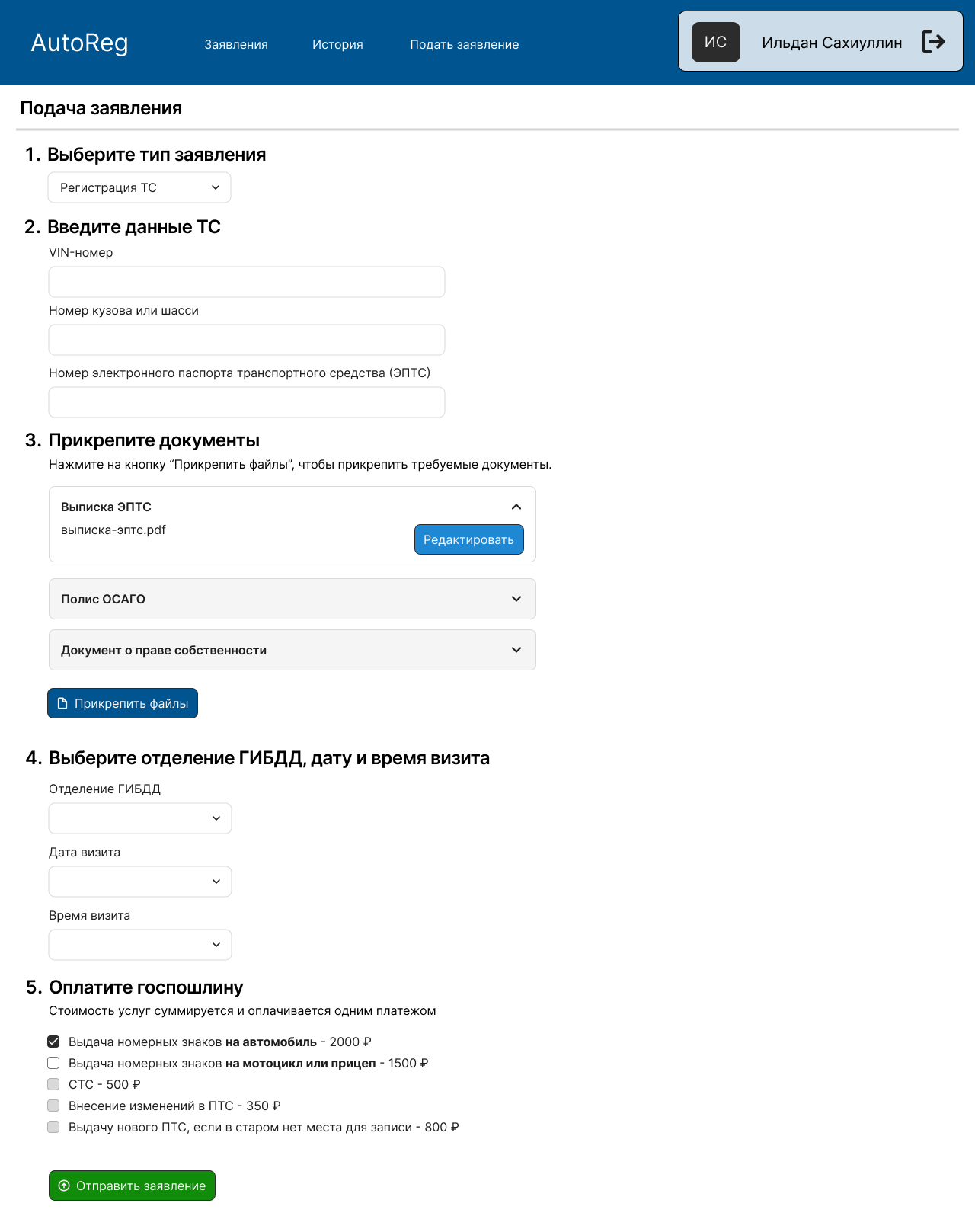
На рисунке 2.4 показана возможность управления пользователями для администраторов.

Рисунок 2.4 – Окно управления пользователями

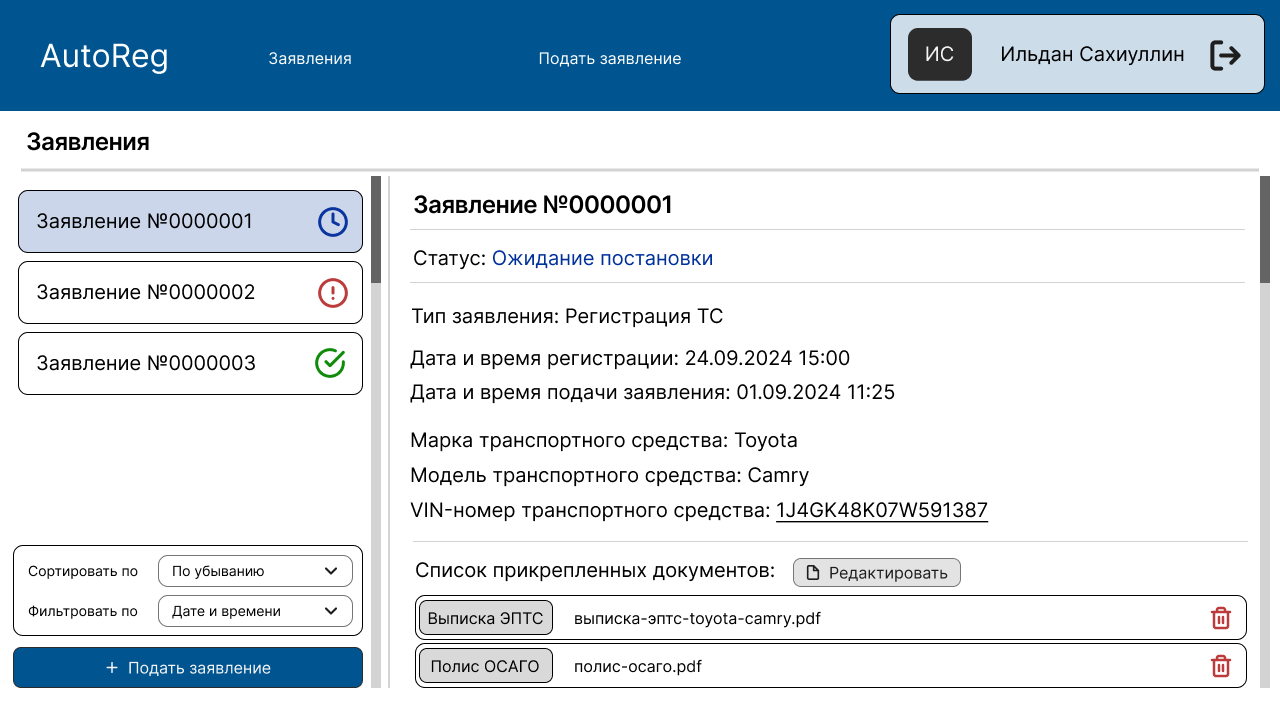
На рисунке 2.5 показана возможность управления заявлениями для сотрудников.

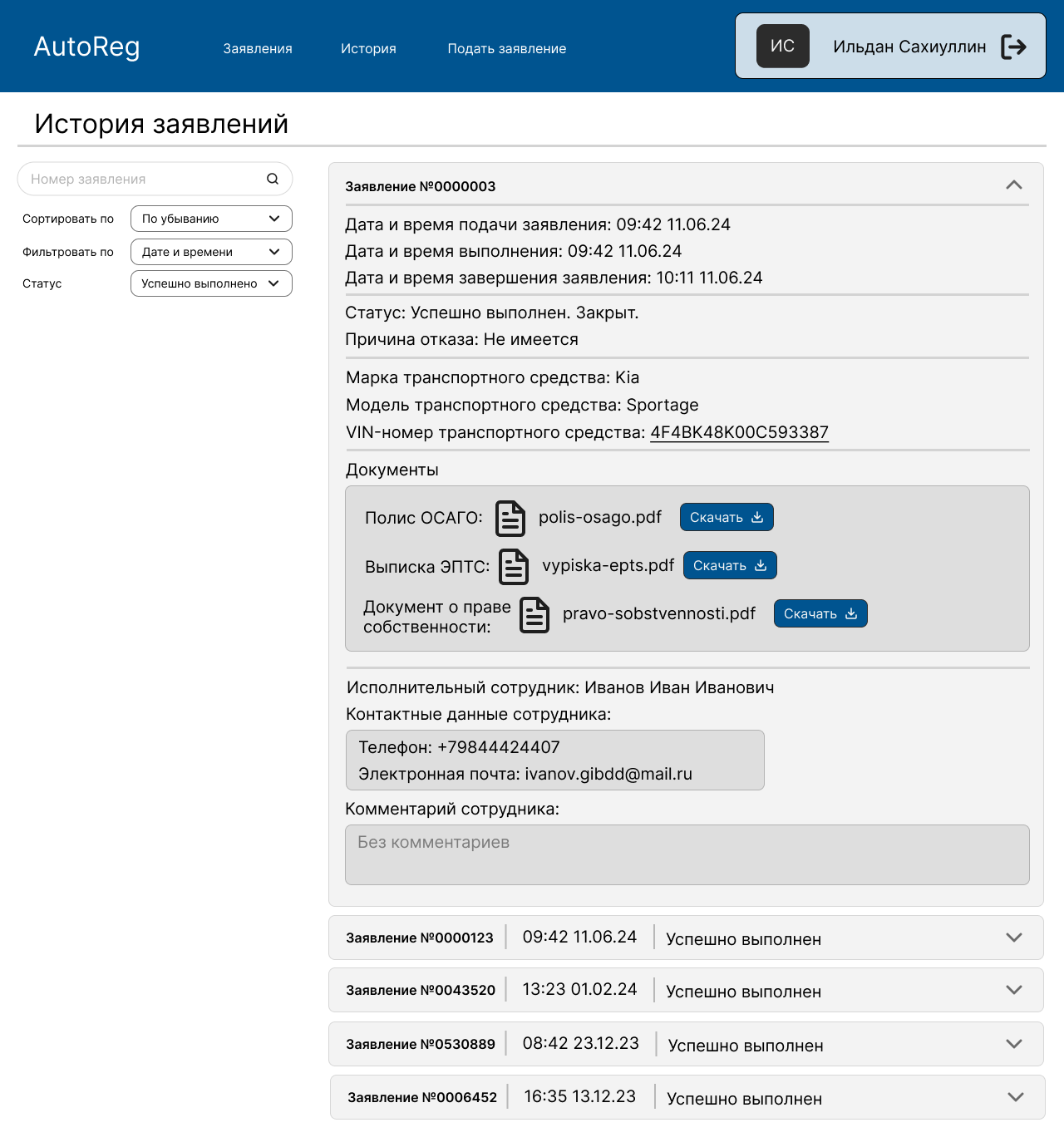
Рисунок 2.5 – Окно управления заявлениями

На рисунке 2.6 показана возможность подачи нового заявления для клиентов системы.

Рисунок 2.6 – Страница подачи заявления

На рисунке 2.7 и 2.8 показана возможность просматривать текущие и архивные заявления для клиентов системы соответственно.

Рисунок 2.7 – Страница с текущими заявлениями

Рисунок 2.8 – Страница истории заявлений

3 Структура проекта

Структура проекта представлена на рисунке 3.1.

Рисунок 3.1 – Структура проекта

Таблица 1.3 – Описание модулей и процедур

|  |  |  |
| --- | --- | --- |
| Модули | Процедура | Назначение |
| 1 | 2 | 3 |
| UserController.cs | GetUser(int id) | Получение данных о пользователе |
|  | LoginUser(string login, string password) | Авторизация пользователя |
|  | GetAllUsers(int condition) | Получение списка всех пользователей |
|  | AddNewUser([FromBody] User newUser) | Добавление нового пользователя |
|  | DeleteUser(int id) | Удаление пользователя |
|  | UpdateUser([FromBody] |  |
|  |  |  |

4 Описание запросов API

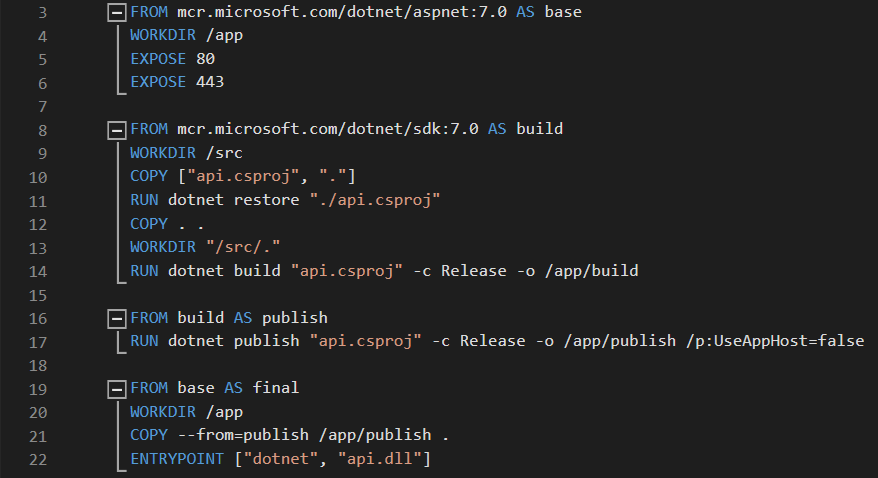
Таблица 4.1 – Описание модулей и процедур

|  |  |  |
| --- | --- | --- |
| Метод | Response body (возвращаемый результат) | Код результата |
| 1 | 2 | 3 |
| GetUser(int id) | {  "idUser": 1,  "login": "asdfasdf1234",  "password": "1234123",  "role": "Сотрудник",  "firstName": "fasdfasdf",  "lastName": "asdfasasd",  "passportSerial": "1234",  "passportNumber": "123456",  "email": "asdfasd@asdfasdf",  "phoneNumber": "1234123",  "applicationProcessingOfficerNavigations": [],  "applicationUsers": [],  "departaments": [],  "officer": null  } | 200 |
| LoginUser(string login, string password) | {"$id":"1","IdUser":2,"Login":"d1d1","Password":"d1d1","Role":"\u0410\u0434\u043C\u0438\u043D\u0438\u0441\u0442\u0440\u0430\u0442\u043E\u0440","FirstName":"3\u0430\u04303","LastName":"\u04303\u044B\u0430\u043043","PassportSerial":"2342","PassportNumber":"243233","Email":"fasfafasfsa@dfsafas","PhoneNumber":"243243243","ApplicationProcessingOfficerNavigations":{"$id":"2","$values":[]},"ApplicationUsers":{"$id":"3","$values":[]},"Departaments":{"$id":"4","$values":[]},"Officer":null} | 200 |
| GetAllUsers(int condition) | [  {  "idUser": 1,  "login": "asdfasdf1234",  "password": "1234123",  "role": "Сотрудник",  "firstName": "fasdfasdf",  "lastName": "asdfasasd",  "passportSerial": "1234",  "passportNumber": "123456",  "email": "asdfasd@asdfasdf",  "phoneNumber": "1234123",  "applicationProcessingOfficerNavigations": [],  "applicationUsers": [],  "departaments": [],  "officer": null  },  {  "idUser": 2,  "login": "d1d1",  "password": "d1d1",  "role": "Администратор",  "firstName": "3аа3",  "lastName": "а3ыаа43",  "passportSerial": "2342",  "passportNumber": "243233",  "email": "fasfafasfsa@dfsafas",  "phoneNumber": "243243243",  "applicationProcessingOfficerNavigations": [],  "applicationUsers": [],  "departaments": [],  "officer": null  },wwwww  {  "idUser": 3,  "login": "nik1decidefate",  "password": "n1234",  "role": "Клиент",  "firstName": "Матрохин",  "lastName": "Никита",  "passportSerial": "1234",  "passportNumber": "654321",  "email": "nikiiita@mail.ru",  "phoneNumber": "79123123123",  "applicationProcessingOfficerNavigations": [],  "applicationUsers": [],  "departaments": [],  "officer": null  },  {  "idUser": 4,  "login": "q123",  "password": "q123",  "role": "Администратор",  "firstName": "asd",  "lastName": "fasdf",  "passportSerial": null,  "passportNumber": null,  "email": "23232@adefasdf",  "phoneNumber": "22233344411",  "applicationProcessingOfficerNavigations": [],  "applicationUsers": [],  "departaments": [],  "officer": null  },  {  "idUser": 5,  "login": "qw234",  "password": "4433",  "role": "Сотрудник",  "firstName": "qrcoqw",  "lastName": "fqwfe2",  "passportSerial": null,  "passportNumber": null,  "email": "12341@mail.ru",  "phoneNumber": "33322111",  "applicationProcessingOfficerNavigations": [],  "applicationUsers": [],  "departaments": [],  "officer": null  }  ] | 200 |
| AddNewUser([FromBody] User newUser) | {  "idUser": 23,  "login": "scoc",  "password": "stuc",  "role": "Администратор",  "firstName": "pruc",  "lastName": "sruc",  "passportSerial": "1232",  "passportNumber": "132132",  "email": "asdfsdc@asdf",  "phoneNumber": "132132132132",  "applicationProcessingOfficerNavigations": [],  "applicationUsers": [],  "departaments": [],  "officer": null  } | 201 |
| DeleteUser(int id) | 23 | 200 |
| UpdateUser(int id, [FromBody] User changedUser) |  | 204 |
| GetApplication(int id) | {  "idApplication": 1,  "vehiclePassportId": 1,  "userId": 1,  "status": "Успешно завершено",  "submissionDate": "2024-03-12T00:00:00",  "scheduleDate": "2024-04-22T00:00:00",  "departamentId": 1,  "archiveDocuments": "\"PATH/TO/ARCH\"",  "processingOfficer": 1,  "closingDate": "2024-04-22T00:00:00",  "processingResult": "Заявка закрыта",  "departament": null,  "processingOfficerNavigation": null,  "user": null,  "vehiclePassport": null  } | 200 |
| RejectApplication(int id, string desctription) | {  "idApplication": 1,  "vehiclePassportId": 1,  "userId": 1,  "status": "Отклонено",  "submissionDate": "2024-03-12T00:00:00",  "scheduleDate": "2024-04-22T00:00:00",  "departamentId": 1,  "archiveDocuments": "\"PATH/TO/ARCH\"",  "processingOfficer": 1,  "closingDate": "2024-04-22T00:00:00",  "processingResult": "Не имеется документ о купле-продаже",  "departament": null,  "processingOfficerNavigation": null,  "user": null,  "vehiclePassport": null  } | 200 |
| GetAllApplications() | [{"IdApplication":1,"VehiclePassportId":1,"UserId":1,"Status":"Успешно завершено","SubmissionDate":"2024-03-12T00:00:00","ScheduleDate":"2024-04-22T00:00:00","DepartamentId":1,"ArchiveDocuments":"\"PATH/TO/ARCH\"","ProcessingOfficer":1,"ClosingDate":"2024-04-22T00:00:00","ProcessingResult":"Заявка закрыта","Departament":{"IdDepartament":1,"DepartmentName":"ГИБДД РБ","AddressId":1,"WorkFrom":"09:00:00","WorkTo":"18:00:00","HeadUserId":1,"PhoneNumber":"2322323","Address":{"IdAddress":1,"RegionId":1,"CityId":1,"DistrictId":1,"Street":"60 лет Декабря","Build":"12","City":{"IdCity":1,"City1":"Уфа","Addresses":[]},"Departaments":[],"District":{"IdDistrict":1,"District1":"Советский","Addresses":[]},"Region":{"IdRegion":1,"Region1":"Республика Башкортостан","Addresses":[]}},"Applications":[],"HeadUser":null,"Officers":[]},"ProcessingOfficerNavigation":null,"User":{"IdUser":1,"Login":"asdfasdf1234","Password":"1234123","Role":"Сотрудник","FirstName":"fasdfasdf","LastName":"asdfasasd","PassportSerial":"1234","PassportNumber":"123456","Email":"asdfasd@asdfasdf","PhoneNumber":"1234123","ApplicationProcessingOfficerNavigations":[],"ApplicationUsers":[],"Departaments":[],"Officer":null},"VehiclePassport":{"IdVehiclePassport":1,"VehicleId":1,"RegistrationNumber":"B321AB11","EngineNumber":"N123123","BodyNumber":"B12341234","IssueDate":"2020-04-11","Applications":[],"Vehicle":{"IdVehicle":1,"Vin":"123123123","ModelId":1,"VehicleTypeId":"Легковой","Year":2011,"ColorId":1,"Color":{"IdColor":1,"Color1":"Белый","Vehicles":[]},"Model":{"IdModel":1,"MakeId":3,"Model1":"Приора","Make":{"IdMake":3,"Make1":"Лада","Models":[]},"Vehicles":[]},"Vehiclepassports":[]}}},{"IdApplication":2,"VehiclePassportId":2,"UserId":2,"Status":"Отклонено","SubmissionDate":"2021-08-28T00:00:00","ScheduleDate":"2021-09-11T00:00:00","DepartamentId":1,"ArchiveDocuments":"\"PATH/TO/ARCH2\"","ProcessingOfficer":2,"ClosingDate":"2021-08-29T00:00:00","ProcessingResult":"Некорректные документы","Departament":{"IdDepartament":1,"DepartmentName":"ГИБДД РБ","AddressId":1,"WorkFrom":"09:00:00","WorkTo":"18:00:00","HeadUserId":1,"PhoneNumber":"2322323","Address":{"IdAddress":1,"RegionId":1,"CityId":1,"DistrictId":1,"Street":"60 лет Декабря","Build":"12","City":{"IdCity":1,"City1":"Уфа","Addresses":[]},"Departaments":[],"District":{"IdDistrict":1,"District1":"Советский","Addresses":[]},"Region":{"IdRegion":1,"Region1":"Республика Башкортостан","Addresses":[]}},"Applications":[],"HeadUser":null,"Officers":[]},"ProcessingOfficerNavigation":null,"User":{"IdUser":2,"Login":"d1d1","Password":"d1d1","Role":"Администратор","FirstName":"3аа3","LastName":"а3ыаа43","PassportSerial":"2342","PassportNumber":"243233","Email":"fasfafasfsa@dfsafas","PhoneNumber":"243243243","ApplicationProcessingOfficerNavigations":[],"ApplicationUsers":[],"Departaments":[],"Officer":null},"VehiclePassport":{"IdVehiclePassport":2,"VehicleId":2,"RegistrationNumber":null,"EngineNumber":"N123123","BodyNumber":"B123123","IssueDate":"2023-01-29","Applications":[],"Vehicle":{"IdVehicle":2,"Vin":"5345435","ModelId":2,"VehicleTypeId":"Легковой","Year":2015,"ColorId":1,"Color":{"IdColor":1,"Color1":"Белый","Vehicles":[]},"Model":{"IdModel":2,"MakeId":2,"Model1":"2115","Make":{"IdMake":2,"Make1":"ВАЗ","Models":[]},"Vehicles":[]},"Vehiclepassports":[]}}}] | 200 |
| Search(string query) | [  {  "idApplication": 2,  "vehiclePassportId": 2,  "userId": 2,  "status": "Отклонено",  "submissionDate": "2021-08-28T00:00:00",  "scheduleDate": "2021-09-11T00:00:00",  "departamentId": 1,  "archiveDocuments": "\"PATH/TO/ARCH2\"",  "processingOfficer": 2,  "closingDate": "2021-08-29T00:00:00",  "processingResult": "Некорректные документы",  "departament": null,  "processingOfficerNavigation": null,  "user": null,  "vehiclePassport": null  }  ] | 200 |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| Curl (входные данные) | URL (Строка запроса) |
| curl -X 'GET' \  'http://localhost:3000/api/User/1' \  -H 'accept: \*/\*' | http://localhost:3000/api/User/1 |
| curl -X 'GET' \  'http://localhost:3000/api/User/Login?login=d1d1&password=d1d1' \  -H 'accept: \*/\*' | http://localhost:3000/api/User/Login?login=d1d1&password=d1d1 |
| curl -X 'GET' \  'http://localhost:3000/api/User/GetAllUsers?condition=1' \  -H 'accept: \*/\*' | http://localhost:3000/api/User/GetAllUsers?condition=1 |
| curl -X 'POST' \  'http://localhost:3000/api/User/AddNewUser' \  -H 'accept: \*/\*' \  -H 'Content-Type: application/json' \  -d '{  "idUser": 23,  "login": "scoc",  "password": "stuc",  "role": "Администратор",  "firstName": "pruc",  "lastName": "sruc",  "passportSerial": "1232",  "passportNumber": "132132",  "email": "asdfsdc@asdf",  "phoneNumber": "132132132132"  }' | http://localhost:3000/api/User/AddNewUser |
| curl -X 'DELETE' \  'http://localhost:3000/api/User/DeleteUser?id=23' \  -H 'accept: \*/\*' | http://localhost:3000/api/User/DeleteUser?id=23 |
| curl -X 'PUT' \  'http://localhost:3000/api/User/1' \  -H 'accept: \*/\*' \  -H 'Content-Type: application/json' \  -d '{  "login": "string",  "password": "string",  "role": "string",  "firstName": "string",  "lastName": "string",  "passportSerial": "string",  "passportNumber": "string",  "email": "string",  "phoneNumber": "string",  "officer": {  "idOfficer": 0,  "userId": 0,  "departmentId": 0,  "postId": 0,  "workPhoneNumber": "string",  "department": "string",  "post": {  "idPost": 0,  "post1": "string"  },  "user": "string"  }  }' | http://localhost:3000/api/User/1 |
| curl -X 'GET' \  'http://localhost:3000/api/Application/1' \  -H 'accept: \*/\*' | http://localhost:3000/api/Application/1 |
| curl -X 'PATCH' \  'http://localhost:3000/api/Application/{id}?description=%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5' \  -H 'accept: \*/\*' | http://localhost:3000/api/Application/{id}?description=%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5 |
| curl -X 'PUT' \  'http://localhost:3000/api/Application/statuses/%D0%92%20%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%B5' \  -H 'accept: \*/\*' \  -H 'Content-Type: application/json' \  -d '{  "idApplication": 0,  "vehiclePassportId": 0,  "userId": 0,  "status": "Успешно завершено",  "submissionDate": "2024-11-26T22:45:48.323Z",  "scheduleDate": "2024-11-26T22:45:48.323Z",  "departamentId": 0,  "archiveDocuments": "string",  "processingOfficer": 0,  "closingDate": "2024-11-26T22:45:48.323Z"  }' | http://localhost:3000/api/Application/statuses/%D0%92%20%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%B5 |
| curl -X 'GET' \  'http://localhost:3000/api/Application/search?query=%D0%9E%D1%82%D0%BA%D0%BB%D0%BE%D0%BD%D0%B5%D0%BD%D0%BE' \  -H 'accept: \*/\*' | http://localhost:3000/api/Application/search?query=%D0%9E%D1%82%D0%BA%D0%BB%D0%BE%D0%BD%D0%B5%D0%BD%D0%BE |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

5 Создание Docker контейнера

Для создания Docker контейнера создается специальный файл Dockerfile с кодом, представленным на рисунке 5.1.

Рисунок 5.1 – Код внутри Dockerfile.

Затем описывается docker-compose.yml как основной файл для создания контейнеров. Код docker-compose.yml представлен в рисунке 5.2.

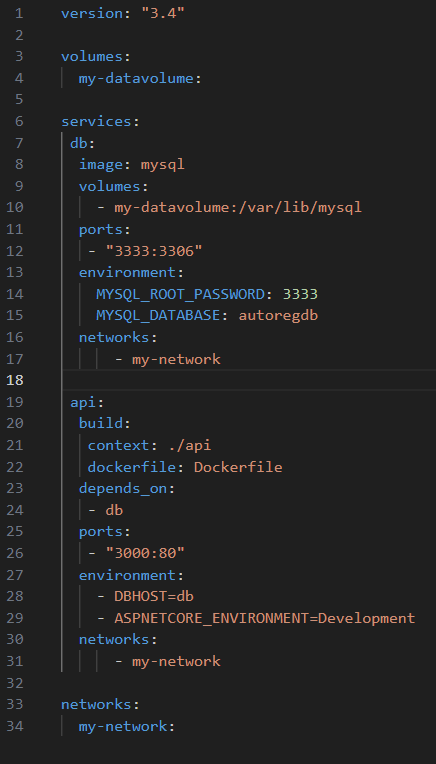
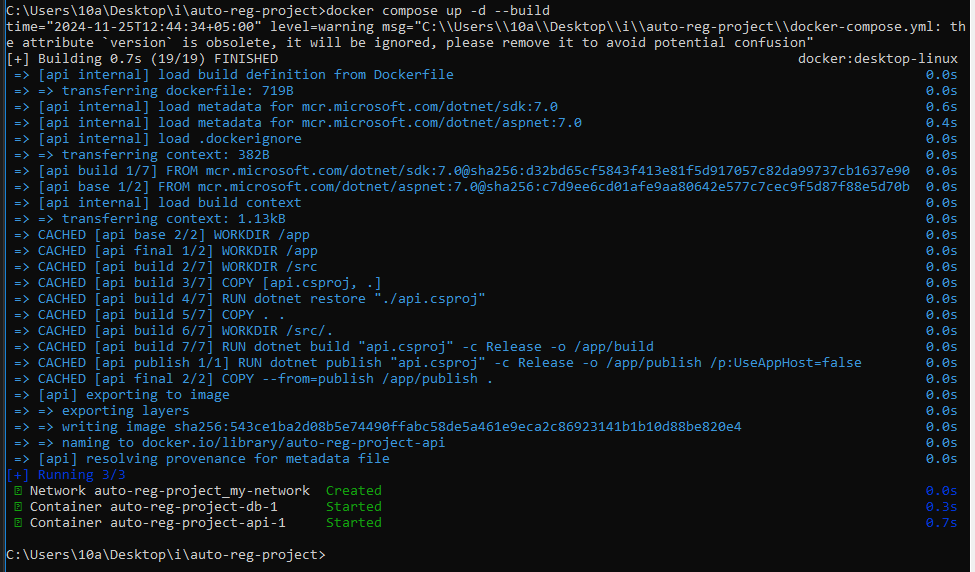
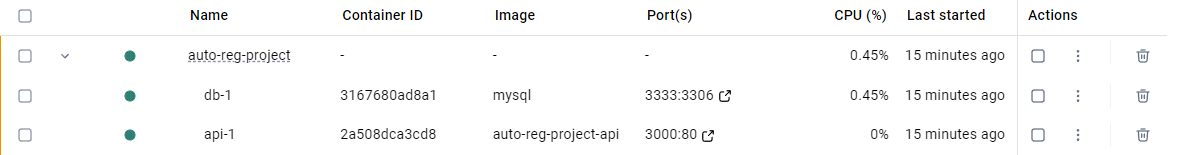


Рисунок 5.2 – Код внутри docker-compose.yml

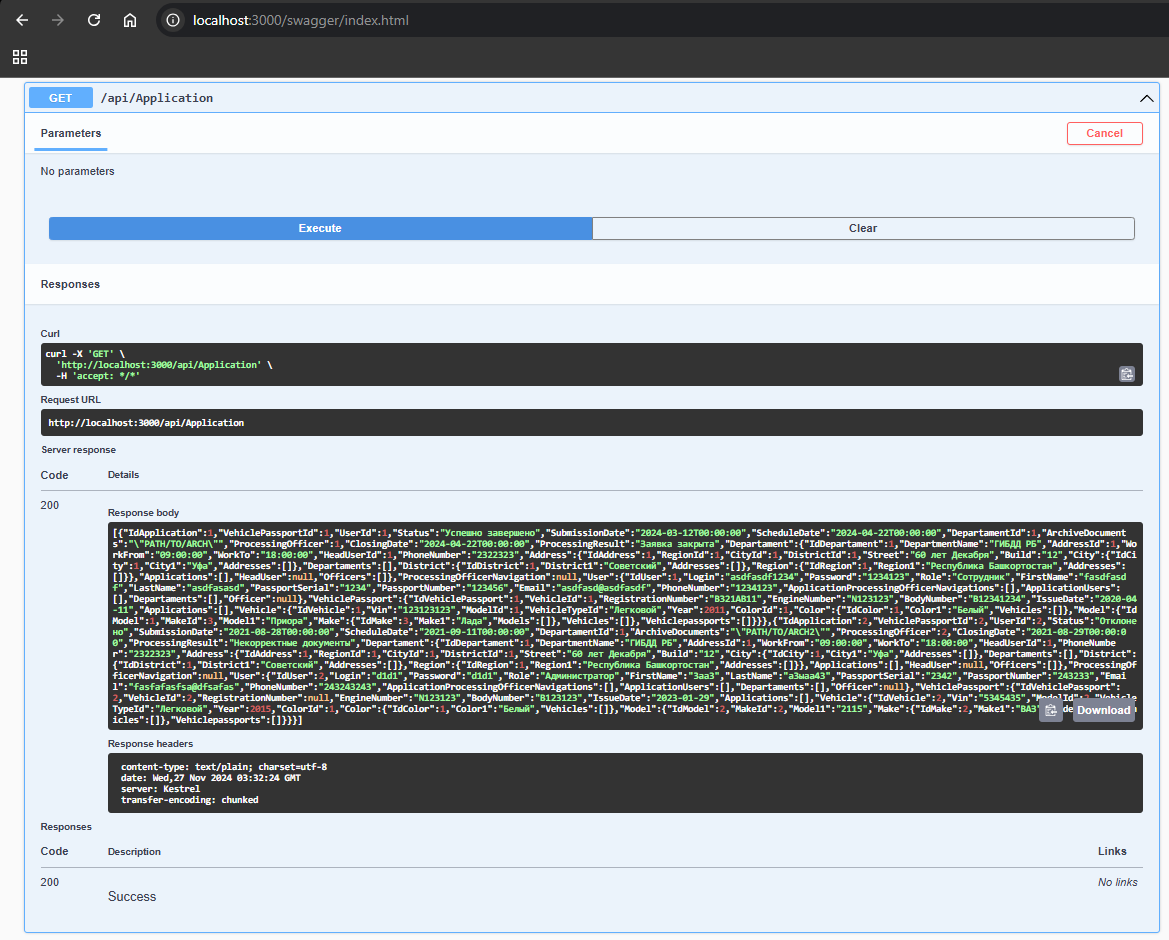
Далее с помощью команды “docker compose up -d --build” собираем проект используя данные в docker-compose.yml и Dockerfile и запускаем в фоновом режиме. Процесс работы команды представлен на рисунке 5.3.

Рисунок 5.3 – Процесс работы команды docker-compose up

Теперь в Docker Desktop появились наши контейнеры. На рисунке 5.4 показаны контейнеры.

Рисунок 5.4 – Контейнеры в Docker Desktop

Обращаясь к ресурсу по локальному адресу <http://localhost:3000/swagger>, убеждаемся что он работает. На рисунке 5.5 показан результат выполнения запроса.

Рисунок 5.5 – Результат запроса

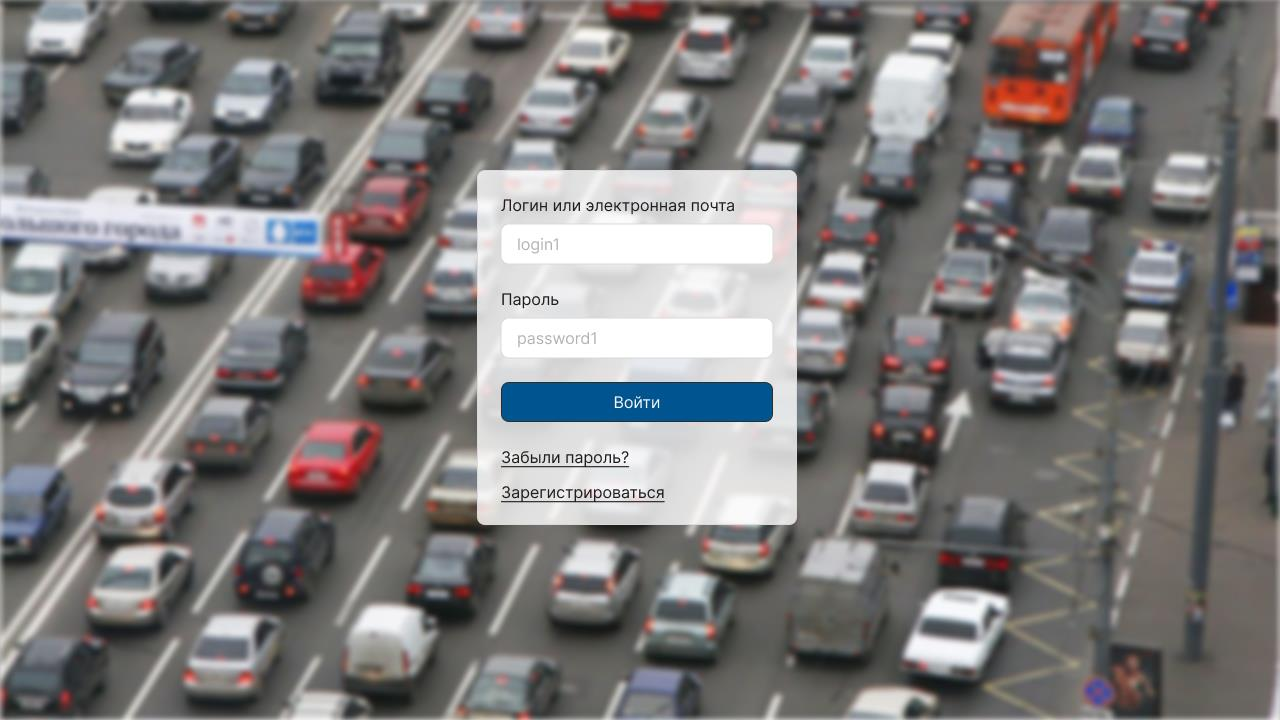
6 Результат работы

Техническая документация предназначена для пользователя, для того чтобы он самостоятельно пользовался программным продуктом.

Пользователь должен иметь опыт работы с персональным компьютером на базе операционной системы MS Windows и свободно выполнять базовые действия в программах.

Цель руководства заключается в расписании подробной инструкции и информации для пользователя, чтобы он смог самостоятельно пользоваться приложением и правильно его эксплуатировал.

При входе на сайт клиент попадает на страницу с авторизацией. Чтобы авторизоваться клиент должен ввести логин и пароль, а затем нажать на кнопку «Вход». Страница изображена на рисунке 6.1.

Рисунок 6.1 – Окно авторизации

Если клиент ввел неверные данные логина или пароля, на страница выводится сообщение о некорректных данных. Сообщение изображено на рисунке 6.2.

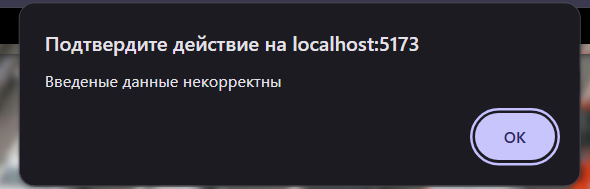
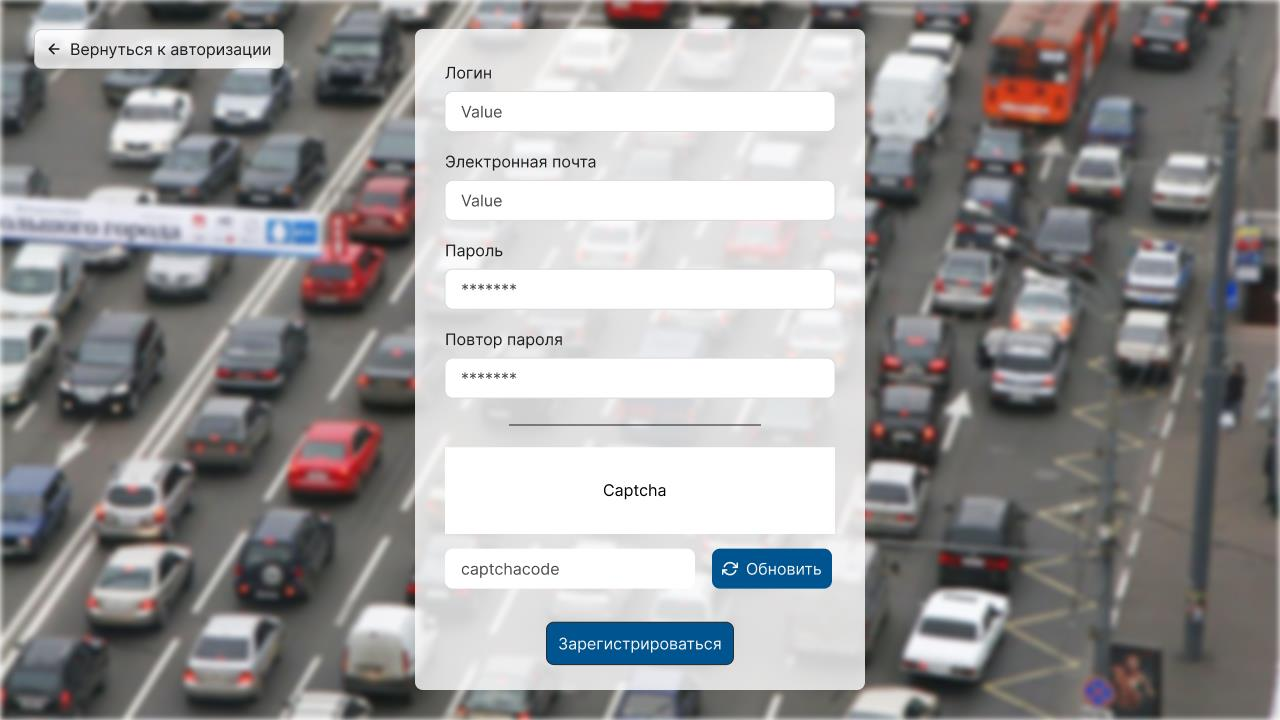
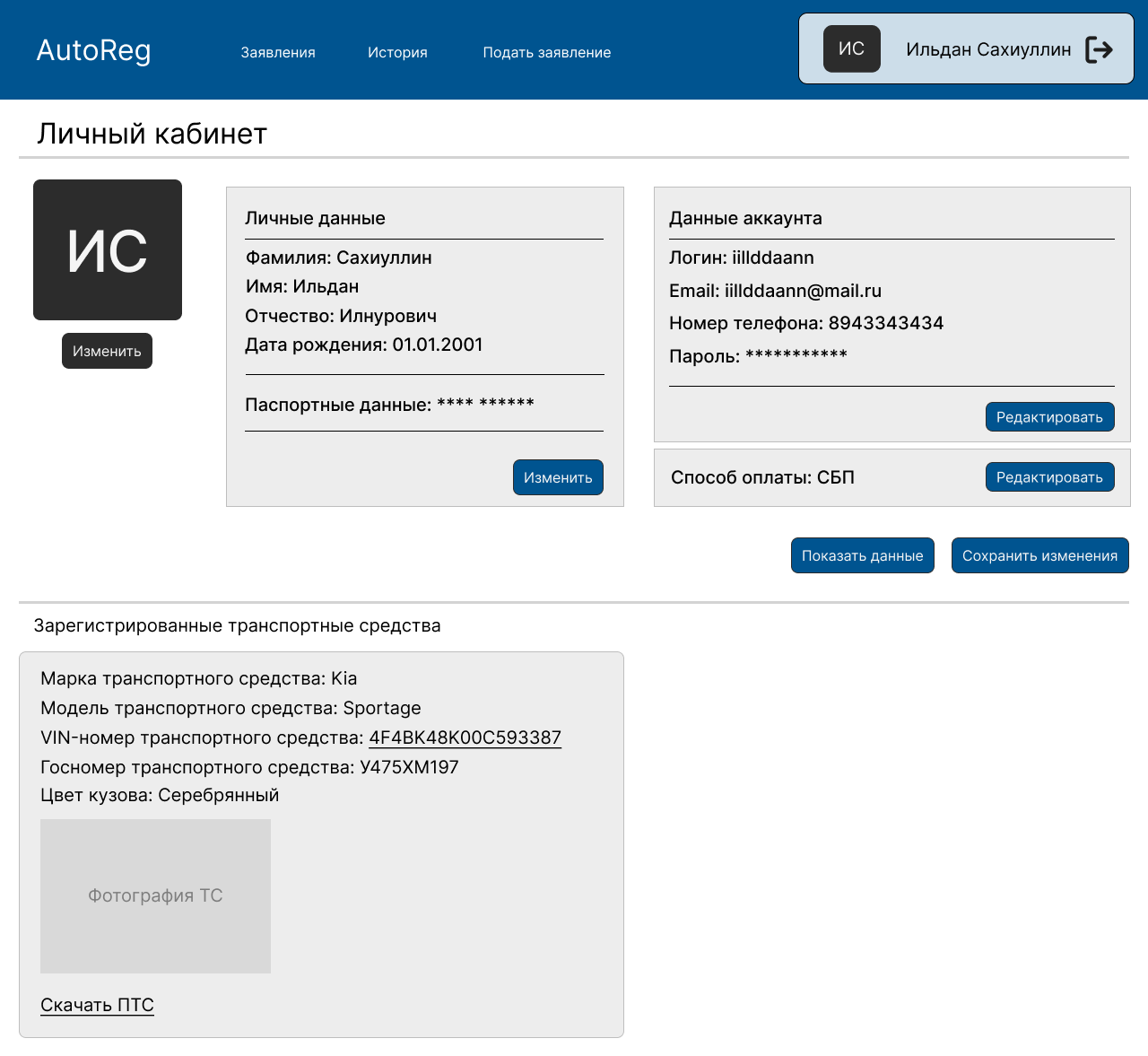


Рисунок 6.2 – Уведомление о некорретных данных

Для клиента есть возможность зарегистрироваться, чтобы это сделать необходимо нажать на текст «Зарегистрироваться» на странице «Авторизации». Отображается форма регистрации, где клиент должен заполнить требуемые поля и нажать на кнопку «Вход». Страница изображена на рисунке 6.3.

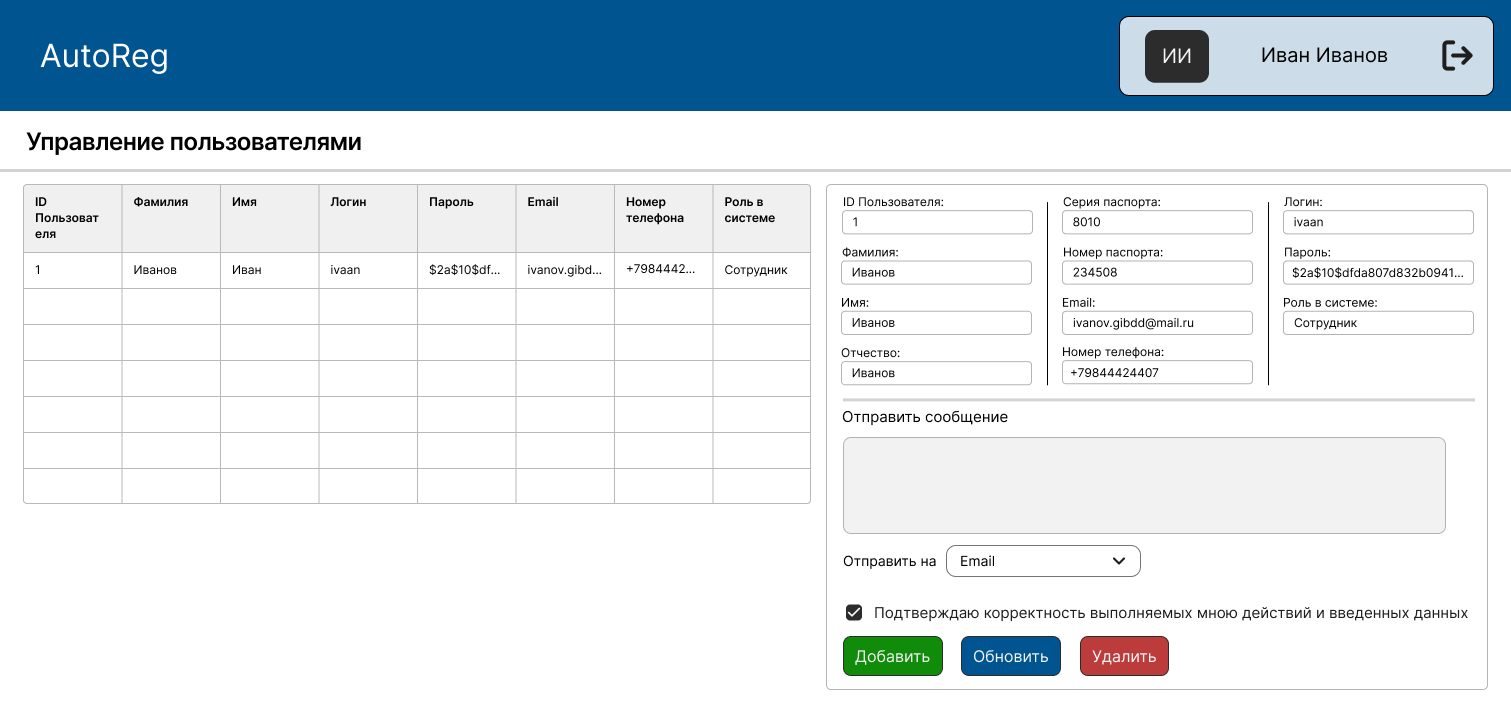
Рисунок 6.4 – Окно регистрации

После авторизации клиенту открывается страница личного кабинета, где содержится общая информация о клиенте. В окне есть возможность перейти на страницу просмотра текущих заявлений и архивных, а также на страницу для подачи заявления. Окно изображено на рисунке 6.5.

Рисунок 6.5 – Окно «личного кабинета»

Руководство для администратора.

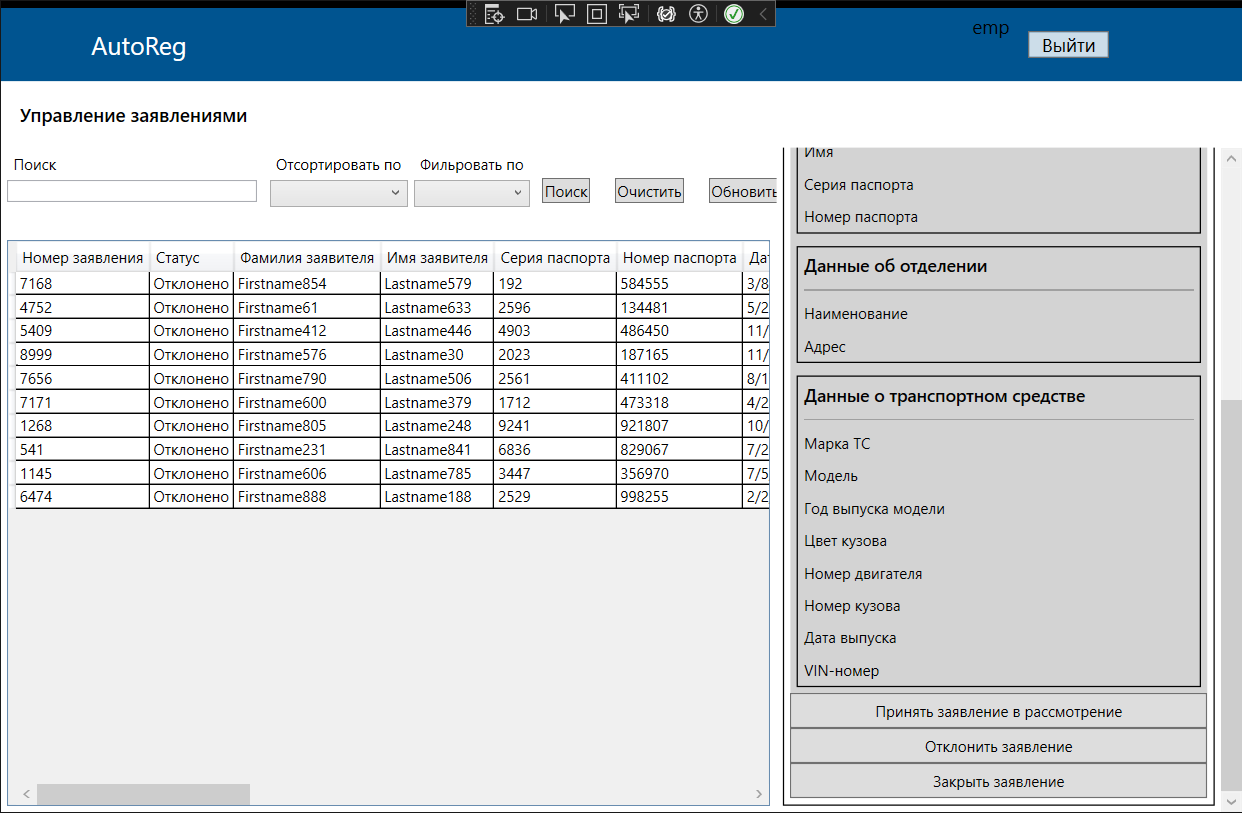
При успешной авторизации администратор попадает на окно для управления пользователями. Он может как добавлять новых пользователей, так и удалять их.

Рисунок 6.6 – Окно управления пользователями

При нажатии на строку в таблице с пользователями поля заполняться имеющимися данными о пользователе. При нажатии на кнопку «Удалить» появится сообщения для подтверждения удаления пользователя. После подтвержения, пользователь будет удален с базы данных информационной системы. Аналогичным образом работает и кнопка «Добавить», «Обновить».

Руководство для сотрудника

При успешной авторизации сотрудник попадает на окно для управления заявлениями. Он может просматривать заявления и внести определенные изменения в заявления, используя кнопки «Принять заявление в рассмотрение», «Отклонить заявление» и «Закрыть заявление», меняя тем самым статус заявления. Окно управления заявлениями представлен на рисунке 6.7.

Рисунок 6.7 – Окно управления заявлениями

ЗАКЛЮЧЕНИЕ

В результате проведенной работы было разработано приложение для автоматизации постановки автомобилей на учет в ГИБДД, соответствующее актуальным требованиям рынка и удовлетворяющее потребности пользователей системы в удобстве и быстром обслуживании. Приложение обладает интуитивно понятным интерфейсом, эффективными системами поиска и фильтрации. Успешная реализация проекта подтверждается проведением тестирования и внесением необходимых корректировок, обеспечивающих стабильную и надежную работу приложения. Разработка учитывала анализ существующих систем для учета постановки автомобилей в ГИБДД, позволяя избежать распространенных недостатков и реализовать преимущества лучших решений.