КОНКУРСНОЕ ЗАДАНИЕ

# ВВЕДЕНИЕ

**Dron Taxi** – стартап, специализирующаяся на надежной, комфортной и быстрой перевозке пассажиров с применением беспилотных летающих аппаратов.

Цель проекта – сделать передвижение людей максимально эффективным, безопасным и комфортным.

Задание состоит из 6 этапов:

1. Управления пользователями;
2. Управление заказами и транспортом;
3. Разработка веб-приложения для оформления заказа и регистрации;
4. Проектирование модуля управления обслуживанием и ремонтом транспорта;
5. Разработка документации.

# ЭТАП 1. УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ

## Основные объекты модуля

### **Пользователь**

Для корректной работы системы, пользователь должен содержать в себе следующие атрибуты:

1. Логин;
2. Пароль.

Вы можете добавлять атрибуты объекта «Пользователь», но перечисленные выше атрибуты должны быть реализованы.

**Обратите внимание, что пароли в системе не должны хранится в открытом виде и для их сокрытия требуется использовать « ».**

Управлением пользователями, ролями и выданными ролям функциями в Dron Taxi занимается администратор системы.

**Администратор** – это пользователь с расширенными полномочиями, которому системно устанавливается роль – «Администратор».

### **Роль**

В системе пользователи разделены по предоставляемым им ролям. Роль должна содержать в себе следующие атрибуты:

1. Системное имя;
2. Наименование.

При этом взаимодействие с ролями внутри системы осуществляется по системному имени, либо синтетическому идентификатору. Недопустимо использовать для этих целей наименование.

Одному пользователю может быть назначено несколько ролей, в зависимости от необходимости.

В свою очередь ролям должны быть предоставлены функции системы.

### **Функция системы**

**Функция системы** – это объект, отвечающий за разграничение доступа к тому или иному функционалу разрабатываемой системы.

Функция системы, как и роль должна содержать атрибуты

1. Системное имя;
2. Наименование.

Пример: Функция «Управление пользователями» должна давать доступ к вкладке приложения «Управление пользователями».

Ваш проект должен опираться именно на функции системы в интерфейсном разграничении доступа. Недопустимо использовать для этих целей роли пользователей напрямую.

## Процессы модуля

Модуль состоит из следующих процессов:

1. Авторизация;
2. Просмотр и редактирования профиля;
3. Управление пользователями;
4. Управление ролями и функциями.

### **Авторизация**

Первое, что нужно пользователю, чтобы войти в систему – это пройти процедуру авторизации.

Для этого пользователь открывает приложение, вводит логин и пароль.

Система проверяет данные на корректность и либо переходит на фрейм профиля, либо выдает сообщение об ошибке авторизации.

Так же, если пользователь нажал на кнопку «Запомнить меня» система должна запомнить логин пользователя и автоматически вставлять его в поле «логин» на форме авторизации при следующем подключении.

Макет окна авторизации представлен на рисунке 1

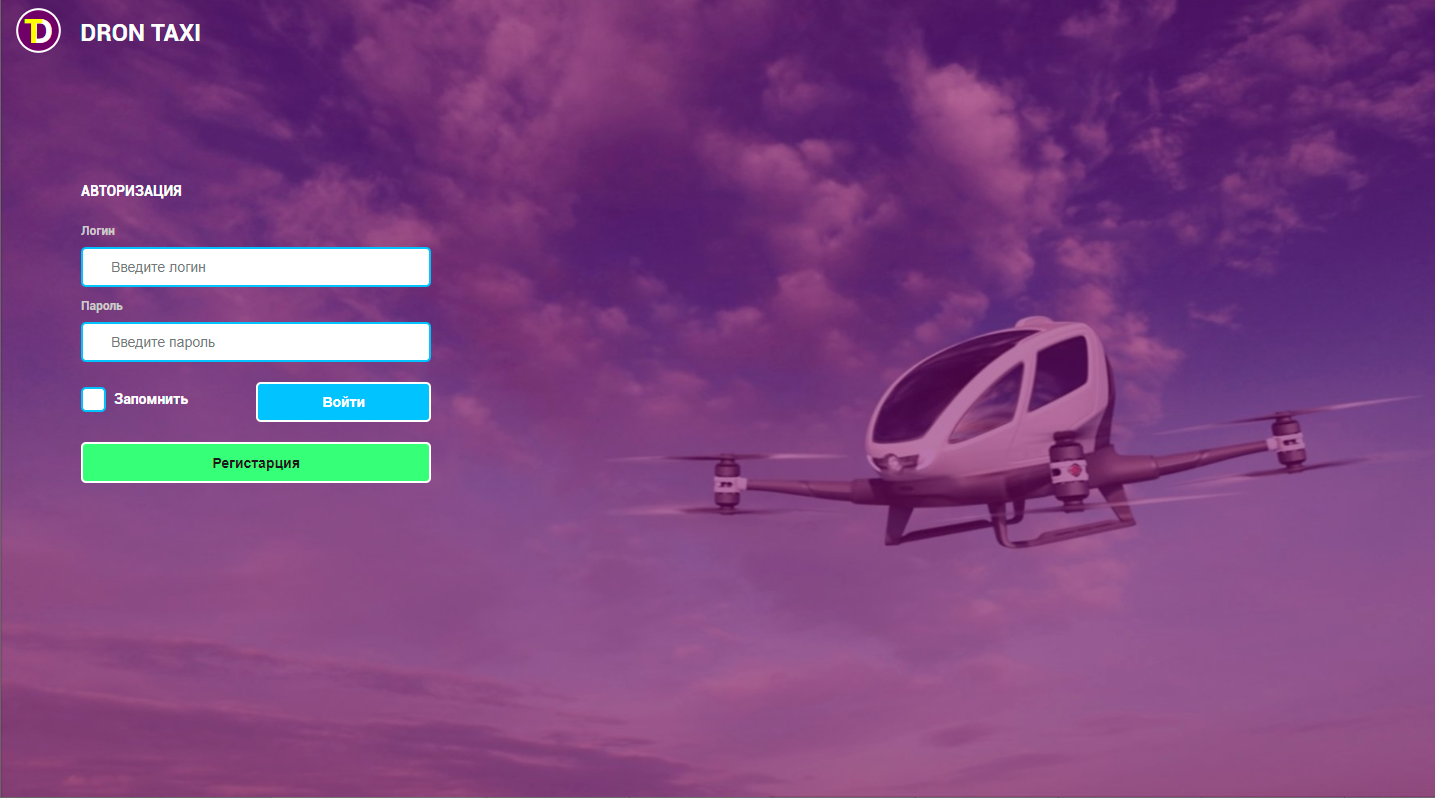


Рисунок 1. Окно авторизации

### **Просмотр и редактирование профиля**

После успешной процедуры авторизации пользователь попадает на страницу своего профиля.

Обратите внимание, что во всех окнах системы у пользователя есть левое меню, в котором отображается весь доступный пользователю функционал.

Кнопка перехода к тому или иному функционалу должна быть доступна только, если пользователь обладает соответствующей функцией.

Внутри профиля на вкладке личные данные, пользователь видит своё фото и данные, которые он указал о себе.

Макет окна профиля представлен на рисунке 2

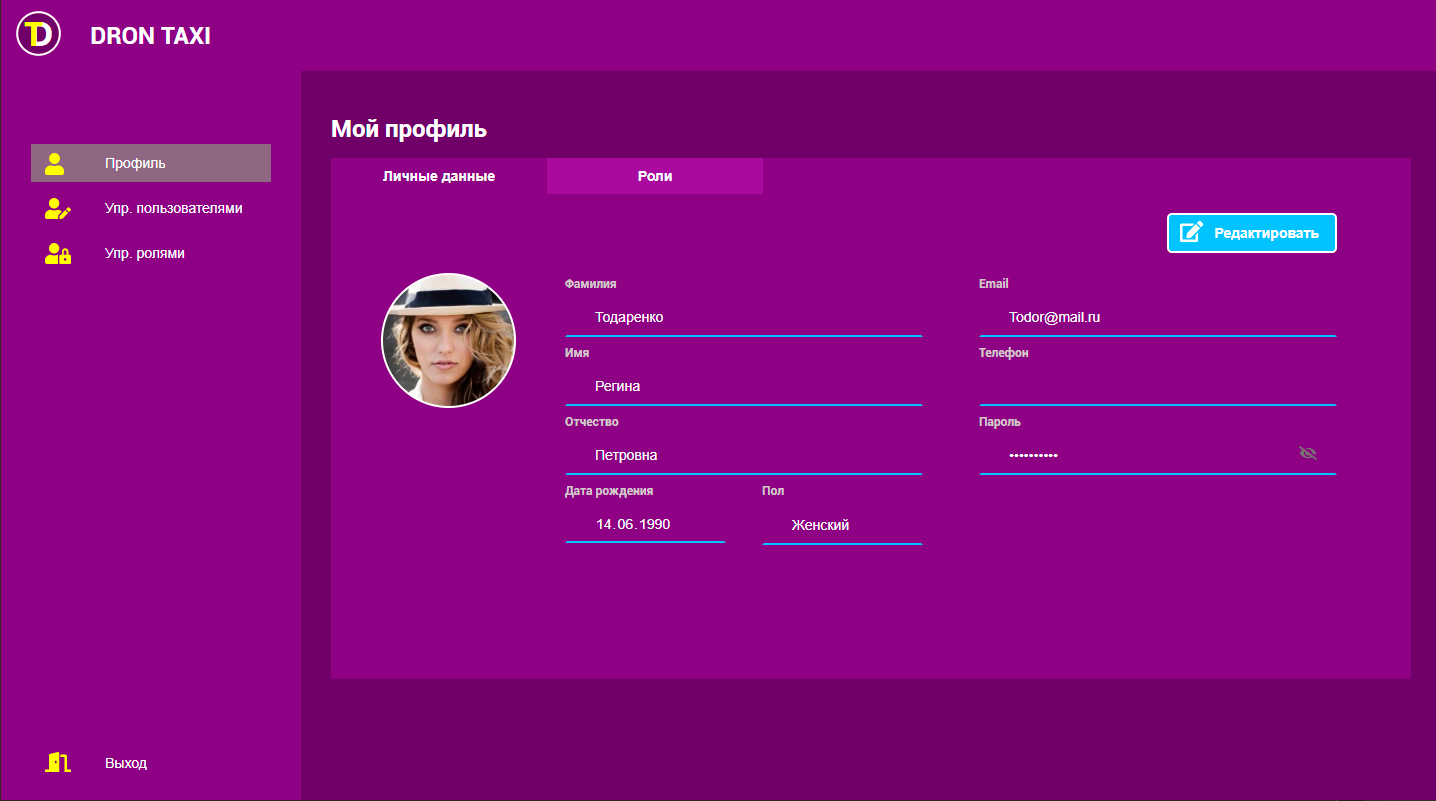


Рисунок 2. Профиль пользователя

Чтобы изменить данные, следует нажать кнопку «Редактировать», при этом форма трансформируется для открытия возможности редактирования.

Макет окна редактирования профиля представлен на рисунке 3.

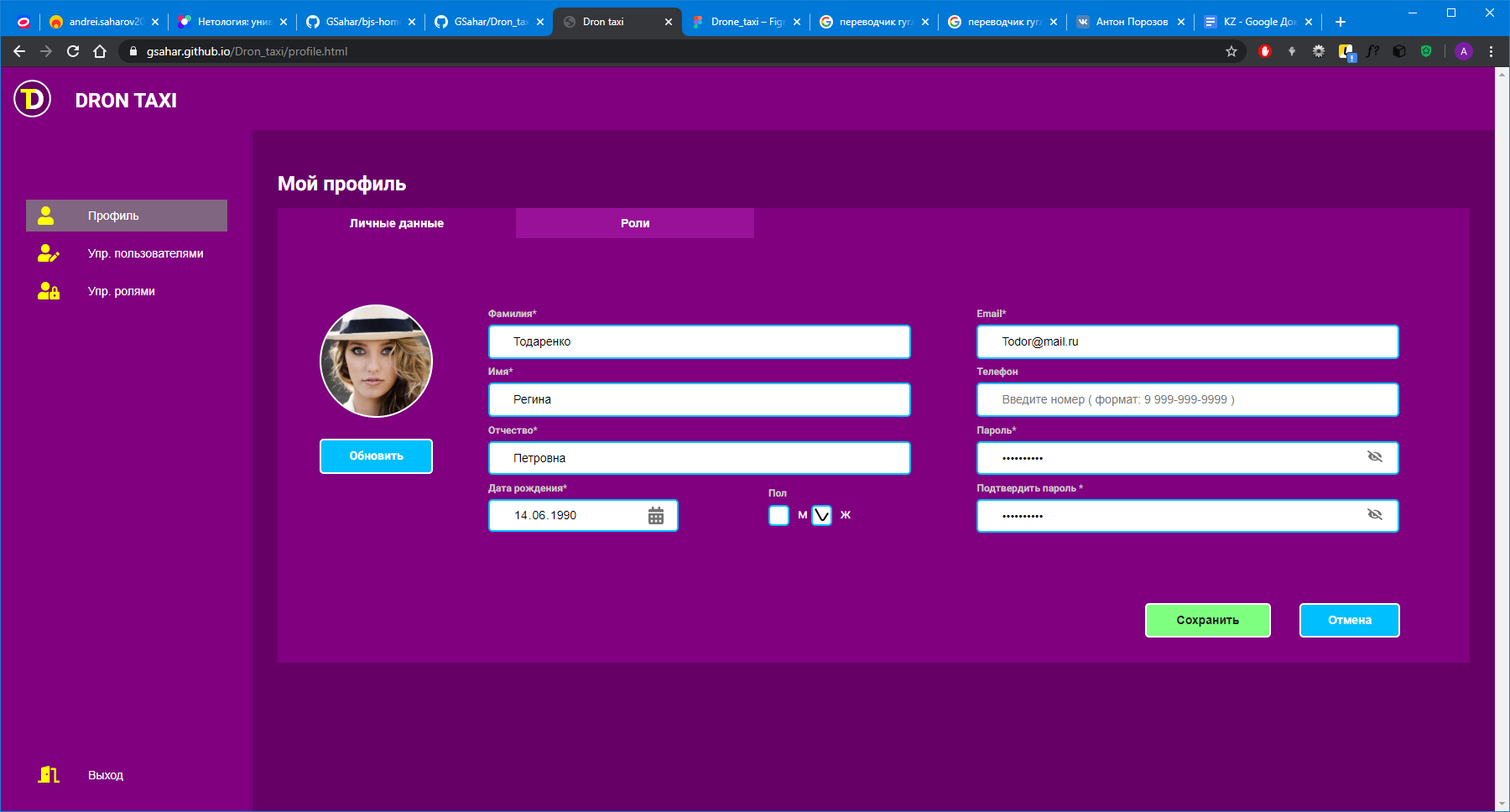


Рисунок 3. Редактирование профиля

После редактирования, при нажатии на кнопку «Сохранить», должна происходить проверка формы по следующим правилам:

1. Поле Email должно содержать корректный email. Например: as2020@mail.ru
2. Телефон должен быть введен по маске 9 999-999-9999
3. Пароль должен содержать:
   1. Минимум 1 символ в верхнем регистре;
   2. Минимум 1 цифру;
   3. Быть длиной не менее 6 символов.

При провале данных проверок, выдавать ошибку, в которой должны фигурировать все не пройденные проверки.

При переходе на вкладку «Роли», пользователь может посмотреть выданные ему роли.

При выборе в списке ролей, справа окна показывается карточка роли и доступные ей функции.

Пример вкладки ролей представлен на рисунке 4.

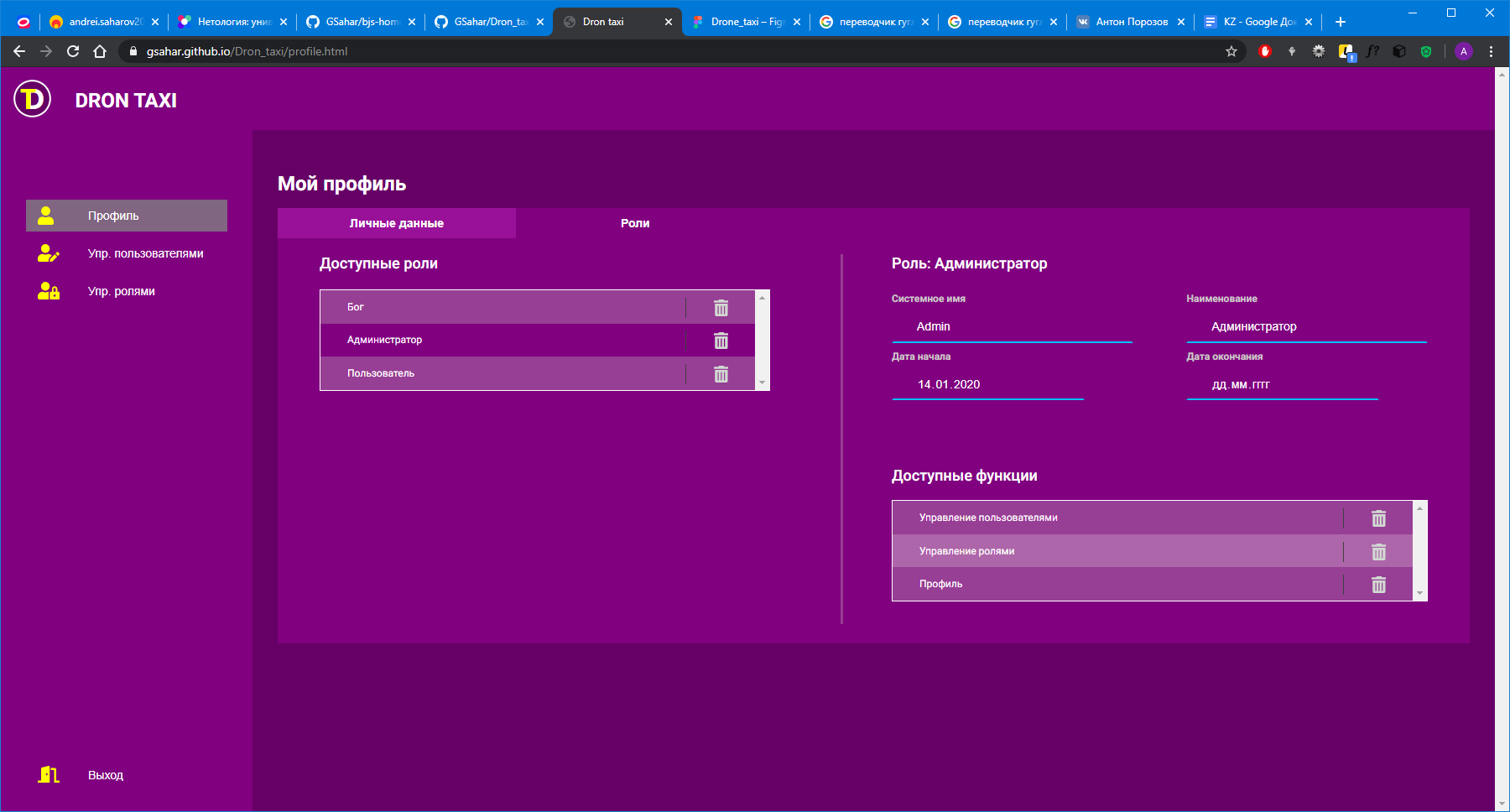


Рисунок 4. Вкладка роли в профиле пользователя

### **Управление пользователями**

При наличии функции для управления пользователями, пользователь имеет пункт меню «Управление пользователями» и может попасть в окно управления пользователями.

Окно представляет из себя список пользователей, с возможностью добавлять, изменять и удалять пользователей.

Пример окна представлен на рисунке 5

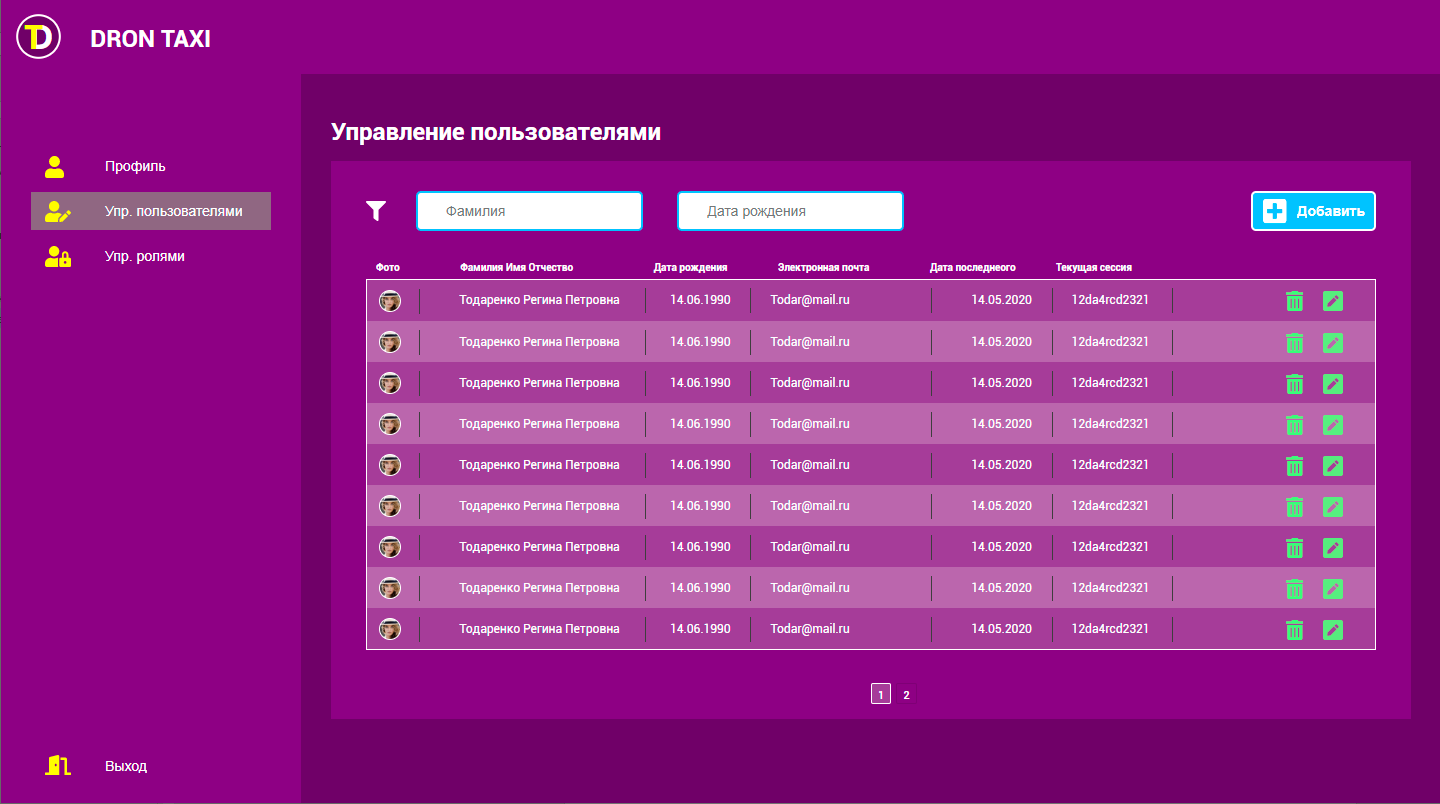


Рисунок 5. Управление пользователями

При добавлении пользователя вкладка меняется на окно добавления нового пользователя, так же, как показано на рисунке ниже

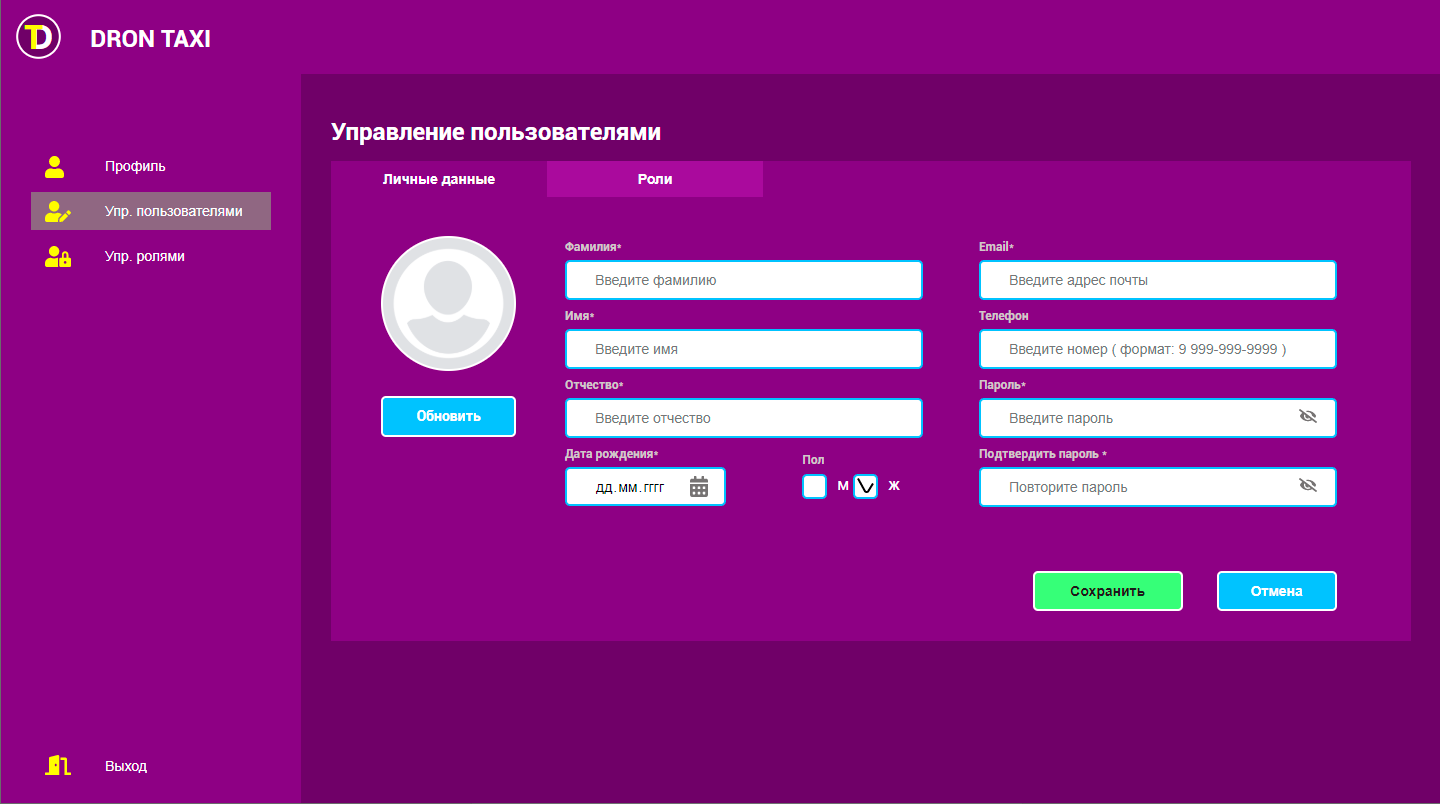


Рисунок 6. Добавление пользователя

При изменении пользователя интерфейс должен измениться как показано на рисунке 7.

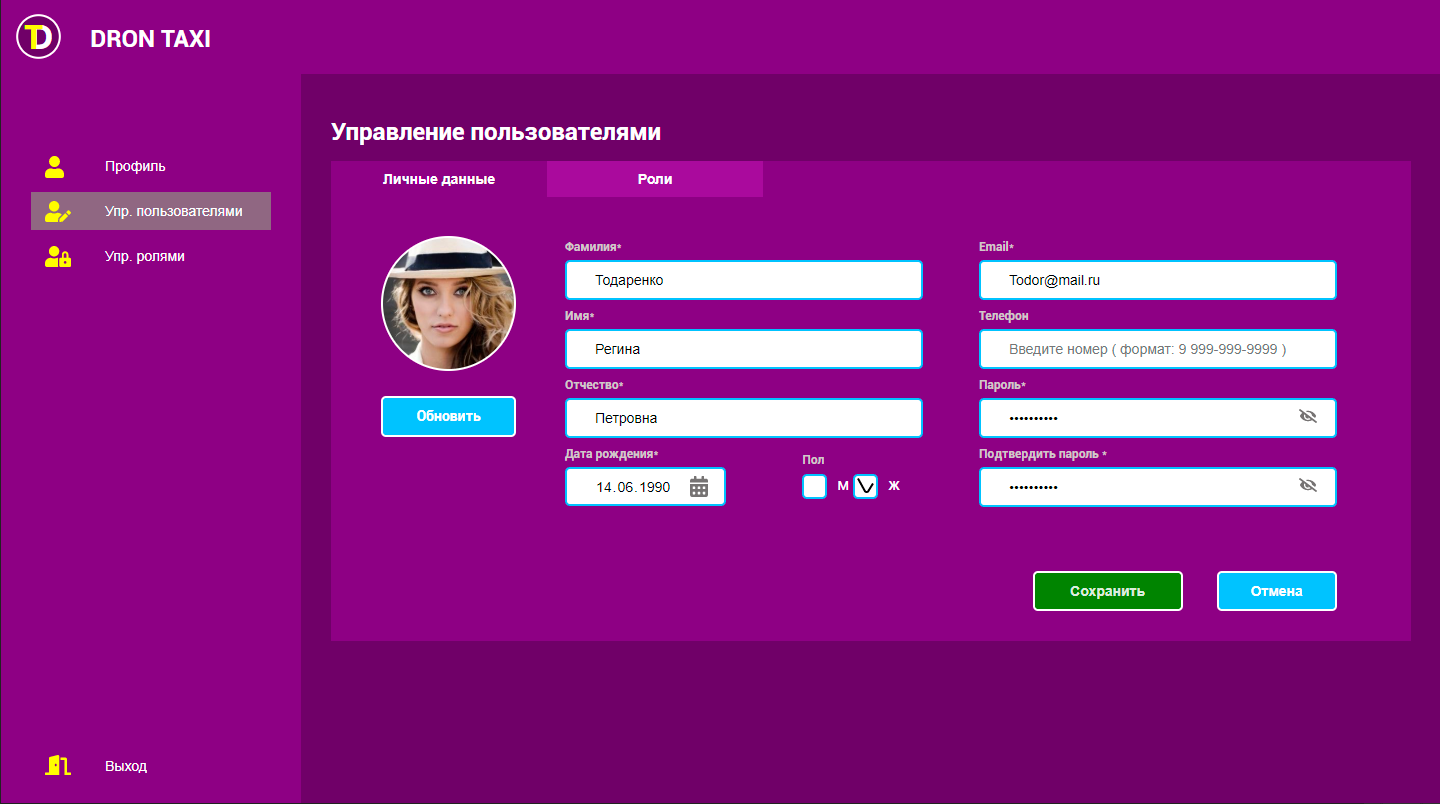


Рисунок 7. Изменение данных пользователя.

В обоих случаях (добавление и изменение) у пользователя возможно добавлять и убирать роли. Вкладка роли пользователя в этот момент выглядит, как показано на рисунке ниже

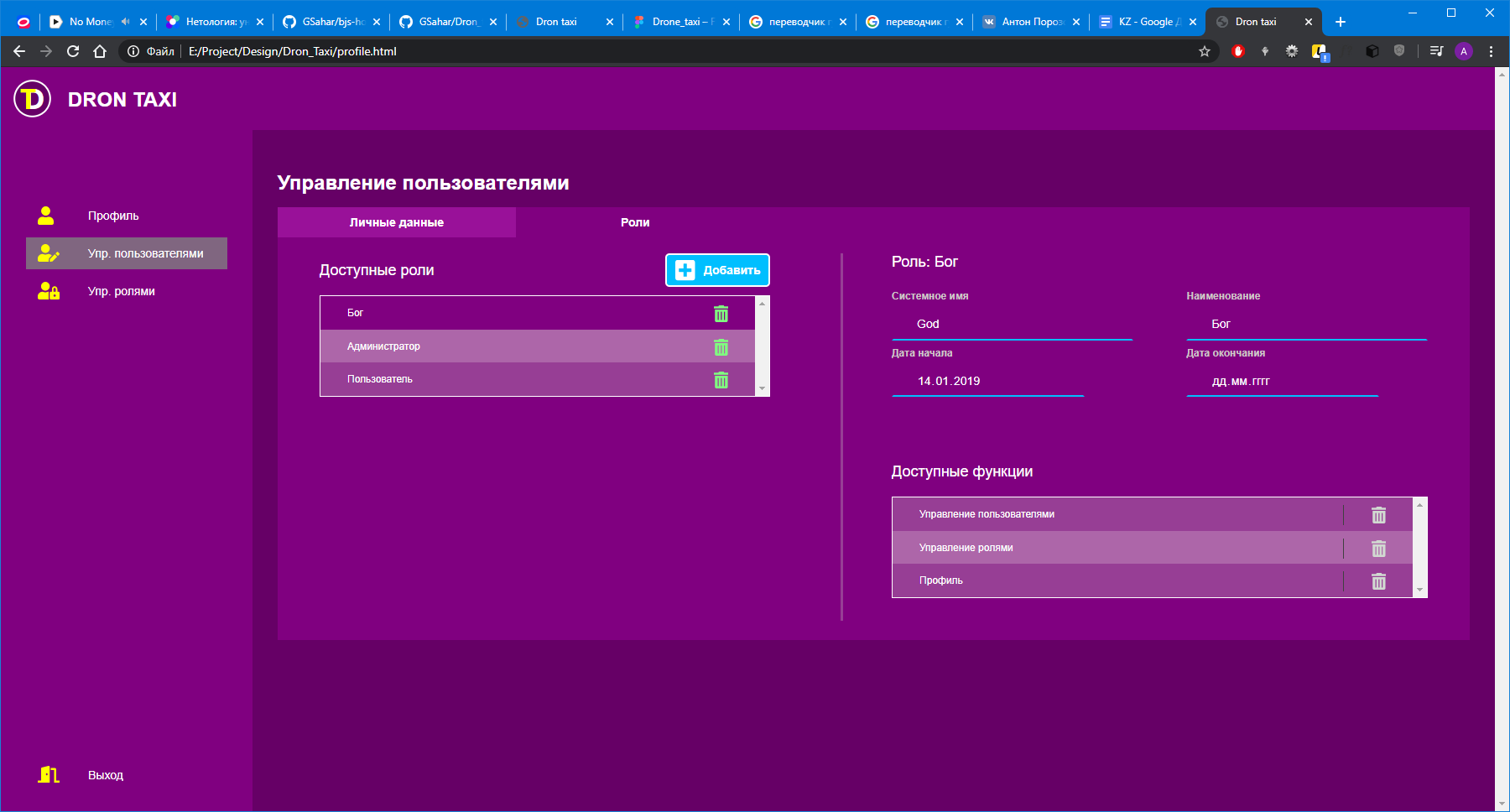


Рисунок 8. Вкладка роли в карточке пользователя при изменении.

При добавлении появляется модальное окно для выбора нужной роли. После чего фокус в списке ролей должен быть проставлен на добавленную роль.

При удалении у пользователя изымается выбранная в списке ролей роль.

При изменении пользователя у него сохраняются те же проверки, что и для полей пользователя в его профиле.

### **Управление ролями**

При наличии функции для управления ролями, пользователь имеет пункт меню «Управление ролями» и может попасть в окно управление ролями.

Окно «Управление ролями» представлено на рисунке 9

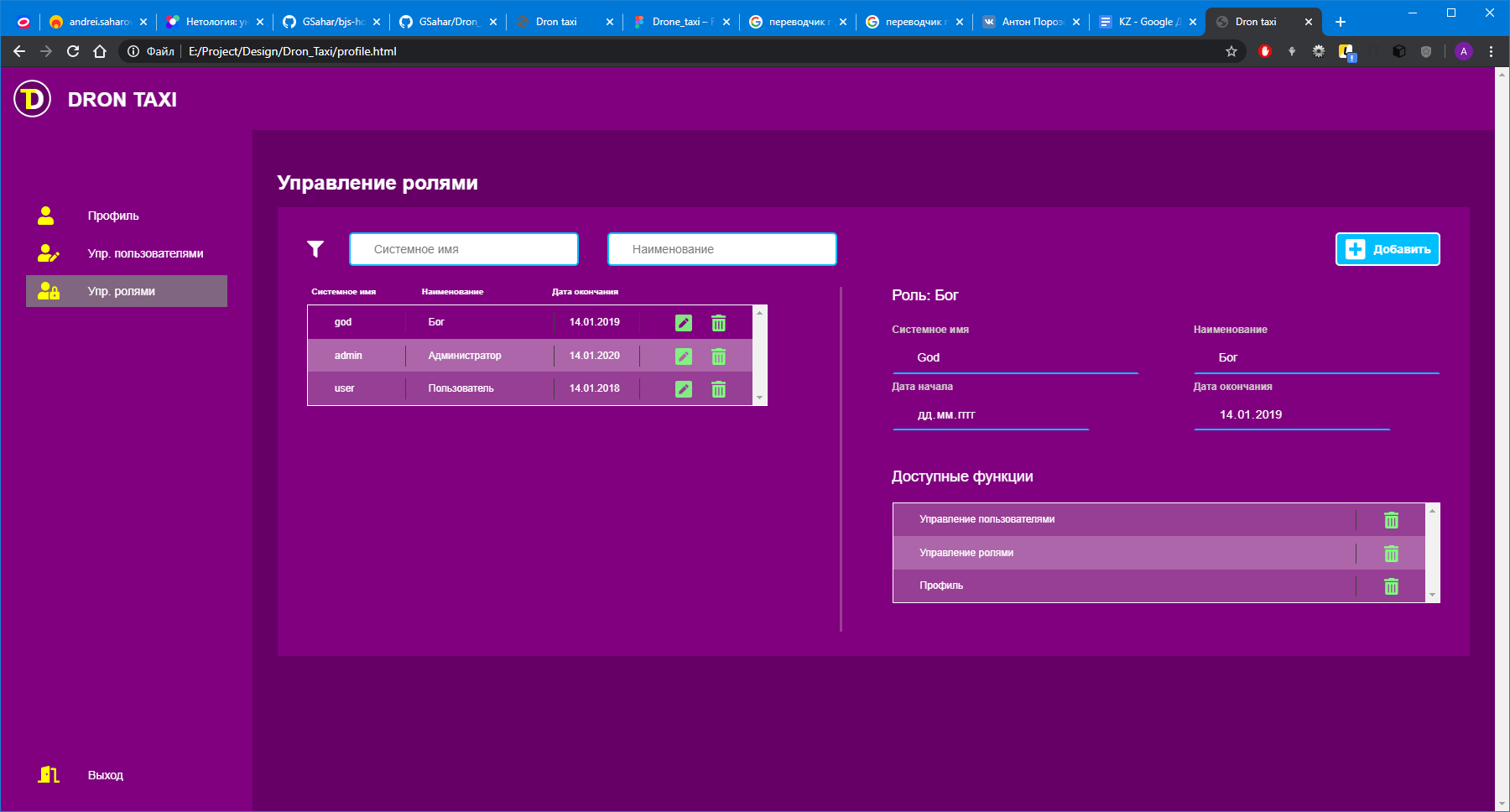


Рисунок 9. Основное окно для управления ролями

Данное окно позволяет пользователю создавать роли.

При нажатии на кнопку «Добавить», окно трансформируется для ввода новой роли как показано на рисунке 10.

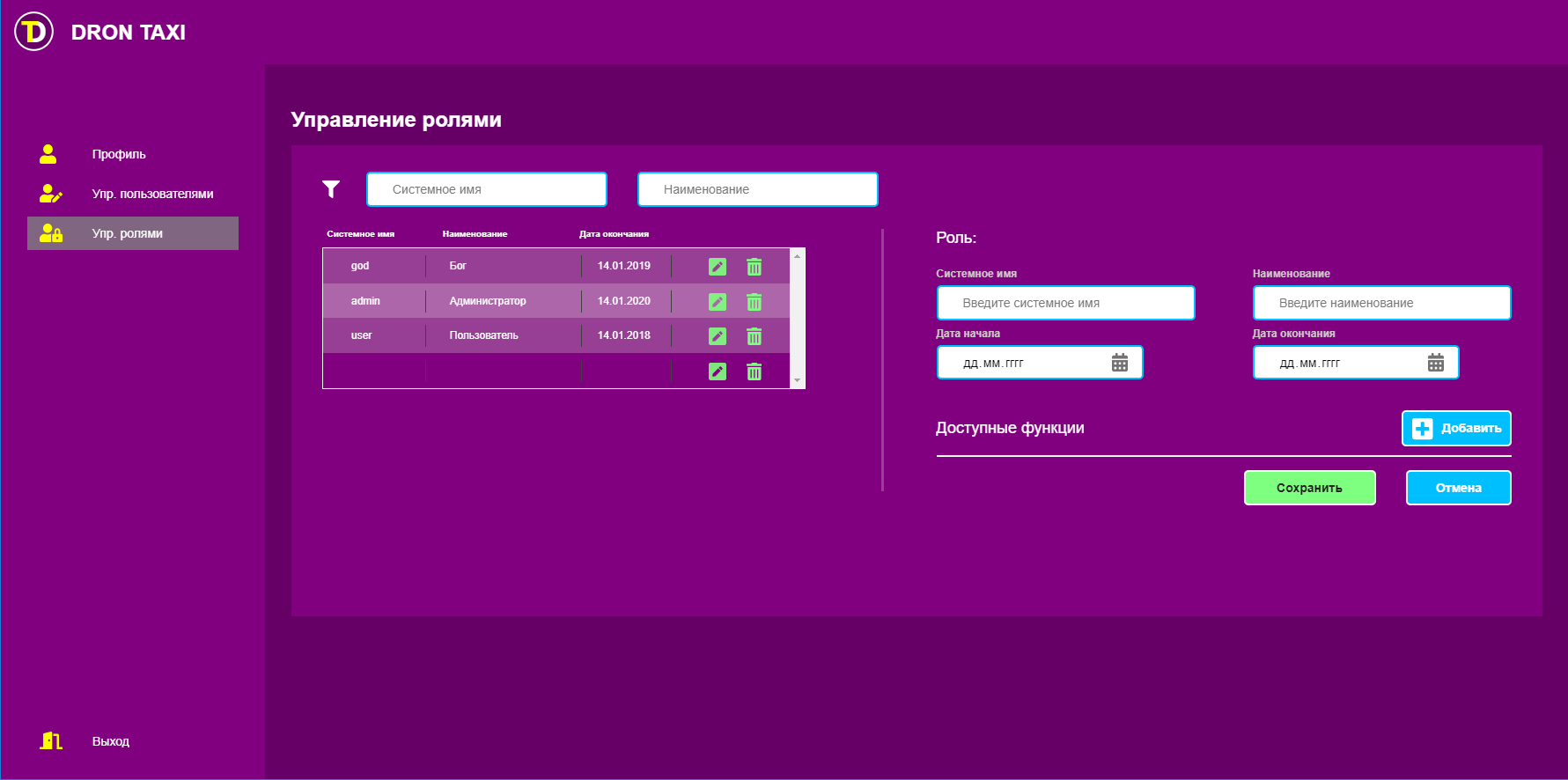


Рисунок 10. Добавление новой роли

При заполнении полей карточки роли, данные в выбранной строке ролей должны так же изменяться, как показано на рисунке 11.

Окончательно изменения фиксируются после нажатия на кнопку «Сохранить», если нажали на кнопку «Отменить» - изменения откатываются.

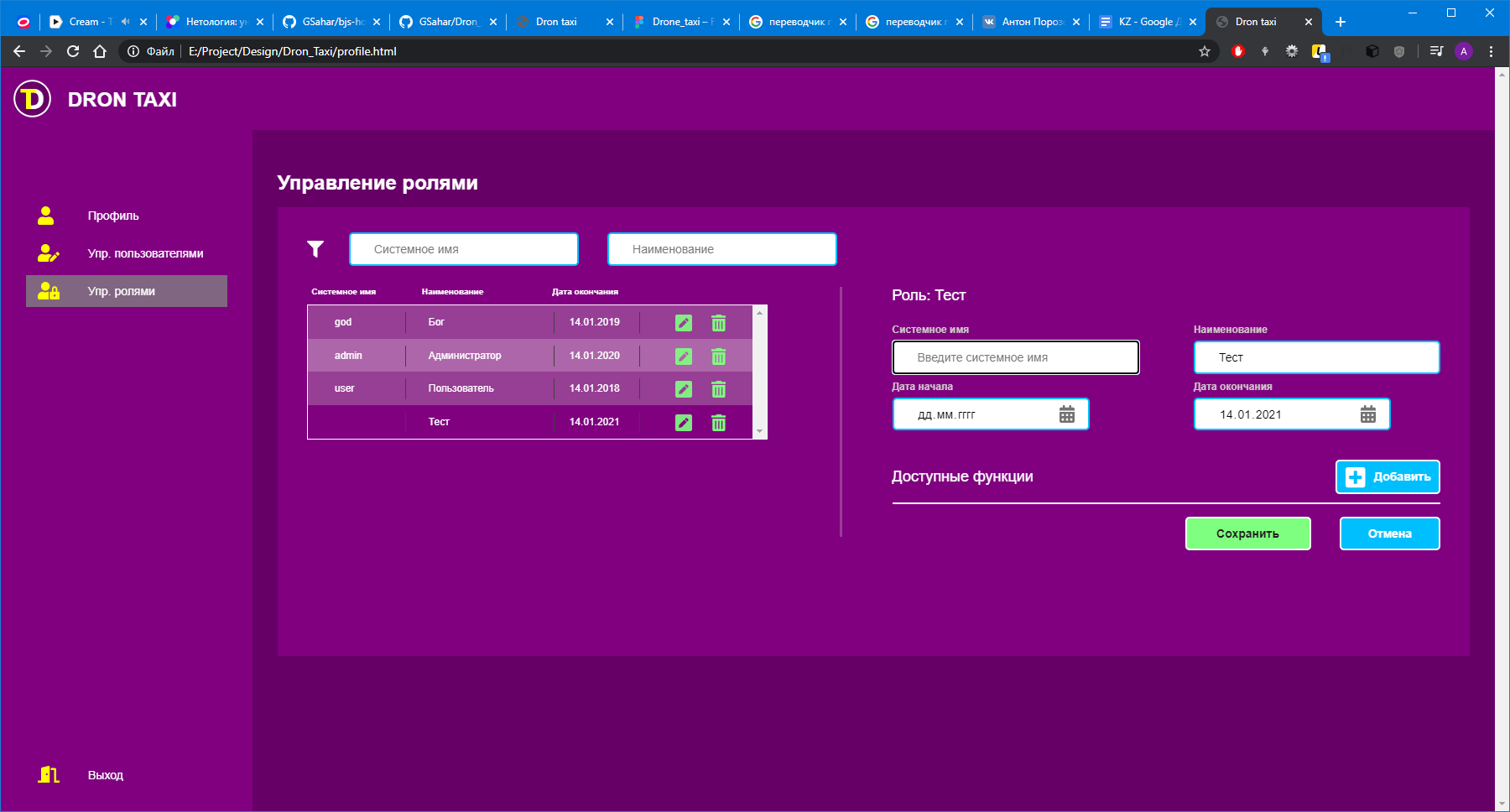


Рисунок 11. Заполнение данных новой роли

Модификация данных представлена на рисунке

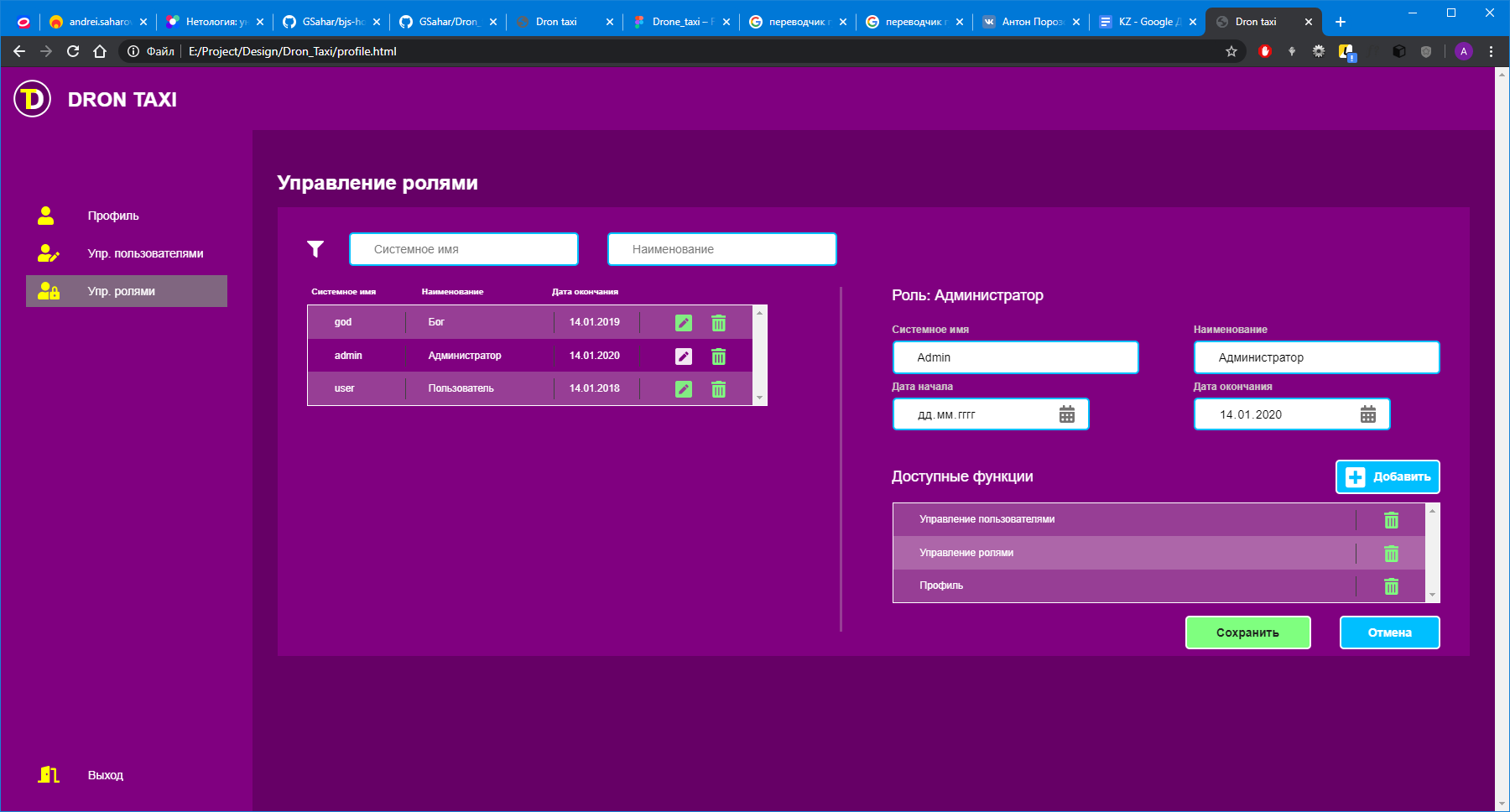


Рисунок 12. Модификация данных роли.

Так же, в данном экране можно добавлять функционал в роли. Пример окна показан на рисунке 13. Фиксация изменений выполняется по кнопке «Сохранить», рисунок 14.

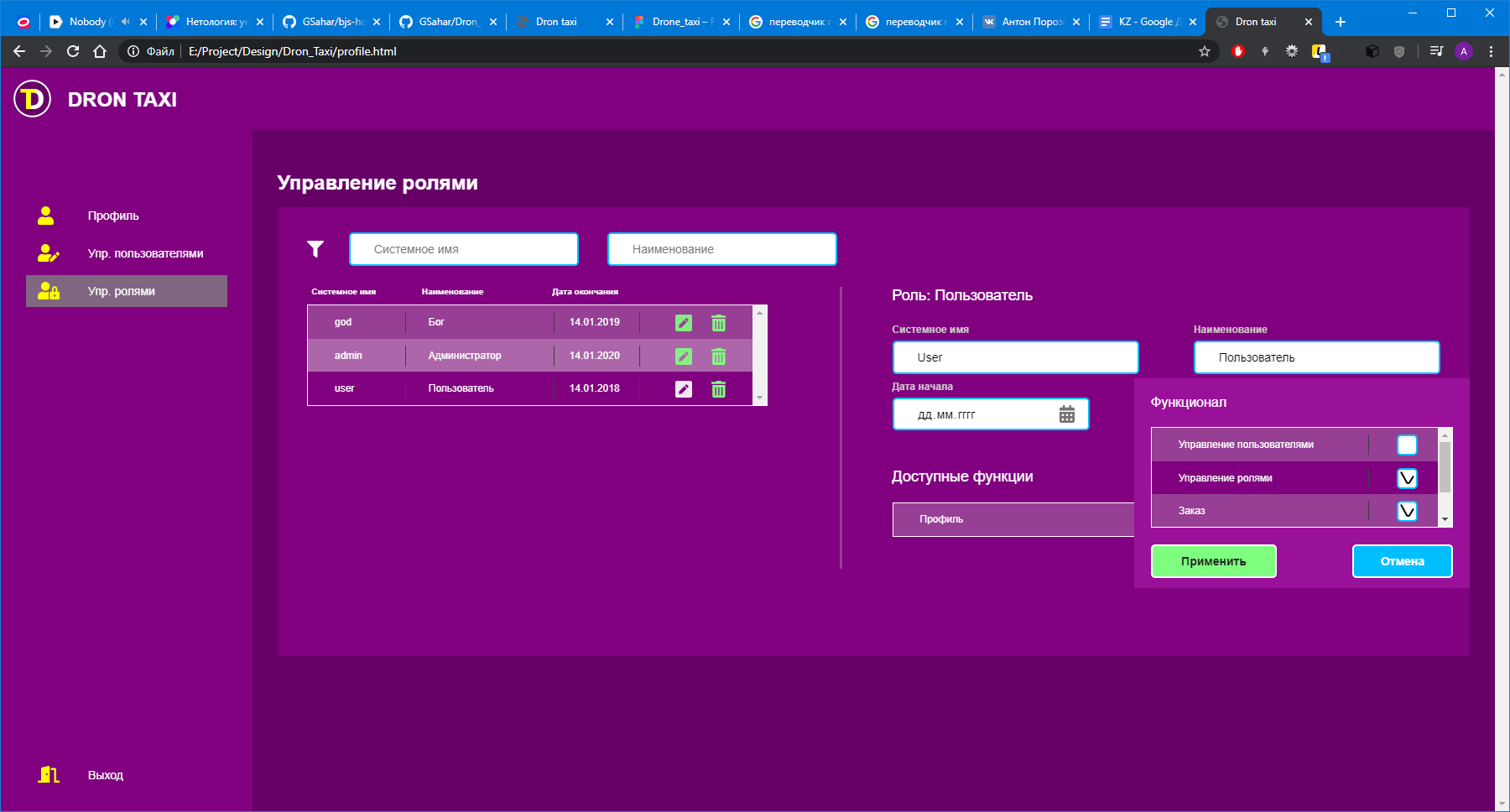


Рисунок 13. Добавление функций в роль.

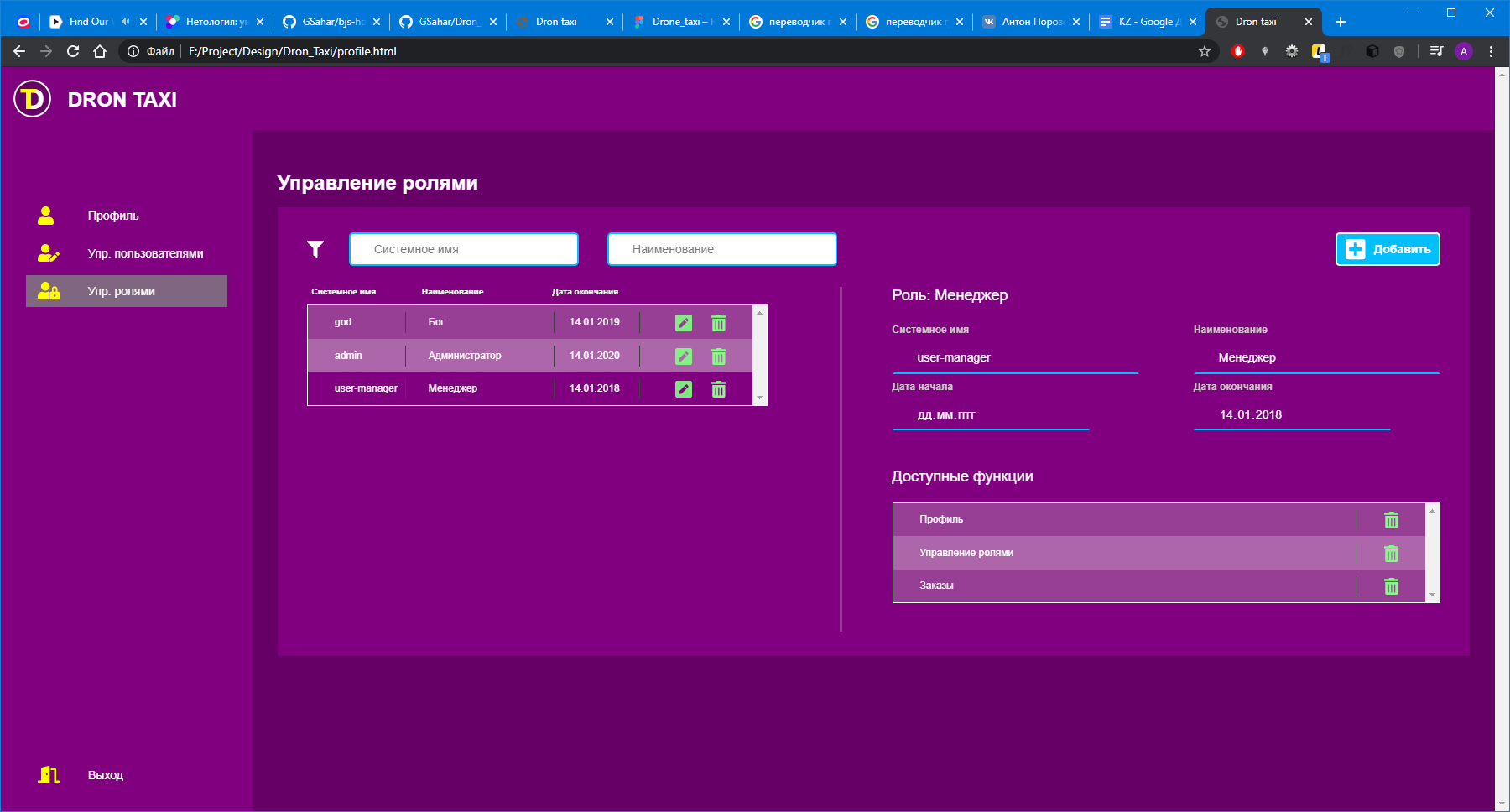


Рисунок 14. Зафиксированные изменения

# ЭТАП 2. УПРАВЛЕНИЕ ЗАКАЗАМИ И ТРАНСПОРТОМ

Для данного модуля не будет представлены примеры оформления окон. Вы должны сами определить их состав и концепцию по аналогии с модулями прошлого этапа и инструкции по стилю, для того, чтобы сохранить единообразие визуального представления системы.

## Основные объекты модуля

### **Заказ**

Компания отталкивается от того, что каждый клиент – это зарегистрированный пользователь системы. Поэтому, обязательными полями для объекта «Заказ», будет:

1. Номер заказа;
2. Дата/Время заказа;
3. Заказчик (пользователь);
4. Адрес точки отправления;
5. Адрес точки назначения;
6. Состояние заказа;
7. Выделенный транспорт;
8. Оператор (пользователь).

Заказ – отвечает на вопрос, когда, откуда, на чем и куда определенного пользователя доставила компания, а также предоставляет информацию о состоянии данного заказа и ответственном его исполнителе.

### **Транспорт**

Транспортные средства – это беспилотные летательные аппараты повышенной грузоподъемности.

На текущий момент существует всего 4 производителя подобных БПЛА (Northrop, DJI, Toyota, Airbus), но в системе должна быть возможность корректировки данного списка.

Каждый летательный аппарат должен содержать в себе следующие данные:

1. Марка;
2. Модель;
3. Год производства;
4. Регистрационный номер;
5. Дата регистрации;
6. Дата списания (должна быть заполнена, если списан).

Для каждого летательного аппарата требуется дать пользователю возможность вводить несколько фотографий.

## Процессы модуля

Модуль состоит из следующих процессов:

1. Принятие заказа;
2. Отмена заказа;
3. Закрытие заказа;
4. Добавление транспорта;
5. Изменение данных транспорта;
6. Списание транспорта.

Как и во всей системе, для получения доступа к вкладкам приложения требуется наличие определенных привилегий, представляющихся с помощью функций.

### **Принятие заказа**

Пользователь входит во вкладку «Управления заказами» и с помощью фильтров по дате заказа и состоянию выбирает активные заказы за нужный период времени и, уникально определив нужный заказ по номеру, входит в карточку заказа.

В карточке заказа пользователь проверяет основные параметры заказа, после чего выбирает свободный летательный аппарат и прикрепляет его к заявке и переводит заявку в состояние «Исполняется».

### **Отмена заказа**

Пользователь так же входит во вкладку «Управления заказами» и выбирает нужный ему заказ. После чего открывает карточку заказа и переводит его в состояние «Отменен».

### **Закрытие заказа**

После доставки пассажира, заказ автоматически переводится в состояние «Выполнен». Заказчик подтверждает в карточке своего заказа, что он выполнен и заказ переходит в состояние «Закрыт».

Если по каким-то причинам заказчик забыл это сделать, то оператор может из вкладки управления заказами выбрать интересующую его заявку, и перевести ее из состояния «Выполнен» в состояние «Закрыт».

### **Добавление транспорта**

Пользователь, переходит во вкладку «Управления транспортом», после чего нажимает на кнопку добавления транспортного средства, после чего вкладка трансформируется в карточку транспортного средства, где пользователь вводит основные атрибуты транспортного средства и нажимает кнопку сохранить. В этот момент производятся следующие проверки:

1. Для транспорта должно быть заведено не менее 2х фотографий
2. Марка, Модель, Год производства, Регистрационный номер и Дата регистрации – обязательные для заполнения поля
3. Регистрационный номер должен соответствовать шаблону *[Три латинские буквы в верхнем регистре][пробел][девятизначный номер, разделенный по три цифры знаком «-»].*

Пример: **ABV 992-123-983**

### **Изменение данных транспорта**

Пользователь, переходит во вкладку «Управления транспортом», после чего выбирает с помощью фильтров по регистрационному номеру, марке или модели нужный летательный аппарат.

Далее пользователь переходит в карточку летательного аппарата, где производит нужные изменения и нажимает кнопку «Сохранить». После чего производятся следующий проверки:

1. Для транспорта должно быть заведено не менее 2х фотографий
2. Марка, Модель, Год производства, Регистрационный номер и Дата регистрации – обязательные для заполнения поля
3. Регистрационный номер должен соответствовать шаблону *[Три латинские буквы в верхнем регистре][пробел][девятизначный номер, разделенный по три цифры знаком «-»].*

Пример: **ABV 992-123-983**

### **Списание транспорта**

Каждый летательный аппарат должен служить не более 5 лет, после чего его необходимо списать, если транспорт подлежит списанию, в списке летательных аппаратов, в окне управления транспортом статус транспортного средства должен быть «Требуется списание», а также такой летательный аппарат должен быть выделен контрастным цветом во всех списках, чтобы оператор не мог выпустить его в полет.

Для списания пользователю достаточно нажать правой кнопкой мыши на выбранном в списке летательном аппарате и выбрать операцию «Списать», после чего аппарат будет списан, а в карточке летательного аппарата появится дата списания.

# ЭТАП 3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

## Авторизация и регистрация

### **Авторизация**

Авторизация должна повторять авторизацию в Desktop версии приложения.

Для этого пользователь открывает приложение, вводит логин и пароль.

Система проверяет данные на корректность и либо переходит на фрейм профиля, либо выдает сообщение об ошибке авторизации.

Так же, если пользователь нажал на кнопку «Запомнить меня» система должна запомнить логин пользователя и автоматически вставлять его в поле «логин» на форме авторизации при следующем подключении.

Макет окна авторизации представлен на рисунке 15

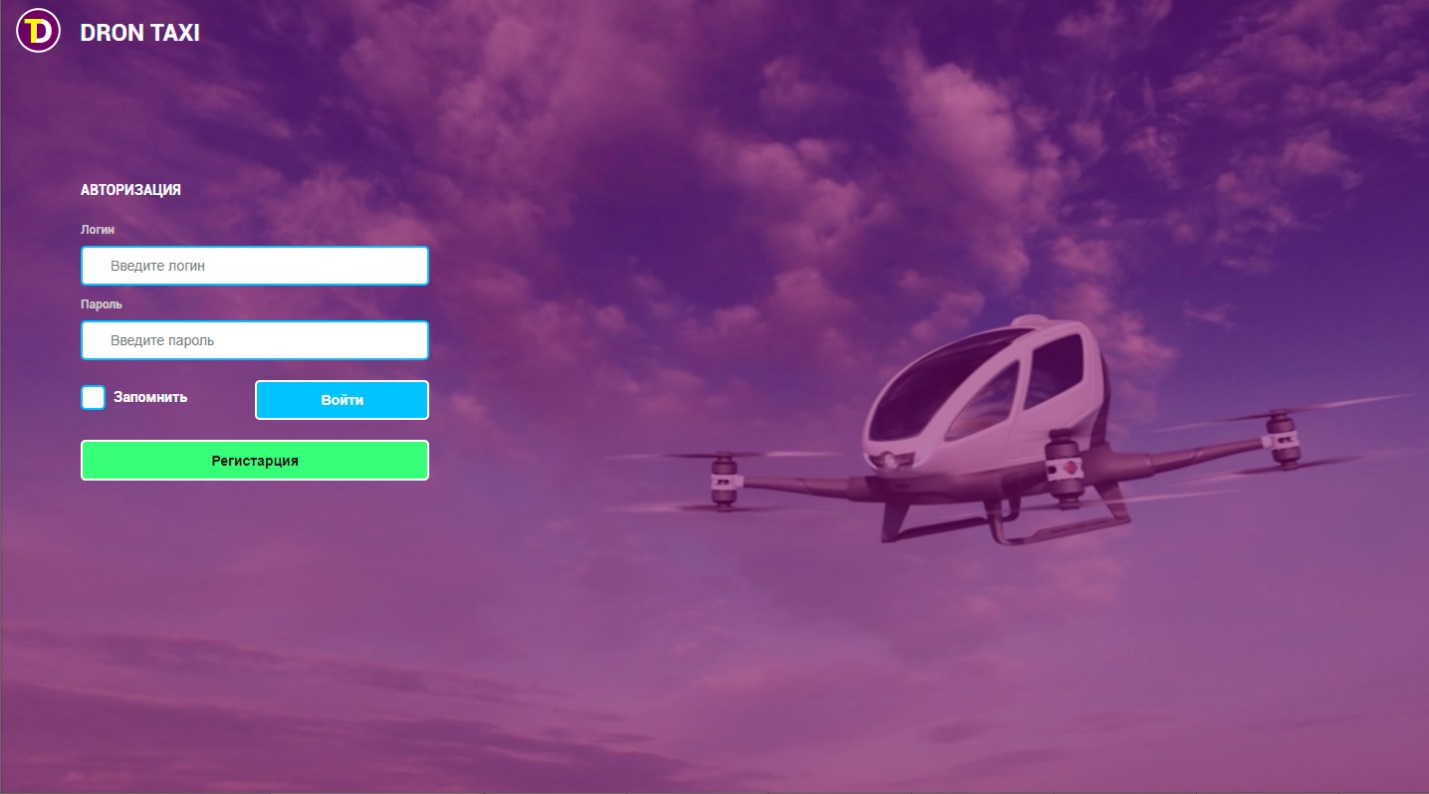


Рисунок 15. Окно авторизации

При нажатии на кнопку «Регистрация», пользователь должен перейти в окно регистрации нового пользователя.

### **Регистрация**

После перехода на окно регистрации пользователь вводит все необходимые поля, и нажимает на кнопку «Сохранить».

Макет формы регистрации представлен на рисунке 15.

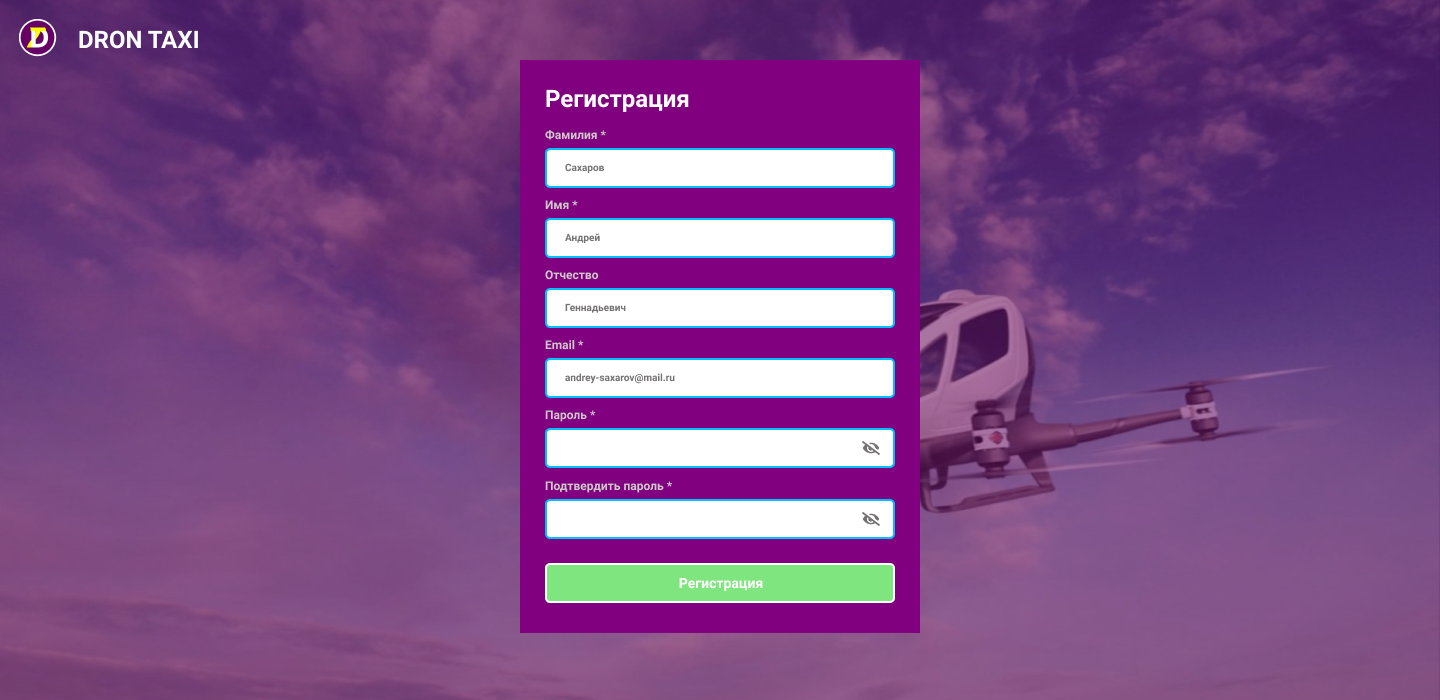


Рисунок 16. Форма регистрации.

При нажатии на кнопку «Сохранить» производятся следующие проверки:

1. Поля Фамилия, Имя, Email, Пароль, Подтверждение пароля – обязательны для заполнения.
2. Поле Email должно содержать корректный email. Например, as2020@mail.ru
3. Пароль должен содержать:
   1. Минимум 1 символ в верхнем регистре;
   2. Минимум 1 цифру;
   3. Быть длиной не менее 6 символов.
4. Поле «Подтверждение пароля» должно совпадать с паролем

## Профиль пользователя

После прохождения авторизации или регистрации пользователь должен попадать на свою страницу, где он может изменять детали своего профиля, аналогично процессу просмотр и редактирование профиля в этапе 1.

## Мои заказы

Авторизированный пользователь должен иметь доступ ко вкладке «Мои заказы», где будут представлены все сделанные пользователем заказы.

### **Регистрация заказа**

При переходе на вкладку «Мои поездки» пользователь видит список созданных им заказов в различных состояниях.

Для формирования нового заказа пользователю достаточно нажать на кнопку «Новая поездка», при этом откроется карточка заказа. Макет карточки представлен на рисунке ниже

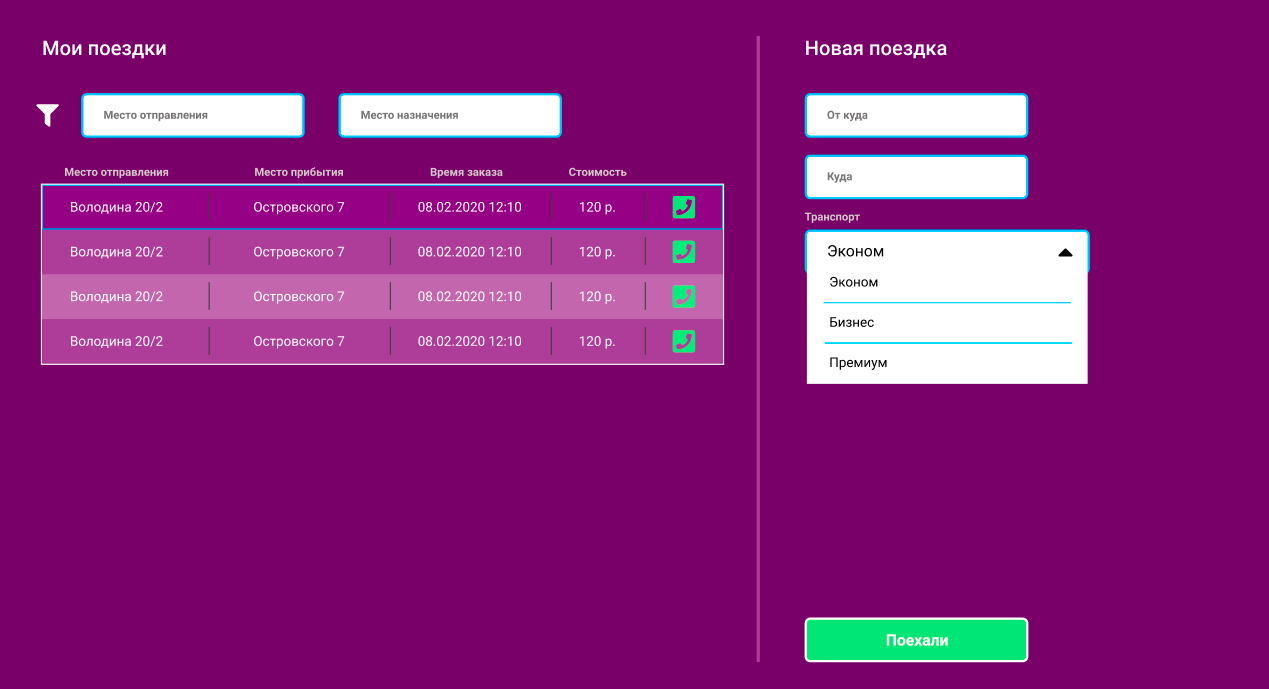


Рисунок 17. Мои поездки

В карточке заказа пользователь вводит точку отбытия и точку назначения, выбирает тип транспорта и нажимает кнопку «Поехали».

При нажатии кнопки «Поехали» происходят следующие проверки:

1. Поля «Откуда», «Куда» и «Транспорт» должны быть заполнены

При корректном заполнении формы формируется заказ в состоянии «Активен», после чего оператор рассмотрит заказ и отправит транспорт клиенту.

### **Отмена заказа**

Если пользователь сделал ошибку и ему нужно отменить заказ, он может выделить необходимую поездку в списке, нажать правую кнопку мыши, вызвав контекстное меню и выбрать пункт «Отменить».

Если заказ находится в состояниях «Активен», или «Исполняется», операция должна перевести его в состояние отменен. При этом беспилотник, если он был направлен к пользователю, должен вернуться обратно.

Пользователь не должен иметь возможности отменить исполненные и отмененные заказы.

# ЭТАП 4. ПРОЕКТИРОВАНИЕ БЛОКА РЕМОНТОВ И ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

## Описание этапа

На этом этапе вы должны перепроектировать блок для учета транспортных средств, добавив в них возможность учета ремонта и технических осмотров беспилотников.

## Процесс

Все начинается с закупки беспилотника. После его покупки, ответственный за транспорт регистрирует его в системе, используя присланный поставщиком паспорт летательного аппарата со следующими обязательными атрибутами:

1. Марка;
2. Модель;
3. Год производства;
4. Регистрационный номер.

Вместе с этим, в паспорте могут быть и другие необязательные атрибуты. Система должна позволять заводить разные наборы атрибутов для каждого конкретного паспорта.

После регистрации, беспилотник будет отправлен на техническую комиссию и первое техническое обслуживание.

Мастер смены ремонтного департамента создает акт комплектности беспилотника, в котором описывается, из каких составных узлов и агрегатов состоит беспилотник и определяет цикл ремонтных работ и технических осмотров каждого из них.

**Цикл ремонтных работ** – это периодический цикл ремонта или замены агрегата летательного аппарата. Периодичность бывает двух видов:

* По времени работы (часы), например, 1 раз в 1000 ч.
* По дальности полета (километры), например, 1 раз в 5000 км.

**Цикл технических осмотров** – это периодический цикл осмотров агрегатов летательного аппарата, по результатам которого принимается решение о замене или ремонте тех или иных агрегатов. Периодичность осмотров бывает трех видов:

* По времени работы (часы), например, 1 раз в 100 ч.
* Периодически, через какой-либо промежуток времени, например, каждую неделю, или каждый день.
* По дальности полета (километры), например, 1 раз в 5000 км.

После регистрации мастер впервые осматривает агрегаты летательного аппарата. Если все в норме, акт технического осмотра закрывается с отметкой «Агрегат исправен» и беспилотник может быть выпущен для работы с пассажирами.

Если в ходе осмотра летательного аппарата вскрылись дефекты, то на основании акта осмотра должен быть создан акт ремонта, в котором в качестве спецификации должен быть осматриваемый мастером агрегат.

Мастер должен провести определенный тип ремонта (малый, средний, капитальный), зависящий от характера неисправности (легкая неисправность, средняя неисправность, большая неисправность).

После проведения ремонта и закрытия акта, цикл ремонтируемого агрегата должен сброситься и начать отсчет следующего ремонта с момента закрытия акта.

## Результат

Результатом вашей работы на этом этапе должны стать:

1. Разработанная диаграмма использования (use case);
2. Разработанная диаграмма состояний (statechart diagram);
3. Разработанная диаграмма последовательности (sequence diagram);
4. Разработанная диаграмма сущность-связь (ER-диаграмма);
5. Разработанный словарь данных.

# ЭТАП 5. РАЗРАБОТКА ДОКУМЕНТАЦИИ

На данном этапе вам требуется создать общую документацию по проекту.

Документация должна наиболее полно описывать весь функционал вашей системы, для того, чтобы заказчик мог пользоваться ей.

Хорошей практикой при написании документации будет использование предоставленных в задании процессов.

Помимо документации, вы должны создать инструкцию для развертывания вашей системы на ПК для проверки. Для этого вам понадобится создать следующие документы:

1. Пул скриптов для развертывания в СУБД;
2. Инструкцию по развертыванию скриптов в СУБД;
3. Инструкцию по установке клиентской части desktop приложения;
4. Инструкцию по развертыванию backend части веб-приложения;
5. Инструкцию по развертыванию frontend части веб-прложения;
6. Общий файл readme, описывающий ваш репозитарий.

**Будьте внимательны!** *Если, следуя вашим инструкциям эксперт по проверке не сможет развернуть ваш проект – эта часть задания будет аннулирована автоматически. Проверяйте ваши конечные сборки на работоспособность.*