**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”  
(УНИВЕРСИТЕТ ИТМО)**

**ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

**Разработка системы контроля доступа транспортных средств в автомобильный паркинг, основанной на компьютерном зрении.**

Автор Гайдукевич Иван Евгеньевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество) (Подпись)

**Центр авторизованного обучения информационным технологиям**

Наименование программы **«Python-разработчик»**

Руководитель Кузьмин Константин Михайлович \_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

**К защите допустить**

|  |  |  |
| --- | --- | --- |
| *Заместитель директора ЦАО ИТ,*  *к.т.н.* |  | / *Т.В. Зудилова*/ |

Санкт-Петербург, 2023г.

Обучающийся Гайдукевич Иван Евгеньевич Группа 124/02

(Фамилия, И. О.)

Работа принята «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Работа выполнена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Секретарь ИАК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, И. О.) (подпись)

Листов хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Демонстрационных материалов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Оглавление**

|  |  |
| --- | --- |
|  | **Стр.** |
| **Оглавление** | **3** |
| **Введение** | **4** |
| **Цели работы** | **5** |
| **Стек технологий для реализации проекта** | **6** |
| **Состав и структура проекта** | **8** |
| **Действия пользователей программы (приложения)** | **9** |
| **Структура данных предметной области** | **14** |
| **Структура данных, используемая для аутентификации и**  **авторизации в приложении** | **15** |
| **Функции системы** | **16** |
| **Практическая значимость** | **18** |
| **Заключение** | **19** |
| **Список литературы** | **20** |
| **Приложение №1** | **21** |

**Введение**

В современном мире обработка и анализ графической информации сталкиваются с растущей потребностью в автоматизации и оптимизации процессов. В этом контексте компьютерное зрение играет существенную роль, позволяя компьютерам анализировать и интерпретировать изображения и видео на основе алгоритмов и моделей машинного обучения.

Актуальность использования компьютерного зрения проявляется во многих сферах деятельности, начиная от медицины и промышленности до полноценной робототехники. Эта технология уникальным образом сочетает в себе преимущества скорости, точности и масштабируемости, превращая неструктурированные данные в ценную информацию для принятия решений.

Компьютерное зрение также может существенно улучшить сферу безопасности. Уже существуют алгоритмы и модели нейронных сетей, которые помогают выявлять людей с преступными умыслами по их движениям, разыскивать преступников в толпе, контролировать огромные периметры с помощью видеонаблюдения без помощи оператора. Компьютерное зрение можно использовать и для решение бытовых задач, связанных с безопасностью. Об одной из таких задач и пойдет речь в данной работе.

Многие из нас живут в современных жилых комплексах, имеющих свои крытые, подземные, отдельностоящие паркинги или хотя бы огороженную территорию для парковки около дома. Доступ в такие паркинги осуществляется одним из следующих способов:

1. с помощью ключа, rfid-карты или любого другого физического устройства доступа
2. с помощью консьержа, который пускает в паркинг или выпускает из него транспортные средства, записанные в бумажном или электронном журнале.
3. с помощью «умных камер», умеющих распознавать государственные регистрационные знаки.
4. контроль доступа не осуществляется в принципе.

Слабое место в безопасности первого способа – утеря, забывание, подделка ключей. Слабое место в безопасности второго способа – человеческий фактор.

Что касается третьего способа, то данный метод встречается крайне редко. Из 12 осмотренных паркингов жилых комплексов так и не удалось обнаружить ни одного, где бы применялась данная технология. Так как технология «умных камер» построена на применении компьютерного зрения, с точки зрения безопасности, она не имеет тех недостатков, которые имеют первые два способа доступа в помещения паркингов. У нее нет ключей, ее не отвлечь и не дать ей взятку.

# Цели работы

* Изучить существующие python-библиотеки для реализации технологии компьютерного зрения.
* Разработать систему контроля доступа транспортных средств в автомобильный паркинг жилого комплекса, основанную на технологиях компьютерного зрения.

**Стек технологий для реализации проекта**

# Для реализации проекта использовался язык программирования Python 3.10, высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

# Для сборки проекта использовался IDE PyCharm — кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains. Предоставляет разработчику удобный комплекс средств для написания кода, работы с БД и визуальный отладчик.

Для написания модуля осуществляющего технологии компьютерного зрения для определения государственных регистрационных знаков использовались библиотеки **OpenCV 4.8 и EasyOCR 1.7.** Данный стек технологий был подобран в результате исследования похожих технологий компьютерного зрения и выбран как самый удобный в использовании и не имеющий конфликтов.

В работе над проектом использовалась система контроля версий **Git**, была выбрана, по причине того, что является стандартом в данной области. В качестве оболочки для **Git** использовался репозиторий **GitHub**, дающий бесплатные возможности по размещению исходного кода проекта.

Для создания веб-приложения использовался веб-фреймворк **Django 4**, который дает удобные механизмы по быстрой разработке, имеет систему аутентификации и авторизации из коробки, а также мощный одноименный шаблонизатор для создания фронтенда.

Для создания шаблонов применялся язык гипертекстовой разметки **HTML5**, чтобы придать приложению современный и актуальный по дизайну вид использовались каскадные таблицы стилей **CSS3**, а именно фреймворк **Bootstrap 5**.

Для хранения данных веб-приложения используется **PostgreSQL 12** - свободная объектно-реляционная система управления базами данных (СУБД). PostgreSQL базируется на языке **SQL** и поддерживает многие из возможностей стандарта SQL:2011. Данная система была выбрана из-за отсутствия явных конфликтов с фреймворком **Django**

Также для реализации проекта потребовались:

* две Full-HD веб-камеры Logitech, которые использовались для создания видеопотока.
* комплект отснятых видеороликов в формате mp4 с транспортными средствами для тестов системы распознавания.

**Cостав и структура системы**

****

****

Запрос номера ТС и разрешений в БД, журналирование въездов выездов

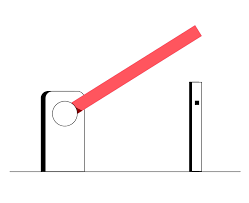
Модуль для управления доступом в паркинг с CV

app DJANGO Camera

(поиск, обработка, валидация данных на табличках гос. рег. знаков)

Видеопоток

****

****

Открыть/Закрыть

Ответ о разрешении въезда/выезда

PostgreSQL хранение данных о зарегистрированных в паркинге ТС

Модуль для администрирования системы

App DJANGO Scip

Аутентификация, авторизация администратора, просмотр журналов въезда/выезда, CRUD ТС, CRUD владельцев, включение камер

Создание, изменение, удаление, чтение сущностей

Приложение размещается на веб-сервере

1. Камера въезда (ip или usb)
2. Камера выезда (ip или usb)
3. Ворота с микроконтроллером, который может принимать сигнал об открытии или закрытии.
4. Веб-сервер с запущенным веб-приложением, которое состоит из двух модулей: camera – осуществляющий определение информации написанной на государственных регистрационных знаках, а также управляющий доступом в паркинг, scip – модуль для администрирования системы.
5. Сервер БД PostgreSQL для развертывания базы данных, которая будет использоваться для хранения данных предметной области.

**Действия пользователей программы (приложения)**

В базовой версии приложения выделяется три группы пользователей системы:

1. **Администратор приложения** - осуществляет предварительную настройку системы в виде создания аккаунтов консьержей.

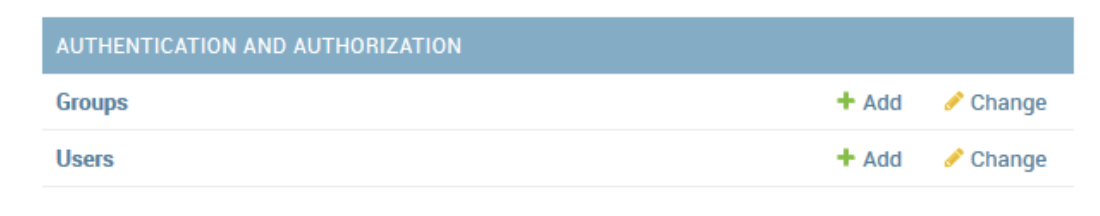
Для создания аккаунтов используется встроенная система администрирования, которую предоставляет фреймворк Django.

Данный функционал доступен по ссылке.

<http://название_ресурса/admin>

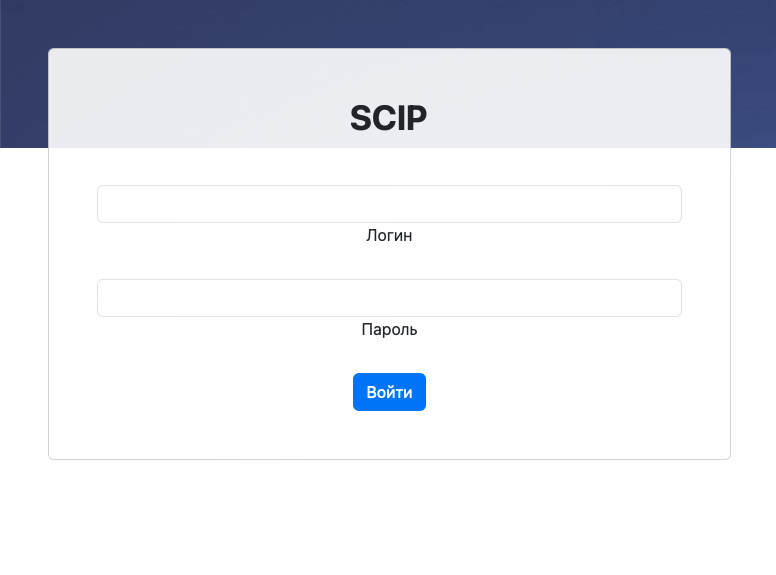
Первично администратор заходит под стандартным логином и паролем, предоставляемым разработчиком системы. В последующем он должен поменять учетные данные для безопасной работы системы.

Администратор создает аккаунты консьержей через создание группы консьержей и конкретных юзеров.



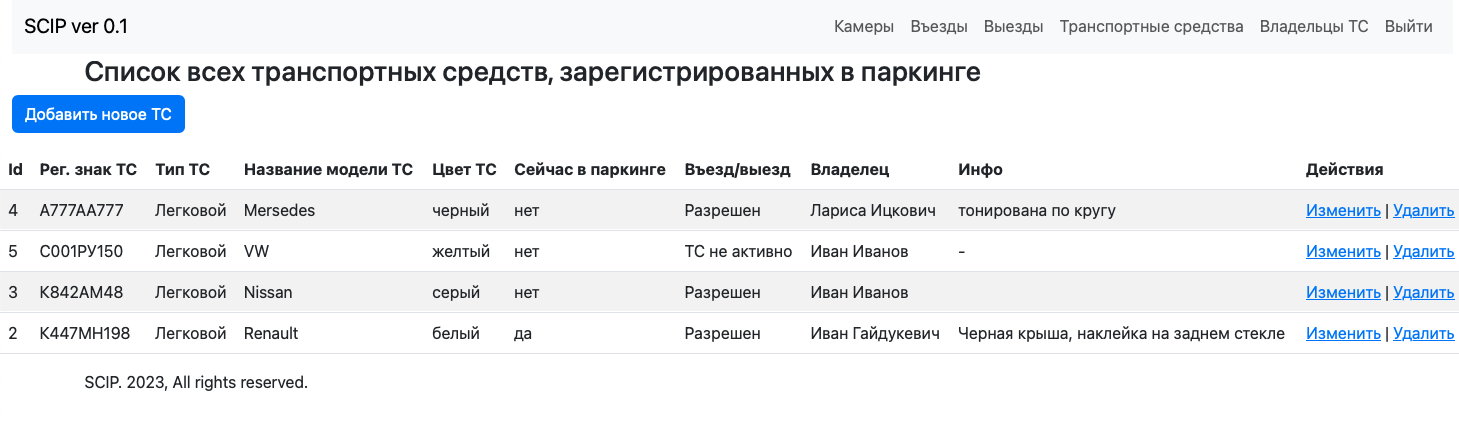
1. **Консьерж** – пользователь, которому доступен вход в защищенную часть приложения с помощью логина и пароля.

Он осуществляет вход в систему по ссылке <http://название_ресурса/>

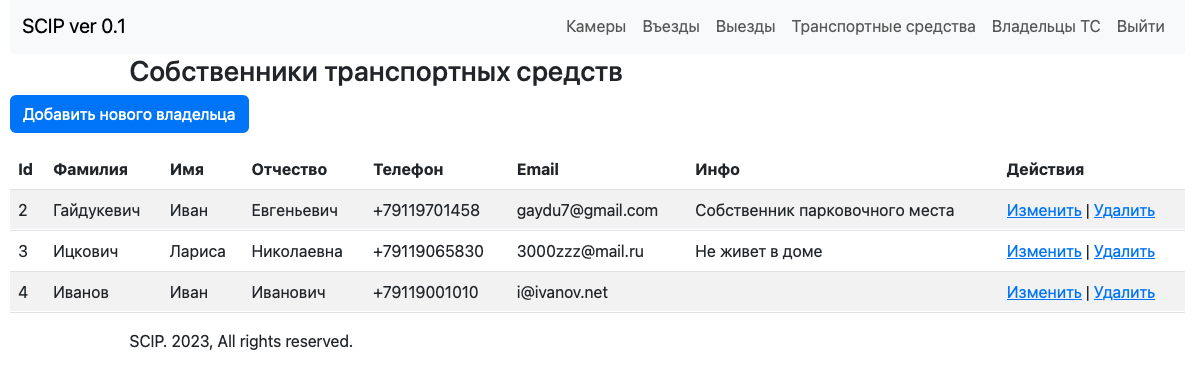


**Консьержу доступен функционал:**

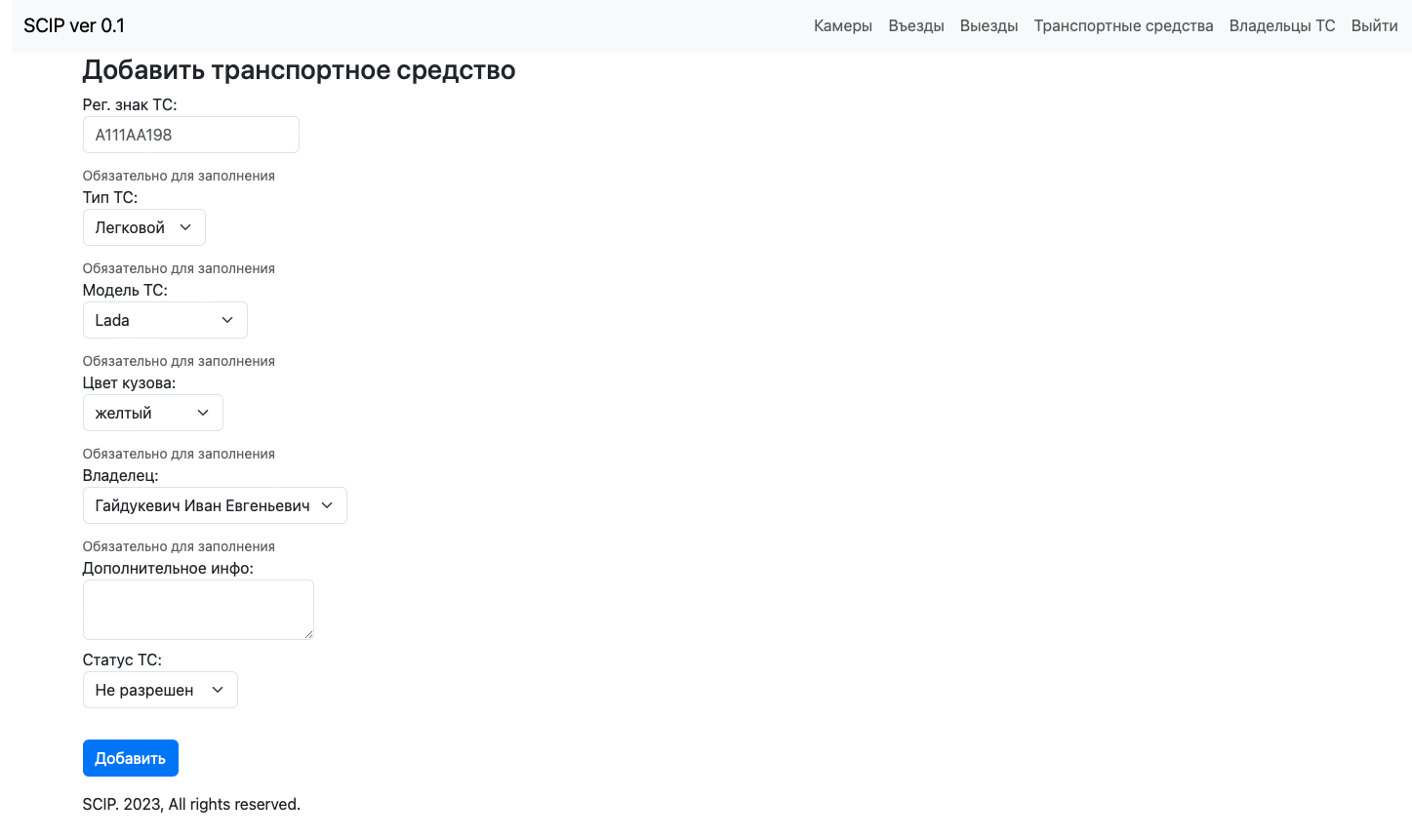
Просмотр всех зарегистрированных в паркинге транспортных средств

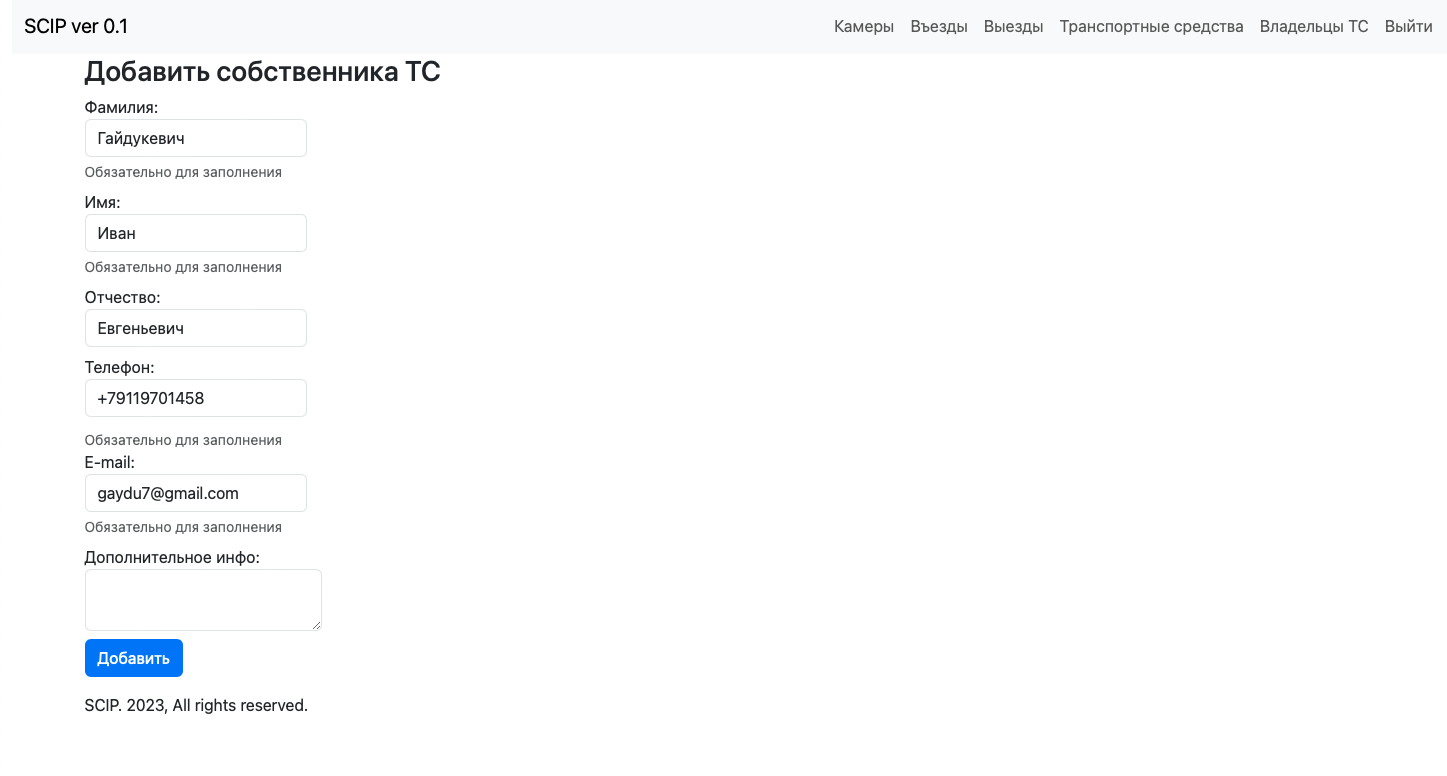


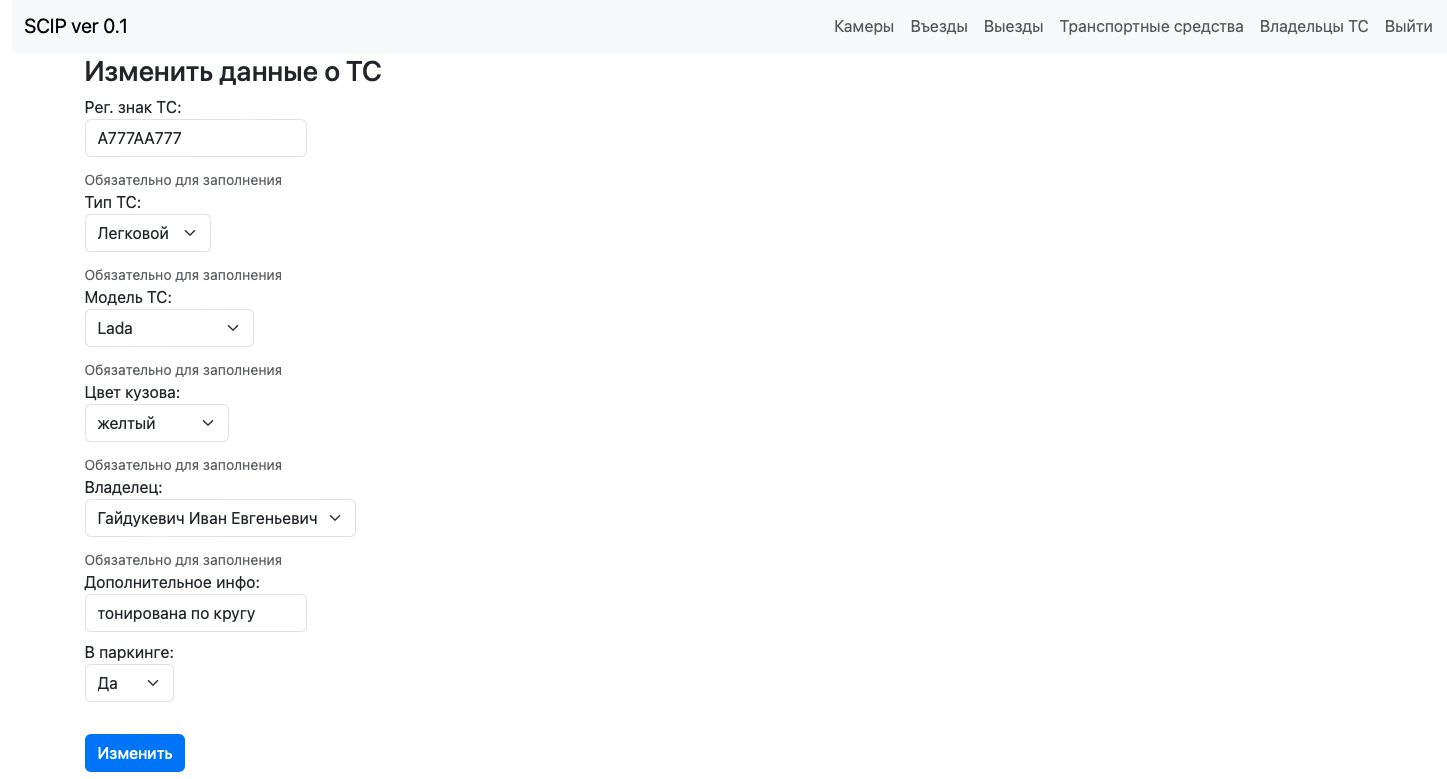
Просмотр всех владельцев транспортных средств



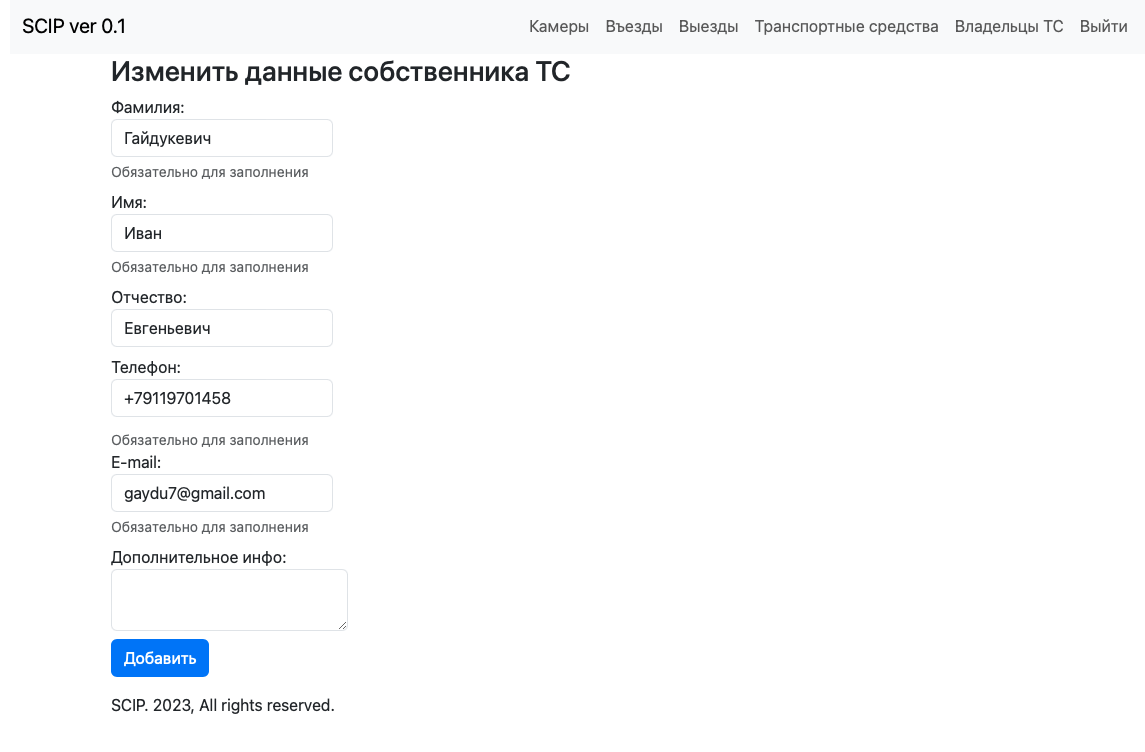
Регистрация (создание) новых транспортных средств



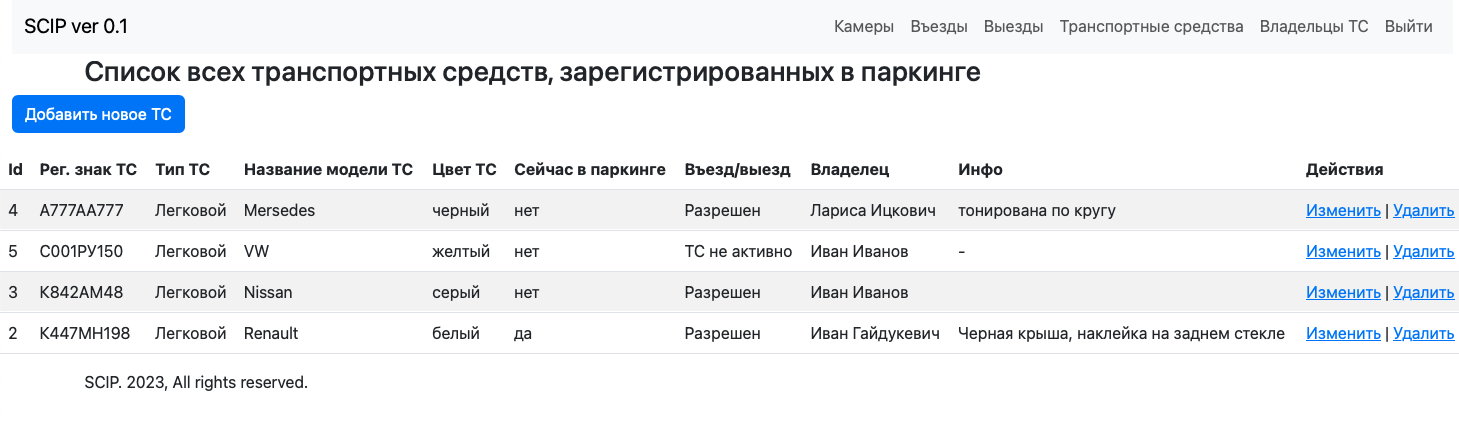
Регистрацию (создание) новых владельцев транспортных средств

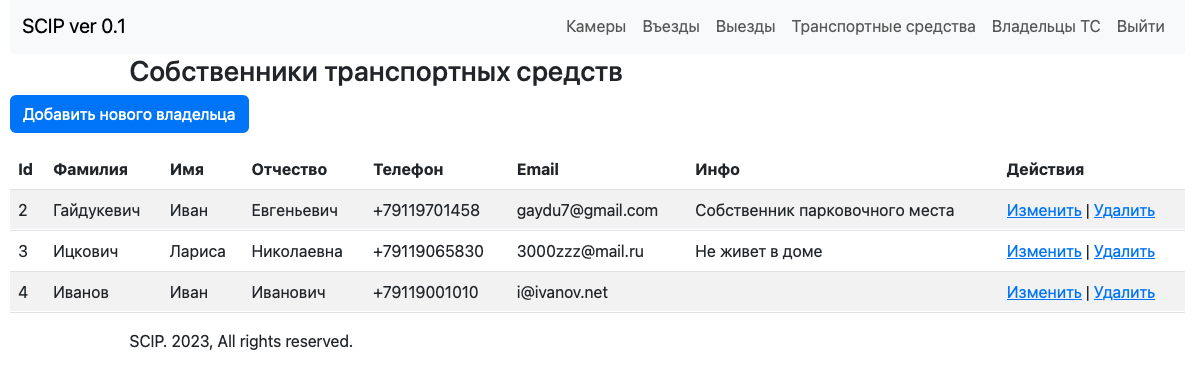
Изменение данных транспортных средств

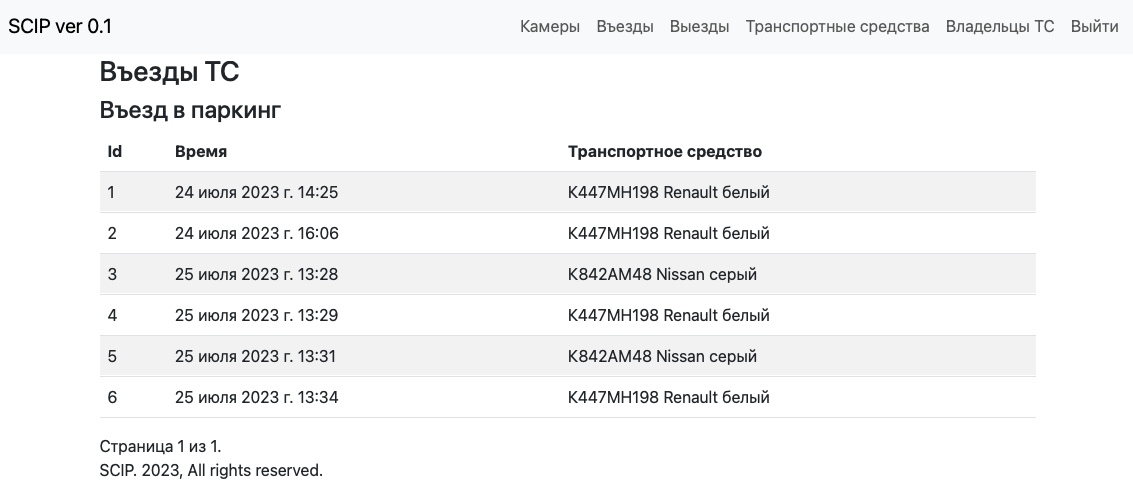
Изменение данных владельцев транспортных средств

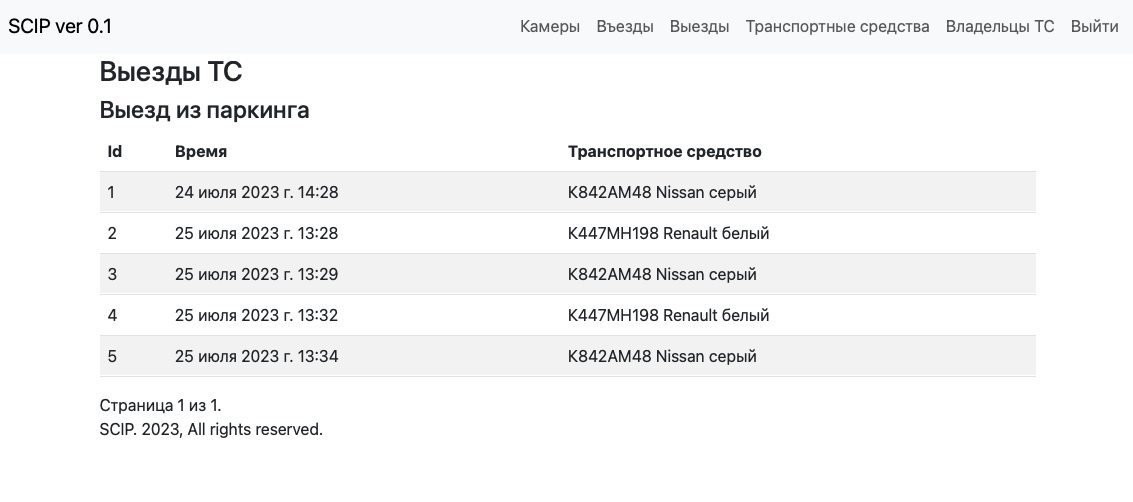


Удаление транспортных средств

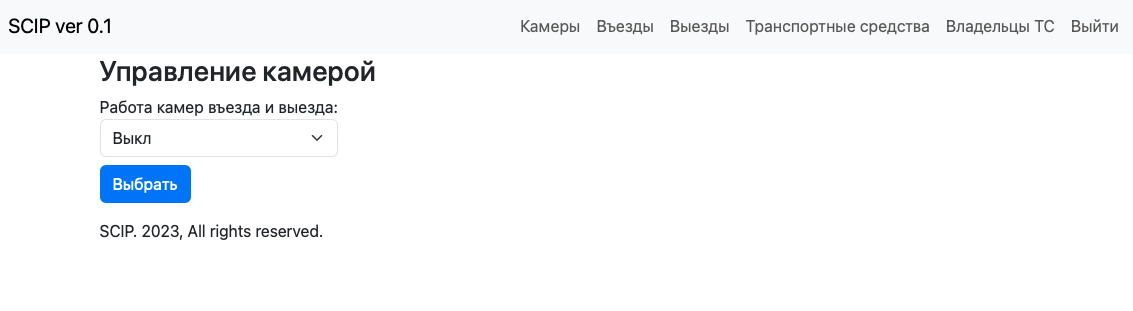


Удаление собственников транспортных средств

Просмотр журнала въездов в паркинг

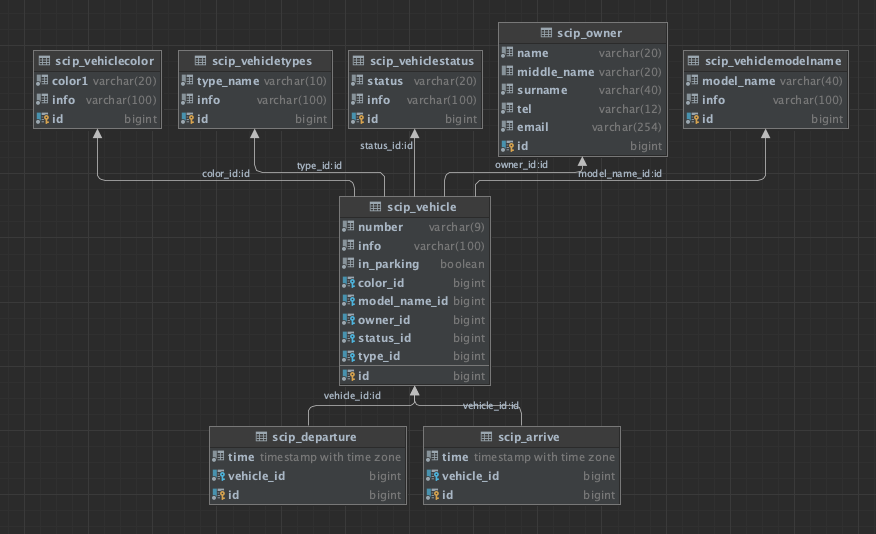
Просмотр журнала выезда из паркинга

Включение и выключение камер доступа



1. Водитель ТС – пользуется функционалом системы въезда и выезда в паркинг с помощью системы компьютерного зрения. Держит связь оговоренным способом с консьержем для внесения данных о ТС в систему.

# Структура данных



Предметная область приложения

**Сущность ТС (Vehicle)** – включает в себя данные о государственных регистрационных знаках, цвете кузова, типе транспортного средства, модели, разрешения на въезд/выезд, а также владельце транспортного средства. Я является ключевой сущностью, так как используется в работе системы распознавания государственных регистрационных знаков.

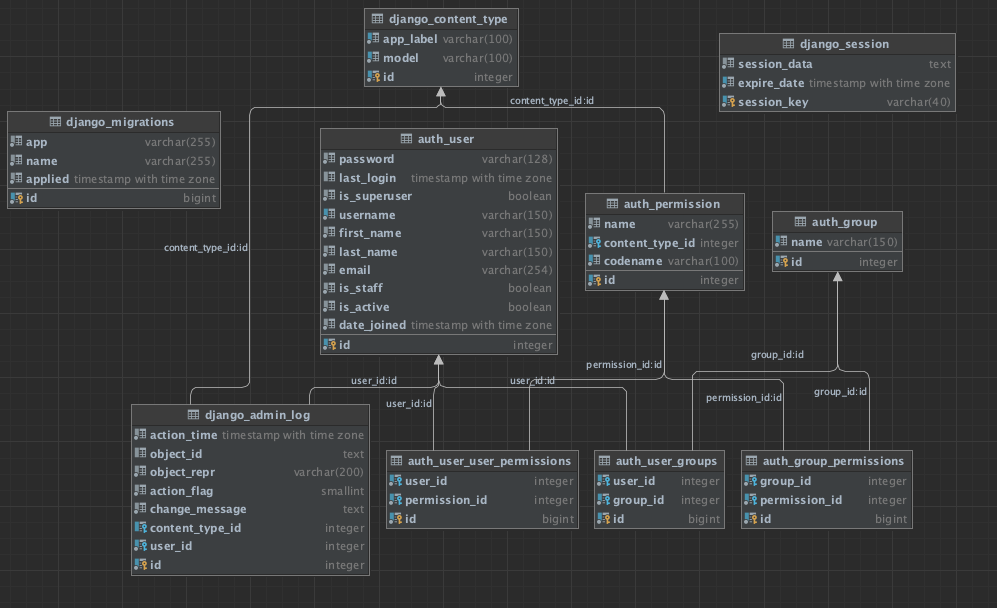
**Сущность Владелец ТС (Owner)** – включает в себя данные об имени, отчестве, фамилии, телефоне, email владельца ТС. Необходима хранения информации, которая может быть использована для связи с владельцами ТС.

**Сущности Въезд и Выезд (Arrive, Departure)** – включают в себя данные о времени въезда и выезда из паркинга и ТС, которое это осуществил. Используются для журналирования въездов и выездов.

Остальные сущности **VehicleColor, VehicleModelName, VehicleType, VehicleStatus** хранят воможные цвета кузовов ТС, названия моделей ТС, типы ТС, статусы ТС. VehicleStatus используется для установки разрешения на въезд и выезд транспортного средства. Эти сущности формируют справочные таблицы в базе данных для удобства расширения и масштабирования.

**Структура данных, используемая для аутентификации и**

**авторизации в приложении**



Предоставляется фреймворком Django из коробки.

**Функции системы**

1. **Распознавание государственных регистрационных знаков**
   1. **Определение места размещения таблички с государственным регистрационным знаком**

Осуществляется импортированной в проект библиотекой cv2 (OpenCV 3), а именно функцией detectMultiScale, которая применяется к объекту, который хранит модель предобученной нейронной сети от OpenCV. Данная модель загружается в виде внешнего файла plates.xml

* 1. **Распознавание текста на найденных государственных регистрационных знаках:**

Осуществляется импортированной в проект объектом Reader из библиотеки EasyOCR. Объект Reader определяется с указанием обязательного аргумента – списка языков распознавания. В нашем случае – 'ru' (русский).

К объекту применяется функция readtext(), в которую качестве аргумента передается строка желаемых к распознаванию символов, что особенно полезно в решении задачи распознавания теста с таблички государственного регистрационного знака.

На выходе данной функции мы имеем список со всеми распознанными строками.

* 1. **Валидация полученных строк с предполагаемыми автомобильными регистрационными знаками:**

С помощью регулярных выражений (модуль re) происходит отсев тех, строк, которые не попадают под шаблон. На выходе остаются только те строки, которые могут быть государственным регистрационным знаком согласно ГОСТу. Всего предусмотрено 14 шаблонов для определения абсолютно всех возможных форматов государственных регистрационных знаков (в том числе на мотоциклы и прицепы).

1. **Доступ транспортного средства в паркинг / из паркинга**
   1. **Доступ в паркинг/ из паркинга**

Для доступа в паркинг/из паркинга каждый распознанный валидный регистрационный знак сравнивается со списком регистрационных знаков, записанных в БД. Если происходит совпадение – система отправляет сигнал на открытие ворот. Но стоит отметить, что функция также проверяет наличие или отсутствие запрета на въезд/выезд транспортного средства, даже если его регистрационный знак внесен в БД. Это является дополнительным фактором безопасности паркинга.

* 1. **Журналирование въездов и выездов в паркинг**

Создание записей в базе данных о въездах и выездах (сущности Arrive, Departure) с временем въезда/выезда и привязкой ТС.

1. **Администрирование системы**

3.1 CRUD ТС: чтение, создание, изменение, удаление записи в базе данных (сущность Vehicle).

3.2 CRUD владельца ТС: чтение, создание, изменение, удаление записи в базе данных (сущность Owner).

3.3 Функция управления камерами. Консьерж через веб-форму получает возможность включать и выключать камеры для определения государственных регистрационных знаков. В момент включения камер для работы функционала каждой из них создается отдельный поток. В момент выключения цикл работы камер прекращается и поток завершается.

3.4. Функция запрета на въезд и выезд. Существует возможность через веб-форму запретить въезд и выезд транспортного средства, зарегистрированного в паркинге. Транспортному средству устанавливается статус Запрещен или Разрешен (сущность VehicleStatus).

**4. Аутентификация пользователя:**

4.1 Логин. Приложение полностью защищено для доступа через веб. Для аутентификации необходимо ввести логин и пароль консьержа или администратора в специальную веб-форму.

4.2 Логаут. Пользователь может выйти из системы через применение встроенного в Django механизма логаута. Для этого реализована соответствующая веб-кнопка.

**Практическая значимость работы**

Данную базовую версию приложения можно использовать в жилых комплексах для бесключевого доступа в помещения паркингов. Ее применение избавит от контроля доступа в паркинг, постоянно находящимся в нем, стороже. Это исключает человеческий фактор, а также экономит денежные средства владельцам мест в паркинге.

# Заключение

Нами в полной мере были реализованы поставленные цели работы: изучены python-библиотеки, реализующие технологии компьютерного зрения, а также создана базовая версия приложения для контроля доступа въезда в автомобильный паркинг, которая дает возможность бесключевого доступа в помещение паркинга, используя данные с государственных регистрационных знаков. Обе части приложения были протестированы ручным способом, ввиду ограниченного времени не удалось создать юнит-тесты.

Основные трудности, с которыми пришлось столкнуться, касались поиска подходящих библиотек и стыковки их между собой. Проект изначально планировалось реализовать на библиотеке Tesseract, но ввиду конфликтов версий с библиотекой OpenCV, было принято решение использовать библиотеку EasyOCR. Данная библиотека не без проблем заработала с OpenCV. Потребовались незначительные ручные доработки. После данных манипуляций мы получили рабочий стек технологий, который использовали в своем проекте.

В процессе работы были выявлены стратегии по повышению защищенности доступа в автомобильный паркинг, основанного на компьютерного зрения, чтобы исключить возможность подмены автомобильных номеров.

На базе сделанной работы планируется реализовать:

1. Двухфакторную аутентификацию транспортных средств (например, по государственному регистрационному знаку и цвету кузова ТС)
2. Систему личных кабинетов для владельцев транспортных средств с возможностью просмотра журнала выездов/выездов их ТС, а также с функцией запрещать/разрешать выезд/въезд ТС через личный кабинет.
3. Расширение аналитического функционала консьержа (например, поиск по времени въездов, выездов).
4. Создание видеосервера, который сохраняет в автоматическом режиме короткие видеоролики въезжающий и выезжающих ТС.
5. Создать юнит-тесты для автоматизированного тестирования основных модулей приложения

# Литература

1. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любо­ пытствующих. - СПб.: Питер, 2017. - 288 с.:
2. Лусиану Р. Python – к вершинам мастерства: Лаконичное и эффективное программирование / пер. с англ. А. А. Слинкина. 2-е изд. – М.: МК Пресс, 2022. – 898 с.
3. Меле А. Django 4 в примерах / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2023. – 800 с.
4. Постолит А. В. Python, Django и PyCharm для начинающих. -СПб.: БХВ-Петербурr, 2021. - 464 с.
5. Свейгарт Э. Python. Чистый код для продолжающих. — СПб.: Питер, 2022. — 384 с.
6. *Фридл Д.* Регулярные выражения – Пер. с англ. – СПб.: Символ-Плюс, 2013. – 608 с.
7. Документация по OpenCV <https://docs.opencv.org/4.x/index.html>
8. Документация по EasyOCR <https://www.jaided.ai/easyocr/documentation/>
9. Документация по Django <https://docs.djangoproject.com/en/4.2/>
10. Документация по Python 3.10 <https://docs.python.org/3.10/>

**Приложение 1.**

Скриншоты процесса определения табличек с автомобильными номерами





