

УТВЕРЖДЕН

RU.1107746574308.068-01 99 01-ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«Автоматизированная платформа
«Платформа агрегации сервисов» (SmartWaste)»

Инструкция по установке и проверке функционирования

RU.1107746574308.068-01 99 01

Листов 57

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

2023

АННОТАЦИЯ

Документ содержит сведения о порядке скачивания, установки, настройки и проверки функционирования программного обеспечения «Автоматизированная платформа «Платформа агрегации сервисов» (SmartWaste)» RU.1107746574308.068-01 (далее – Система).

Документ содержит пошаговую инструкцию по развертыванию экземпляра Системы и предназначен для использования экспертом при развертывании проверочного экземпляра Системы на тестовом стенде.

СОДЕРЖАНИЕ

1. Общие сведения	4
1.1. Наименование программы.....	4
1.2. Назначение программы.....	4
1.3. Область применения	5
1.4. Особенности распространения и применения.....	5
2. Требования к техническим средствам	6
2.1. Требования к техническим средствам при установке на стенде	6
2.2. Требования к техническим средствам при установке на мобильном устройстве	6
3. Требования к общему программному обеспечению	7
3.1. Требования к программному обеспечению при установке на стенде.....	7
3.2. Требования к программному обеспечению при установке на мобильном устройстве.....	7
4. Уровень подготовки пользователей	8
4.1. Требования к квалификации персонала при проверке Системы.....	8
4.2. Требования к квалификации персонала при штатной установке и эксплуатации	8
5. Сборка и развертывание	9
5.1. Подготовка виртуальной машины для сборки Системы.....	9
5.2. Сборка и развертывание Системы для работы в веб-приложении.....	9
5.2.1. Сборка.....	9
5.2.1.1. Backend (Django приложение).....	9
5.2.1.2. Frontend (Angular приложение).....	10
5.2.1.3. Сервисы телеметрии (Python приложение).....	10
5.2.1.4. OLTP (Python приложение)	11
5.2.1.5. Websocket (Python приложение)	11
5.2.2. Развертывание.....	12
5.2.2.1. Naproxy-configurator	12
5.2.2.2. Сервисы телеметрии	13
5.2.2.3. PostgreSQL	26
5.2.2.4. Основные сервисы Системы (Django приложение, Frontend, Traefic и Redis).....	26
5.3. Сборка и развертывание Системы для работы в мобильном приложении	38
5.3.1. Подготовка к сборке .apk файла.....	38
5.3.2. Создание .apk файла.....	38
6. Запуск	39
6.1. Запуск Системы для работы в веб-приложении.....	39
6.2. Запуск Системы для работы в мобильном приложении	39
7. Проверка функционирования	40
7.1. Проверка функционирования Системы при работе в веб-приложении	40
7.2. Проверка функционирования Системы при работе в мобильном приложении	54

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Наименование программы

Полное наименование программы: Автоматизированная платформа «Платформа агрегации сервисов» (SmartWaste) RU.1107746574308.068-01.

Сокращенное наименование программы: АП «Платформа агрегации сервисов». В рамках настоящего документа употребляется также термин «Система».

Обозначение программы: RU.1107746574308.068-01.

Автоматизированная платформа «Платформа агрегации сервисов» (SmartWaste) RU.1107746574308.068-01 – это российское программное обеспечение, организация-разработчик: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

Сайт организации-разработчика: <https://big3.ru>.

Организация-правообладатель: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

1.2. Назначение программы

1.2.1. Функциональное назначение

Система предназначена для обеспечения взаимодействия между участниками рынка обращения с отходами производства и потребления III–V классов опасности за исключением твердых коммунальных отходов: образователями отходов, перевозчиками и получателями отходов (переработчиками).

1.2.2. Эксплуатационное назначение

Система предназначена для обеспечения эффективного взаимодействия организаций-пользователей между собой при проведении следующих работ:

- поиск исполнителей, оказывающих услуги по обращению с отходами;
- заключение договоров об оказании услуг;
- оптимизация процессов планирования и исполнения заказов перевозчиками;
- учет и контроль выполнения работ по обращению с отходами.

1.3. Область применения

Информационное обеспечение автоматизированных процессов поиска исполнителей, заключения и исполнения договоров об оказании услуг по обращению с отходами производства и потребления III–V классов опасности (за исключением твердых коммунальных отходов) в соответствии с требованиями действующего законодательства Российской Федерации.

Особенности распространения и применения

Система реализована в виде информационной системы с трехуровневой клиент-серверной архитектурой и представляет собой совокупность следующих компонентов:

- серверная часть:
 - сервер баз данных,
 - сервер приложений;
- веб-приложение для пользователей с ролями «Администратор компании», «Отходообразователь», «Перевозчик» и «Переработчик»;
- мобильное приложение для пользователей с ролями «Водитель» и «Приемщик».

Серверная часть развернута на стороне центра обработки данных ООО «Селектел» и функционирует под управлением операционной системы (далее – ОС) Debian GNU/Linux (версии 11 и выше).

В качестве системы управления базами данных (СУБД) применяется СУБД PostgreSQL 14.7.

Веб-приложение обеспечивает взаимодействие с пользователем по технологии «тонкий клиент» с применением распространенных веб-браузеров (Mozilla Firefox, Microsoft Edge, Google Chrome, Safari, «Яндекс.Браузер»).

Мобильное приложение обеспечивает взаимодействие с пользователем посредством мобильных устройств (мобильные телефоны, планшетные устройства) под управлением ОС Android (версии 6.0 и выше).

Система функционирует под управлением ОС Windows, Linux в многопользовательском режиме.

2. ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ

2.1. Требования к техническим средствам при установке на стенде

Для проверки функционирования Системы при работе в веб-приложении на стенде применяется ПЭВМ с характеристиками не хуже:

- центральный процессор (CPU): Intel Core i3, 2 ГГц (или эквивалент);
- объем оперативной памяти (RAM): 4 ГБ;
- жесткий диск (HDD): 1x HDD – объем свободного пространства 10 ГБ;
- видеоадаптер: встроен в системную плату;
- сетевая плата: Ethernet 10 Мбит/с;
- дополнительное оборудование: монитор SVGA 1024x768, мышь, клавиатура

ПЭВМ должна быть подключена к сети Интернет.

2.2. Требования к техническим средствам при установке на мобильном устройстве

Для проверки функционирования мобильного приложения Системы применяется мобильное устройство с характеристиками не хуже:

- объем оперативной памяти: не менее 4 ГБ;
- объем встроенной доступной пользователю памяти: не менее 1 ГБ;
- поддержка стандартов связи: GSM 850/900/1800/1900;
- интерфейсы: Wi-Fi;
- геолокация: GPS и/или Глонасс;
- наличие камеры со вспышкой: да;
- дополнительное ПО: Telegram, приложение для отправки электронных писем (для связи со службой поддержки).

Для работы мобильного приложения на мобильном устройстве должно быть не менее 100 Мб свободного пространства постоянной памяти.

3. ТРЕБОВАНИЯ К ОБЩЕМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

3.1. Требования к программному обеспечению при установке на стенде

Для установки и проверки функционирования Системы на ПЭВМ должно быть установлено следующее общее программное обеспечение:

- операционная система (далее – ОС): Debian GNU/Linux (версии 11 и выше);
- программное средство Docker (<https://www.docker.com>), включая инструментальное средство Docker Swarm;
- веб-браузер Google Chrome актуальной версии.

3.2. Требования к программному обеспечению при установке на мобильном устройстве

Для установки и проверки функционирования мобильного приложения Системы на мобильном устройстве должна быть установлена ОС Android (версии 6.0 и выше).

4. УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ

4.1. Требования к квалификации персонала при проверке Системы

Для выполнения установки и проверки функционирования Системы пользователь должен обладать следующими знаниями и навыками:

- навыки администрирования Linux-подобных систем;
- уверенное знание системы управления конфигурациями Ansible;
- навыки использования программного средства Docker и Docker Swarm;
- навыки уверенного пользователя мобильного устройства под управлением ОС Android (версии 6.0 и выше);
- навыки использования веб-браузера типа Google Chrome;
- знание общих принципов функционирования веб- и мобильных приложений;
- общих принципов функционирования компьютерных сетей.

4.2. Требования к квалификации персонала при штатной установке и эксплуатации

Установка Системы в процессе основного жизненного цикла выполняется организацией-разработчиком Системы. Конечными пользователями установка не выполняется.

Эксплуатация выполняется конечными пользователями устройств, которые должны обладать следующими знаниями и навыками:

- знаниями соответствующей предметной области;
- знаниями функциональности Системы согласно эксплуатационной документации;
- навыками работы на персональном компьютере под управлением ОС Microsoft Windows (версии 7 и выше) и Linux;
- навыками работы с мобильным устройством под управлением ОС Android (версии 6.0 и выше);
- навыками использования распространенных веб-браузеров, в том числе и мобильных, актуальных общедоступных версий.

5. СБОРКА И РАЗВЕРТЫВАНИЕ

Сборка Системы предполагает уже готовый настроенный сварм-кластер. Инициализация и настройка сварм-кластера для Системы не имеет особых ограничений и может быть вариативной. Реализация данного этапа полностью на стороне программиста, который занимается разверткой.

5.1. Подготовка виртуальной машины для сборки Системы

На виртуальной машине, на которой будет осуществляться сборка необходимы следующие программы:

- docker (<https://docs.docker.com/engine/install/>);
- git (<https://git-scm.com/downloads>);
- базовый образ Astra Linux 1.7.2 (имя образа: registry.big3.ru/devops/astra/astra-python-contaner:default).

Образ Astra Linux 1.7.2 необходимо добавить в локальный реестр (Docker registry), а затем удалённый: export REGISTRY_HOST=адрес докер регистри.

Далее эта перенесенная на протяжении всех стадий сборки должна быть в переменных окружения.

- docker load -i astra-python-contaner.tar.gz
- docker tag registry.big3.ru/devops/astra/astra-python-contaner:default \$ {REGISTRY_HOST}/devops/astra/astra-python-contaner:default

Примечание. Образ идёт в комплекте с архивами исходных кодов.

5.2. Сборка и развертывание Системы для работы в веб-приложении

5.2.1. Сборка

Разархивировать дистрибутив: tar -xzvf distribution-kit.tar.gz && cd distribution-kit

5.2.1.1. Backend (Django приложение)

В корневой директории исходных кодов запустить сборку базового докер образа:

```
1. cd sw-back && docker build -t sw-app-deps:latest -f requirements.dockerfile --build-arg  
${REGISTRY_HOST}/devops/astra/astra-python-contaner:default
```

ВАЖНО! Образ sw-app-deps:latest собирается из базового образа registry.big3.ru/devops/astra/astra-python-contaner:default, соответственно необходимо через --build-arg передать правильное имя образа.

Собираем образ с базисом.

```
2. docker build -t sw-app-bazis:latest -f bazis.dockerfile --build-arg DEPS_IMAGE=sw-app-deps:latest
```

Собираем конечный образ.

```
3. docker build -t ${REGISTRY_HOST}/sw-app:latest -f Dockerfile --build-arg DEPS_IMAGE=sw-app-bazis:latest
```

5.2.1.2. Frontend (Angular приложение)

В корневой директории исходных кодов Frontend запустить сборку базового образа:

```
1. cd ../workspace-wa && docker build -f node_modules.dockerfile -t sw-front-deps:latest
```

```
2. docker build -f Dockerfile -t sw-front:latest --build-arg DEPS_IMAGE=sw-front-deps:latest --build-arg SCRIPT=prod
```

5.2.1.3. Сервисы телеметрии (Python приложение)

В корневой директории исходных кодов телеметрии запустить сборку базового образа:

Определить переменные для сборки образа:

`BASE_IMAGE` – полное имя базового образа python:3.8-bullseye.

`DOCKER_IMAGE` – полное имя конечного образа

Пример:

```
export BASE_IMAGE=python:3.8-bullseye
```

```
export DOCKER_DEPS_IMAGE=telemetry-deps:latest
```

```
export DOCKER_IMAGE=${REGISTRY_HOST}/telemetry:latest
```

Собрать образ с зависимостями:

```
docker build -f requirements.dockerfile \  
-t ${DOCKER_DEPS_IMAGE} \  
--build-arg BASE_IMAGE=${BASE_IMAGE} .
```

Собрать конечный образ

```
docker build -f swarm.dockerfile \  
-t ${DOCKER_IMAGE} \  
--build-arg DEPS_IMAGE=${DOCKER_DEPS_IMAGE} .
```

```
--build-arg DEPS_IMAGE=${DOCKER_DEPS_IMAGE} \  
docker push $ DOCKER_IMAGE
```

5.2.1.4. OLTP (Python приложение)

В корневой директории исходных кодов OLTP запустить сборку базового образа:

Определить переменные для сборки образа в определённом закрытом контуре.

BASE_IMAGE – полное имя базового образа python:3.7-bullseye.

DOCKER_IMAGE – полное имя конечного образа

Пример:

```
export BASE_IMAGE=python:3.7-bullseye  
export DOCKER_DEPS_IMAGE=oltp-deps:latest  
export DOCKER_IMAGE=${REGISTRY_HOST}/oltp:latest
```

Собрать образ с зависимостями:

```
docker build -f ci/requirements.dockerfile \  
    -t ${DOCKER_DEPS_IMAGE} \  
    --build-arg BASE_IMAGE=${BASE_IMAGE} .
```

Собрать конечный образ

```
docker build -f ci/Dockerfile \  
    -t ${DOCKER_IMAGE} \  
    --build-arg DEPS_IMAGE=${DOCKER_DEPS_IMAGE} .
```

```
docker push $ DOCKER_IMAGE
```

5.2.1.5. Websocket (Python приложение)

В корневой директории исходных кодов Websocket запустить сборку базового образа:

Определить переменные для сборки образа в определённом закрытом контуре.

BASE_IMAGE – полное имя базового образа python:3 выкачанного из докерхаба в приватном регистре

DOCKER_DEPS_IMAGE – полное имя образа с зависимостями

DOCKER_IMAGE – полное имя конечного образа

Пример:

```
export BASE_IMAGE=python:3-bullseye  
export DOCKER_DEPS_IMAGE=websocket-deps:latest  
export DOCKER_IMAGE=${REGISTRY_HOST}/websocket:latest
```

Собрать образ с зависимостями:

```
docker build -f ./ci/requirements.dockerfile \  
-t ${DOCKER_DEPS_IMAGE} \  
--build-arg BASE_IMAGE=${BASE_IMAGE} .
```

Собрать конечный образ

```
docker build -f ./ci/Dockerfile \  
-t ${DOCKER_IMAGE} \  
--build-arg DEPS_IMAGE=${DOCKER_DEPS_IMAGE} \  
docker push $ DOCKER_IMAGE
```

5.2.2. Развертывание

5.2.2.1. Noproxy-configurator

1. Скопировать исходные коды хапрокси конфигуратора на целевую ВМ в директорию /opt/app/haproxy-configurator из дистрибутива

```
mkdir -p /opt/app && cp -r haproxy /opt/app/haproxy-configurator
```

2. Создать виртуальное окружение: cd /opt/app/haproxy-configurator && python3.9 -m venv venv

ВАЖНО! В Системе обязательно должен стоять пайтон не ниже версии 3.9!

3. Активировать виртуальное окружение и все дальнейшие действия должны производиться из него: source venv/bin/activate

4. pip install -r requirements.txt

5. Создать файл с переменной окружения: echo

```
"AUTH_TOKEN=Ikml6ZgA1GZGmkxLWwhAwPaLs2u7pE" > .env
```

Значение может быть любым.

6. Создать систему юнит:

```
cat << EOF > /lib/systemd/system/haproxy-configurator.service
```

```
[Unit]
```

```
Description=haproxy-configurator backend
```

```
[Service]
```

```
EnvironmentFile=/opt/app/haproxy-configurator/.env
```

```
User=root
```

```
ExecStart=/opt/app/haproxy-configurator/venv/bin/python /opt/app/haproxy-configurator/app.py
Restart=always
TimeoutStartSec=10
RestartSec=10
WorkingDirectory=/opt/app/haproxy-configurator
```

[Install]

```
WantedBy=multi-user.target
EOF
```

7. `systemctl daemon-reload`
8. Скопировать папку `errors` в директорию хапрокси: `cp -r errors /etc/haproxy`
9. Установить хапрокси: `apt update && apt install haproxy -y`
10. Запустить хапрокси конфигурактор: `systemctl start haproxy-configurator.service`
11. Проверить успешность запуска службы: `systemctl status haproxy-configurator.service`

5.2.2.2. Сервисы телеметрии

Сервисы телеметрии взаимодействуют с хапрокси и хапрокси конфигурактором, поэтому должны быть запущены на той же ВМ, где и хапрокси.

1. Инициализировать сварм: `docker swarm init`
2. Скопировать файл `init.sql` из дистрибутива на целевую ВМ где будет запущен кликхаус в директорию `/opt/clickhouse/init.sql`: `mkdir -p /opt/clickhouse && cp -r init.sql /opt/clickhouse/init.sql`
3. Создать компоуз файл для кликхауса:

```
cat << EOF > clickhouse.yml
version: '3.8'
```

```
x-common: &common
deploy: &common_deploy
  mode: replicated
  replicas: 1
  placement:
    constraints:
      - "node.role == manager"
```

```
restart_policy:  
  condition: any  
  delay: 5s  
  max_attempts: 10  
  window: 120s  
update_config:  
  parallelism: 1  
  delay: 10s  
  order: start-first
```

services:

clickhouse:

<<: *common

image: yandex/clickhouse-server:21.7.7.47

ports:

- "8123:8123"
- "9000:9000"

environment:

- CLICKHOUSE_USER=\\${CLICKHOUSE_USER}
- CLICKHOUSE_PASSWORD=\\${CLICKHOUSE_PASSWORD}
- CLICKHOUSE_DB=\\${CLICKHOUSE_DB}

volumes:

- /opt/clickhouse/init.sql:/docker-entrypoint-initdb.d/init.sql
- ch:/var/lib/clickhouse

networks:

default:

aliases:

- \\${CLICKHOUSE_HOST}

networks:

default:

external: true

name: \\${TELEMTRY_NET}

volumes:

ch:

EOF

4. Определить переменные окружения для запуска стэка с кликхаусом:

```
export CLICKHOUSE_USER=clk_sw
export CLICKHOUSE_PASSWORD=strong_clickhouse_password
export CLICKHOUSE_DB=oc
export CLICKHOUSE_HOST=clickhouse
export TELEMETRY_NET=telemetry
```

5. Создать оверлейную сеть для общения между сервисами телеметрии и кликхаусом:

```
docker network create ${TELEMETRY_NET} --driver overlay --scope swarm
```

6. Создать стэк с кликхаусом:

```
docker stack deploy --compose-file clickhouse.yml clickhouse
```

Проверить успешность запуска командой

```
docker stack ps clickhouse
```

7. Создать компоуз файл для RabbitMQ с конфигами:

Придумать свои пароли.

```
mkdir -p /opt/rabbit && cd /opt/rabbit
```

```
cat << EOF > definitions.json
```

```
{
  "users": [
    {
      "name": "admin",
      "password": "super_strong_rabbit_password",
      "tags": "administrator"
    }
  ],
  "vhosts": [
    {
      "name": "oc"
    },
    {
      "name": "bus"
    }
  ]
}
```

```
}  
],  
"permissions": [  
  {  
    "user": "admin",  
    "vhost": "oc",  
    "configure": ".*",  
    "write": ".*",  
    "read": ".*"  
  },  
  {  
    "user": "admin",  
    "vhost": "bus",  
    "configure": ".*",  
    "write": ".*",  
    "read": ".*"  
  }  
],  
"topic_permissions": [  
  
],  
"parameters": [  
  
],  
"global_parameters": [  
],  
"policies": [  
],  
"queues": [  
],  
"exchanges": [  
],  
"bindings": [  
]
```



```
}  
EOF  
cat << EOF > rabbitmq.conf  
## Clustering  
cluster_partition_handling = ignore  
  
## Defaults  
# During the first start, RabbitMQ will create a vhost and a user  
# These config items control what gets created  
default_permissions.configure = .*  
default_permissions.read = .*  
default_permissions.write = .*  
default_vhost = oc  
default_user = admin  
default_pass = super_strong_rabbit_password  
  
## Networking  
listeners.tcp.default = 5672  
  
## Management  
management.tcp.ip = 0.0.0.0  
management.tcp.port = 15672  
  
## Resource limits  
# Set an absolute free disk space limit  
disk_free_limit.absolute = 2g  
  
# 2 hours in milliseconds  
consumer_timeout = 7200000  
  
management.load_definitions = /etc/rabbitmq/definitions.json  
EOF
```

```
cat << EOF > rabbit.yml
```

```
version: '3.8'
```

```
x-common: &common
```

```
networks:
```

```
- default
```

```
deploy: &common_deploy
```

```
mode: replicated
```

```
replicas: 1
```

```
placement:
```

```
constraints:
```

```
- "node.role == manager"
```

```
restart_policy:
```

```
condition: any
```

```
delay: 5s
```

```
max_attempts: 10
```

```
window: 120s
```

```
update_config:
```

```
parallelism: 1
```

```
delay: 10s
```

```
order: stop-first
```

```
services:
```

```
rabbit:
```

```
<<: *common
```

```
image: rabbitmq:management-alpine
```

```
volumes:
```

```
- /opt/rabbit/definitions.json:/etc/rabbitmq/definitions.json
```

```
- /opt/rabbit/rabbitmq.conf:/etc/rabbitmq/rabbitmq.conf
```

```
networks:
```

```
default:
```

```
aliases:
```

```
- rabbit
```

networks:

default:

external: true

name: \\${TELEMTRY_NET}

EOF

8. Определить переменные окружения для запуска стэка с рэббитом:

```
export TELEMTRY_NET=telemetry
```

9. Запустить стэк с рэббитом:

```
docker stack deploy --compose-file rabbit.yml rabbit
```

10. Создать компоуз файл с сервисами телеметрии:

```
cat << EOF > telemetry-stack.yml
```

```
version: '3.8'
```

```
x-common: &common
```

```
image: \${REGISTRY_HOST}/telemetry:latest
```

```
networks:
```

```
default:
```

```
aliases:
```

```
- \${TELEMTRY_NET}
```

```
deploy: &common_deploy
```

```
mode: replicated
```

```
replicas: 1
```

```
placement:
```

```
constraints:
```

```
- "node.role == manager"
```

```
restart_policy:
```

```
condition: any
```

```
delay: 5s
```

```
max_attempts: 10
```

```
window: 120s
```

```
update_config:
```

```
parallelism: 1
```

```
delay: 10s
```

order: stop-first

services:

wialon_1:

<<: *common

ports:

- "8886:8886"

environment:

- BIND_ADDR=0.0.0.0:8886
- DEBUG=1
- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}
- MQ_CONNECT=\\${MQ_CONNECT}
- LOGS_DIR=/var/log/telemetry
- PROXY_PROTO=1
- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- wialon_1:/var/log/telemetry

command: telemetry-wialon

wialon_2:

<<: *common

ports:

- "8887:8887"

environment:

- BIND_ADDR=0.0.0.0:8887
- DEBUG=1
- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}
- MQ_CONNECT=\\${MQ_CONNECT}
- LOGS_DIR=/var/log/telemetry
- PROXY_PROTO=1
- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- wialon_2:/var/log/telemetry

command: telemetry-wialon

wialon_retranslator_1:

<<: *common

ports:

- "8600:8600"

environment:

- BIND_ADDR=0.0.0.0:8600

- DEBUG=1

- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}

- MQ_CONNECT=\\${MQ_CONNECT}

- LOGS_DIR=/var/log/telemetry

- PROXY_PROTO=1

- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- wialon_retranslator_1:/var/log/telemetry

- telemetry_raw:/var/log/telemetry/raw

command: telemetry-wialon-retranslator

egts_1:

<<: *common

ports:

- "8700:8700"

environment:

- BIND_ADDR=0.0.0.0:8700

- DEBUG=1

- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}

- MQ_CONNECT=\\${MQ_CONNECT}

- LOGS_DIR=/var/log/telemetry

- PROXY_PROTO=1

- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- egts_1:/var/log/telemetry

command: telemetry-egts

avtograf_1:

<<: *common

ports:

- "8800:8800"

environment:

- BIND_ADDR=0.0.0.0:8800
- DEBUG=1
- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}
- MQ_CONNECT=\\${MQ_CONNECT}
- LOGS_DIR=/var/log/telemetry
- PROXY_PROTO=1
- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- avtograf_1:/var/log/telemetry

command: telemetry-avtograf

wialon_test_1:

<<: *common

ports:

- "8500:8500"

environment:

- BIND_ADDR=0.0.0.0:8500
- DEBUG=1
- TEST_MODE=1
- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}
- MQ_CONNECT=\\${MQ_CONNECT}
- LOGS_DIR=/var/log/telemetry
- PROXY_PROTO=1
- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- wialon_test_1:/var/log/telemetry

command: telemetry-wialon

wialon_retranslator_test_1:

<<: *common

ports:

- "8610:8610"

environment:

- BIND_ADDR=0.0.0.0:8610

- DEBUG=1

- TEST_MODE=1

- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}

- MQ_CONNECT=\\${MQ_CONNECT}

- LOGS_DIR=/var/log/telemetry

- PROXY_PROTO=1

- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- wialon_retranslator_test_1:/var/log/telemetry

- telemetry_raw:/var/log/telemetry/raw

command: telemetry-wialon-retranslator

egts_test_1:

<<: *common

ports:

- "8510:8510"

environment:

- BIND_ADDR=0.0.0.0:8510

- DEBUG=1

- TEST_MODE=1

- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}

- MQ_CONNECT=\\${MQ_CONNECT}

- LOGS_DIR=/var/log/telemetry

- PROXY_PROTO=1

- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- egts_test_1:/var/log/telemetry

command: telemetry-egts

avtograf_test_1:

<<: *common

ports:

- "8520:8520"

environment:

- BIND_ADDR=0.0.0.0:8520
- DEBUG=1
- TEST_MODE=1
- CLICKHOUSE_URI=\\${CLICKHOUSE_URI}
- MQ_CONNECT=\\${MQ_CONNECT}
- LOGS_DIR=/var/log/telemetry
- PROXY_PROTO=1
- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- avtograf_test_1:/var/log/telemetry

command: telemetry-avtograf

redis:

<<: *common

image: redis:5.0.3-alpine

command: redis-server --requirepass \\${REDIS_PASSWORD}

restart: always

server_logs:

<<: *common

ports:

- "8080:8080"

environment:

- LOGS_DIR=/var/log/telemetry
- FLASK_APP=run_log_reader.py
- REDIS_PASSWORD=\\${REDIS_PASSWORD}

volumes:

- telemetry:/var/log/telemetry

command: python -m flask run --host=0.0.0.0 --port=8080

volumes:

wialon_1:

wialon_2:

wialon_retranslator_1:

telemetry_raw:

egts_1:

avtograf_1:

wialon_test_1:

wialon_retranslator_test_1:

egts_test_1:

avtograf_test_1:

onboard_control:

telemetry:

networks:

default:

external: true

name: \\${TELEMETRY_NET}

EOF

11. Определить переменные окружения для запуска стэка с сервисами телеметрии:

```
export RABBIT_USERNAME=admin
```

```
export RABBIT_PASSWORD=super_strong_rabbit_password
```

```
export RABBIT_HOST=rabbit
```

```
export CLICKHOUSE_URI=ch://clk_sw:strong_clickhouse_password@clickhouse/oc
```

```
export
```

```
MQ_CONNECT=amqp://\${RABBIT_USERNAME}:\${RABBIT_PASSWORD}@\${RABBIT_HOST}/
```

```
bus
```

```
export REDIS_PASSWORD=strong_redis_password
```

```
export STREAMING_SERVER_IP=ип адрес вм на которой запущена телеметрия
```

```
export STREAMING_SERVER_DB=/var/data/ports.db
```

```
export TELEMETRY_NET=telemetry
```

12. Запустить стэк с сервисами телеметрии:

```
docker stack deploy --compose-file telemetry-stack.yml telemetry
```

Проверить успешность запуска командой `docker stack ps telemetry`

5.2.2.3. PostgreSQL

1. Установить PostgreSQL версии 14 по официальной инструкции на целевую ВМ для PostgreSQL:

<https://www.postgresql.org/download/>

2. Настроить PostgreSQL:

- зайти под пользователем postgres: `su - postgres`
- создать пользователя sw: `psql -U postgres -c "create role sw with login superuser;"`
- установить пароль: `psql -U postgres -c "\password sw;"`
- создать пустую БД sw: `psql -U postgres -c "create database sw owner sw;"`
- восстановить БД из предоставленного дампа: `pg_restore -U sw -d sw -Fr -j4 dump.backup`

3. В файле настроек PostgreSQL `/etc/postgresql/14/main/postgresql.conf` исправить конфигури:

- `listen_addresses = '*'`
- `max_connections = 1500`

4. В файле настроек подключений PostgreSQL `/etc/postgresql/14/main/pg_hba.conf` добавить в конец:

```
host all all 0.0.0.0/0 md5
```

5. Перезапустить постгрес: `systemctl restart postgresql@14-main.service`

6. Подключиться под созданным пользователем:

```
export PGPASSWORD=пароль который был установлен в шаге 2
```

```
psql -U sw -d sw -h 127.0.0.1
```

В случае успешного подключения, настройка PostgreSQL считается завершенной.

5.2.2.4. Основные сервисы Системы (Django приложение, Frontend, Traefic и Redis)

1. На отдельной ВМ для Django приложения и Frontend приложения инициализировать сварм:

```
docker swarm init
```

2. Создать оверлейные экстернал сети для Django и Traefic:

```
docker network create web --driver overlay --scope swarm
```

```
docker network create traefik --driver overlay --scope swarm
```

3. Создать файл настроен для Redis:

```
mkdir -p /opt/redis/
```

```
cat << EOF > /opt/redis/redis.conf
maxmemory 15360mb
maxmemory-policy allkeys-lru
port 6379
EOF
```

4. Создать стэкфайл с Redis:

```
cat << EOF > /opt/redis/redis.yml
version: '3.8'
```

```
x-common: &common
networks:
  - web
deploy: &common_deploy
mode: replicated
replicas: 1
placement:
  constraints:
    - "node.role == manager"
restart_policy:
  condition: any
  delay: 5s
  max_attempts: 10
  window: 120s
update_config:
  parallelism: 1
  delay: 10s
  order: stop-first

services:
  redis:
    <<: *common
    image: redis:7
    command: redis-server /usr/local/etc/redis/redis.conf
    volumes:
```

- /opt/redis/redis.conf:/usr/local/etc/redis/redis.conf

- redis:/data

networks:

web:

aliases:

- redis

volumes:

redis:

networks:

web:

external: true

name: \${SW_NET}

EOF

5. Определить переменные окружения для запуска стэка с Redis:

```
export SW_NET=web
```

6. Запустить стэк с Redis: `docker stack deploy --compose-file /opt/redis/redis.yml redis`

Успешность запуска командой проверить `docker stack ps redis`

7. Создать стэкфайл с Django приложением и Frontend:

```
cat << EOF > stack.yml
```

```
version: '3.8'
```

```
x-common: &common
```

```
env_file: \${_BS_APP_ENV_FILE}
```

```
networks:
```

```
- web
```

```
deploy: &common_deploy
```

```
mode: replicated
```

```
replicas: 1
```

```
placement:
```

```
constraints:
```

```
- "node.role == manager"
```

```
restart_policy:  
  condition: any  
  delay: 5s  
  max_attempts: 10  
  window: 120s  
update_config:  
  parallelism: 1  
  delay: 10s  
  order: stop-first
```

services:

front-nginx:

```
<<: *common
```

```
image: "sw-front:latest"
```

```
networks:
```

```
- traefik
```

```
volumes:
```

```
- sw_static:/usr/share/nginx/html/static:ro
```

```
deploy:
```

```
<<: *common_deploy
```

```
labels:
```

```
- "traefik.enable=true"
```

```
- "traefik.http.routers.lk-nginx-http.priority=1"
```

```
- "traefik.http.routers.lk-nginx-http.rule=Host(`$$SW_APP_PUBLIC_HOST`) &&
```

```
!PathPrefix(`/api`,`/admin`,`/allure`)"
```

```
- "traefik.http.routers.lk-nginx-http.service=nginx-http-service"
```

```
- "traefik.http.routers.admin-nginx-http.rule=Host(`$$SW_APP_ADMIN_HOST`) &&
```

```
!PathPrefix(`/api`,`/admin`,`/media`)"
```

```
- "traefik.http.routers.admin-nginx-http.service=nginx-http-service"
```

```
- "traefik.http.services.nginx-http-service.loadBalancer.server.port=80"
```

```
- "traefik.http.services.nginx-http-service.loadBalancer.passHostHeader=true"
```

```
- "traefik.http.routers.admin-nginx-http.middlewares=app-admin-auth"
```

```
- "traefik.http.middlewares.app-admin-  
auth.basicauth.users=admin:\$\$1\$\$FybM98Zl\$\$uqzrVf4mKiykiysR0v1mC1"
```

app-public:

```
<<: *common
```

```
image: "sw-back:latest"
```

```
command: bash -c "/opt/app/deploy/run/app.sh"
```

```
environment:
```

```
  BS_DEBUG: "False"
```

```
networks:
```

```
- traefik
```

```
- web
```

```
volumes:
```

```
- sw_static:/opt/static
```

```
deploy:
```

```
<<: *common_deploy
```

```
labels:
```

```
- "traefik.enable=true"
```

```
- "traefik.http.routers.back-http.rule=Host(\$\$SW_APP_PUBLIC_HOST\`) &&
```

```
PathPrefix(\`/api\`,\`/ws\`)"
```

```
- "traefik.http.routers.back-http.service=back-http-service"
```

```
- "traefik.http.services.back-http-service.loadBalancer.server.port=\${BS_APP_PORT:-8080}"
```

```
- "traefik.http.services.back-http-service.loadBalancer.passHostHeader=true"
```

```
- "traefik.http.middlewares.back-http-
```

```
websocket.headers.customrequestheaders.Host=\$\$SW_APP_PUBLIC_HOST"
```

```
- "traefik.http.middlewares.back-http-websocket.headers.customrequestheaders.X-Forwarded-
```

```
Proto=https"
```

```
- "traefik.http.routers.back-http.middlewares=back-http-websocket"
```

```
- "traefik.http.routers.back-http.entryPoints=web"
```

app-admin:

```
<<: *common
```

```
image: "sw-back:latest"
```

volumes:

- sw_static:/opt/static

command: bash -c "/opt/app/deploy/run/admin.sh"

environment:

BS_DEBUG: "True"

networks:

- traefik
- web

deploy:

<<: *common_deploy

labels:

- "traefik.enable=true"
- "traefik.http.routers.admin-http.rule=Host(`\$SW_APP_ADMIN_HOST`) &&

PathPrefix(`/admin`,`/media`)"

- "traefik.http.routers.admin-http.service=admin-http-service"
- "traefik.http.services.admin-http-service.loadBalancer.server.port=\${BS_ADMIN_PORT:-8081}"
- "traefik.http.services.admin-http-service.loadBalancer.passHostHeader=true"
-

"traefik.http.middlewares.testheader.headers.customrequestheaders.Host=\${SW_APP_ADMIN_HOST}"

- "traefik.http.middlewares.testheader.headers.customrequestheaders.X-Forwarded-Proto=https"
- "traefik.http.routers.admin-http.middlewares=testheader"

app-public-debug:

<<: *common

image: "sw-back:latest"

command: bash -c "/opt/app/deploy/run/app.sh"

environment:

BS_DEBUG: "True"

networks:

- traefik
- web

volumes:

- sw_static:/opt/static

deploy:

<<: *common_deploy

labels:

- "traefik.enable=true"

- "traefik.http.routers.back-debug-http.rule=Host(\`\${SW_APP_ADMIN_HOST}\`) &&

PathPrefix(\`/api\`,\`/ws\`)"

- "traefik.http.routers.back-debug-http.service=back-debug-http-service"

- "traefik.http.services.back-debug-http-service.loadBalancer.server.port=\`\${BS_APP_PORT:-8080}\`"

- "traefik.http.services.back-debug-http-service.loadBalancer.passHostHeader=true"

- "traefik.http.middlewares.back-debug-http-

websocket.headers.customrequestheaders.Host=\`\${SW_APP_ADMIN_HOST}`"

- "traefik.http.middlewares.back-debug-http-websocket.headers.customrequestheaders.X-Forwarded-Proto=https"

- "traefik.http.routers.back-debug-http.middlewares=back-http-websocket"

- "traefik.http.routers.back-debug-http.entryPoints=web"

bg:

<<: *common

image: "sw-back:latest"

command: bash -c "/opt/app/deploy/run/bg.sh"

volumes:

redis:

jaeger_data:

sw_static:

external: true

networks:

web:

external: true

name: \`\${SW_NET}`

traefik:


```
external: true  
name: \${TRAEFIK_NET}
```

EOF

8. Создать файл с переменными окружения для будущих контейнеров в /opt/.env

Переменные обязательные к определению:

```
BS_CRYPTOPRO_LICENSE=лицензия крипто про  
BS_SECRET_KEY=токен для механизма авторизации приложения SW  
BS_DATABASES__DEFAULT__HOST=ип адрес PostgreSQL  
BS_DB_HOST=ип адрес PostgreSQL  
BS_DB_PASSWORD=пароль пользователя Системы в PostgreSQL  
BS_DATABASES__DEFAULT__PASSWORD=пароль пользователя Системы в PostgreSQL  
BS_ADMIN_PASSWORD=пароль в админку  
BS_SERVER_NAME=домен, по которому будет доступен Frontend приложение
```

Переменные не обязательные для редактирования

```
STORAGE_PASS=eiCool7veetooTo  
BS_BAZIS_S3_ACCESS_KEY_ID_BG=59580_sw-secure_user  
BS_BAZIS_S3_ACCESS_KEY_ID_FILE_UPLOAD=59580_sw-secure_user  
BS_BAZIS_S3_ACCESS_KEY_ID_STENCIL=59580_sw-secure_user  
BS_BAZIS_S3_BUCKET_BG=59580_sw-secure_user  
BS_BAZIS_S3_BUCKET_FILE_UPLOAD=59580_sw-secure_user  
BS_BAZIS_S3_BUCKET_STENCIL=sw-common  
BS_BAZIS_S3_ENDPOINT_BG=https://s3.storage.selcloud.ru  
BS_BAZIS_S3_ENDPOINT_FILE_UPLOAD=https://s3.storage.selcloud.ru  
BS_BAZIS_S3_ENDPOINT_STENCIL=https://s3.storage.selcloud.ru  
BS_BAZIS_S3_PROXY_ENDPOINT_BG=https://${BS_SERVER_NAME}/selcloud  
BS_BAZIS_S3_PROXY_ENDPOINT_STENCIL=https://${BS_SERVER_NAME}/selcloud  
BS_BAZIS_S3_PROXY_ENDPOINT_UPLOAD=https://${BS_SERVER_NAME}/selcloud  
BS_BAZIS_S3_REGION_BG=ru-1  
BS_BAZIS_S3_REGION_FILE_UPLOAD=u-1  
BS_BAZIS_S3_REGION_STENCIL=ru-1  
BS_BAZIS_S3_SECRET_ACCESS_KEY_BG=7TI6a#KBIX  
BS_BAZIS_S3_SECRET_ACCESS_KEY_FILE_UPLOAD=7TI6a#KBIX
```

```
BS_BAZIS_S3_SECRET_ACCESS_KEY_STENCIL=7TI6a#KBIX
BS_DATABASES__DEFAULT__PORT=5432
BS_DATABASES__DEFAULT__NAME=sw
BS_DATABASES__DEFAULT__USER=sw
BS_ADMIN_NAME=admin
BS_ADMIN_PORT=8080
BS_APP_PORT=8080
BS_CACHE_BACKEND=django_redis.cache.RedisCache
BS_CACHE_CLIENT_CLASS=django_redis.client.DefaultClient
BS_CACHE_LOCATION=redis://:@redis:6379/1
BS_CACHES__DEFAULT__BACKEND=django_redis.cache.RedisCache
BS_CACHES__DEFAULT__LOCATION=redis://:@redis:6379/1
BS_DB_NAME=sw
BS_DB_PORT=5432
BS_DB_USER=sw
BS_DEBUG=True
BS_ENVIRONMENT_NAME=develop
BS_HOST_URL=https://${BS_SERVER_NAME}
BS_MEDIA_ROOT=/opt/media
BS_MEDIA_URL=/media/
BS_REDIS_HOST=redis
BS_REDIS_PORT=6379
BS_STATIC_ROOT=/opt/static
BS_STATIC_URL=/static/
CONTAINER_NAME=rpn01
```

9. Определить переменные окружения для запуска стэка:

```
export SW_NET=web
export TRAEFIK_NET=traefik
export _BS_APP_ENV_FILE=/opt/.env
export SW_APP_PUBLIC_HOST=домен, по которому будет доступно Frontend приложение
export SW_APP_ADMIN_HOST=домен, по которому будет доступна админка
```

10. Запустить стэк с Django и Frontend:

```
docker stack deploy --compose-file stack.yml smart-waste
```

Проверяем успешность деплоя: `docker stack ps smart-waste`

11. Создать стэк файл с Traefic

```
cat << EOF > traefik.yml
```

```
version: "3.8"
```

```
services:
```

```
traefik:
```

```
image: "traefik:latest"
```

```
networks:
```

```
- web
```

```
command:
```

```
- "--accesslog=true"
```

```
- "--log.level=DEBUG"
```

```
- "--log.format=json"
```

```
- "--providers.docker=true"
```

```
- "--providers.docker.swarmMode=true"
```

```
- "--providers.docker.network=\${TRAEFIK_NETWORK}"
```

```
- "--providers.docker.exposedByDefault=false"
```

```
- "--providers.file.directory=/opt/conf"
```

```
- "--providers.file.watch=true"
```

```
- "--entryPoints.web.address=:80"
```

```
- "--entryPoints.dashboard.address=:8080"
```

```
- "--api.dashboard=true"
```

```
- "--pilot.token=c367d5b1-f83e-4201-b5b2-ffbdf83d8f9b"
```

```
- "--metrics.prometheus=true"
```

```
- "--metrics.prometheus.manualRouting=true"
```

```
ports:
```

```
- "\${PUBLISHED_PORT}:80"
```

```
- "8080:8080"
```

```
volumes:
```

```
- /etc/localtime:/etc/localtime:ro
```

```
- /var/run/docker.sock:/var/run/docker.sock:ro
```

```
- /opt/traefik-config.conf:/opt/conf/traefik-config.yml
```

```
deploy:
```

```
mode: replicated
replicas: 1
restart_policy:
  condition: any
  delay: 5s
  max_attempts: 10
  window: 120s
update_config:
  parallelism: 1
  delay: 10s
  order: stop-first
labels:
  - "traefik.enable=true"
  # Dummy service for Swarm port detection. The port can be any valid integer value.
  - "traefik.http.services.dummy-svc.loadbalancer.server.port=9999"
  - "traefik.http.routers.traefik-dashboard.priority=1"
  - "traefik.http.routers.traefik-dashboard.rule=Host(`$TRAEFIK_HOST`)"
  - "traefik.http.routers.traefik-dashboard.entryPoints=dashboard"
  - "traefik.http.routers.traefik-dashboard.service=api@internal"
  - "traefik.http.routers.traefik-dashboard.middlewares=traefik-auth"
  - "traefik.http.routers.prometheus.rule=Host(`$TRAEFIK_HOST`) && PathPrefix(`/metrics`)"
  - "traefik.http.routers.prometheus.entryPoints=dashboard"
  - "traefik.http.routers.prometheus.service=prometheus@internal"

networks:
  web:
    external: true
    name: \${TRAEFIK_NETWORK}
EOF
```

12. Создать файл настроек для того, чтобы Frontend приложение могло обращаться к с3 через Traefic:

```
export SW_APP_PUBLIC_HOST=домен по которому будет доступно Frontend приложение,  
например sw-test.ru
```

```
export BS_BAZIS_S3_ENDPOINT_BG=урл с3, например https://s3.storage.selcloud.ru
```

```
cat << EOF > /opt/traefik-config.conf
```

```
http:
```

```
middlewares:
```

```
  s3-replacepathregex:
```

```
    replacePathRegex:
```

```
      regex: "^/selcloud/(.*)"
```

```
      replacement: "/$1"
```

```
routers:
```

```
  s3:
```

```
    rule: 'Host(`${SW_APP_PUBLIC_HOST}`) && PathPrefix(`/selcloud`)'
```

```
    service: s3-service
```

```
    middlewares: s3-replacepathregex
```

```
    entryPoints:
```

```
      - web
```

```
services:
```

```
  s3-service:
```

```
    loadBalancer:
```

```
      passHostHeader: false
```

```
    servers:
```

```
      - url: '${BS_BAZIS_S3_ENDPOINT_BG}'
```

```
EOF
```

13. Создать стэк с Traefic:

Определяем переменные окружения для запуска стэка с Traefic.

```
TRAEFIK_NETWORK=traefik
```

```
TRAEFIK_HOST=ип адрес вм на которой будет поднят Traefic
```

```
docker stack deploy --compose-file traefik.yml traefik
```

```
проверяем: docker stack ps traefik
```

Открыть браузер, ввести url с доменом, на котором развернуто Fronted приложение (SW_APP_PUBLIC_HOST).

5.3. Сборка и развертывание Системы для работы в мобильном приложении

5.3.1. Подготовка к сборке .apk файла

1. Распаковать архив sw-mobile-app.zip.
2. В файле \src\environments\environment.prod.ts заменить все экземпляры <https://secure.sw.big3.ru> с URL-адресом API, который будет использовать собираемое приложение.

5.3.2. Создание .apk файла

node v18, npm v9

Находясь в корневом каталоге, поочередно выполните следующие команды:

```
npm install -g @ionic/cli
```

```
npm i --force
```

```
ionic build --configuration=production
```

```
ionic capacitor copy android
```

```
ionic capacitor update
```

Далее откройте папку android (путь от корня вашего репозитория) в вашей Android studio

Menu: Build → Build bundle(s) / APK(s) → Build APK(s)

Когда процесс сборки будет завершен, вы получите уведомление в Android Studio со ссылкой на целевой каталог, в котором будет создан ваш apk-файл.

Или окончательный файл .apk файл можно найти в папке

\android\app\build\outputs\apk\debug (путь от корня вашего репозитория).

Эта папка содержит отладочную версию файла APK, которую вы можете установить на свое устройство Android.

6. ЗАПУСК

6.1. Запуск Системы для работы в веб-приложении

Клиентская часть системы функционирует в рамках веб-браузера и запускается автоматически в контексте веб-браузера. Открыв веб-браузер и указав в адресной строке URL-адрес, происходит запуск клиентской части Системы (в контексте веб-браузера) и отображается страница входа в Систему (см. рис.1 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)).

6.2. Запуск Системы для работы в мобильном приложении

Установив на мобильное устройство и запустив арк-файл, будет отображаться стартовая страница Системы (см. рис. 214 «Руководство оператора» (RU.1107746574308.068-01 34 01)).

7. ПРОВЕРКА ФУНКЦИОНИРОВАНИЯ

7.1. Проверка функционирования Системы при работе в веб-приложении

В таблице 1 приведен порядок выполнения действий для проверки функционирования Системы при работе в веб-приложении.

Таблица 1 – Порядок проверки функционирования Системы при работе в веб-браузере

№ шага	Цель проверки	Действие	Ожидаемый результат
1	Развертывание Системы	Выполнить действия, описанные в п. 5.2 настоящего документа	Стартовая страница Системы открывается
2	Регистрация компании в Системе	Выполнить действия, описанные в п. 4.3 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. Компания зарегистрирована в Системе. 2. Назначены дополнительные роли администратору компании. 3. Добавлен сотрудник компании
3	Проверка функционирования Системы при работе под ролью «Администратор компании»		
3.1	Выполнить вход в Систему (процедура аутентификации и авторизации)	Выполнить действия, описанные в п. 4.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	Стартовая страница Системы с дополнительными функциональными элементами открывается
3.2	Проверка функционирования личного кабинета пользователя в разделе «Профиль»	Выполнить действия, описанные в п. 4.4.1.1 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Профиль» отображается информация о пользователе, его организации, видах деятельности, платежные реквизиты, контактные данных представителей. 2. Редактирование данных выполняется корректно
3.3	Проверка функционирования личного кабинета пользователя в разделе «Настройки»	Выполнить действия, описанные в п. 4.4.1.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Настройки» отображается подраздел «Вход и безопасность». 2. Редактирование данных выполняется корректно
3.4	Проверка функционирования Системы в разделе «Пользователи»	Выполнить действия, описанные в п. 4.4.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Пользователи» отображается список профилей всех сотрудников компании, в том числе и профиль администратора компании. 2. Фильтрация и сортировка данных выполняется корректно
3.5	Проверка функционирования Системы в разделе «Запросы на	Выполнить действия, описанные в п. 4.4.3 документа «Руководство оператора»	1. В разделе «Запросы на полномочия» отображается список всех запросов на

	полномочия»	(RU.1107746574308.068-01 34 01)	полномочия, которые направляли сотрудники компании. 2. Фильтрация и сортировка данных выполняется корректно. 3. Страница запроса на полномочия открывается, данные отображаются корректно. 4. Действия с запросами на полномочия выполняются корректно
4	Проверка функционирования Системы при работе под ролью «Отходообразователь»		
4.1	Выполнить вход в Систему (процедура аутентификации и авторизации)	Выполнить действия, описанные в п. 4.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	Стартовая страница Системы с дополнительными функциональными элементами открывается
4.2	Проверка функционирования личного кабинета пользователя в разделе «Профиль»	Выполнить действия, описанные в п. 4.5.1.1 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Профиль» отображается информация о пользователе, его организации, контактных данных представителей. 2. Редактирование данных выполняется корректно
4.3	Проверка функционирования личного кабинета пользователя в разделе «Счет»	Выполнить действия, описанные в п. 4.5.1.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Счет» отображаются четыре подраздела: «Пополнить счет», «Вывод средств», «История операций», «Резервирования». 2. В подразделе «Пополнить счет» операция пополнения баланса и формирования счета выполняется корректно. 3. В подразделе «Вывод средств» операция отправки запроса на вывод денежных средств из Системы на счет пользователя выполняется корректно
4.4	Проверка функционирования личного кабинета пользователя в разделе «Уведомления»	Выполнить действия, описанные в п. 4.5.1.3 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Уведомления» отображается полный перечень уведомлений. 2. Фильтрация и сортировка данных выполняется

			корректно
4.5	Проверка функционирования личного кабинета пользователя в разделе «Настройки»	Выполнить действия, описанные в п. 4.5.1.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<ol style="list-style-type: none"> 1. В разделе «Настройки» отображаются два подраздела: «Уведомления и рассылки», «Вход и безопасность». 2. Редактирование данных выполняется корректно
4.6	Проверка функционирования Системы в разделе «Площадки»	Выполнить действия, описанные в п. 4.5.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<ol style="list-style-type: none"> 1. В разделе «Площадки» отображается перечень площадок (мест накопления отходов), которые использует организация. 2. Фильтрация и сортировка данных выполняется корректно. 3. Страница площадки открывается, данные отображаются корректно. 4. Переходы по разделам страницы площадки осуществляются корректно. 5. Редактирование данных выполняется корректно
4.7	Проверка функционирования Системы в разделе «Создание заявок и заказов»	Выполнить действия, описанные в п. 4.5.3 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<ol style="list-style-type: none"> 1. Ссылка «Создать заявку» корректно отображает всплывающее меню с выбором типа заявки. При выборе пункта меню осуществляется переход в соответствующий подраздел. 2. Фильтрация и сортировка данных выполняется корректно. 3. Операция создания заявки на разовый договор выполняется корректно. 4. Операция создания заявки на долгосрочный договор выполняется корректно. 5. Операция создания заказа к долгосрочному договору выполняется корректно. 6. Редактирование данных выполняется

			корректно
4.8	Проверка функционирования Системы в разделе «Долгосрочные договоры»	Выполнить действия, описанные в п. 4.5.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Долгосрочные договоры» отображается список всех долгосрочных договоров, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница долгосрочного договора открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы долгосрочного договора осуществляются корректно.</p> <p>5. Операция печати страницы долгосрочного договора с предварительным ее просмотром выполняется корректно.</p> <p>6. Операция создания нового заказа к текущему договору через вкладку «Детальная информация» выполняется корректно.</p> <p>7. Выгрузка данных вкладки «Детализация стоимости» на компьютер пользователя в формате .XLSX выполняется корректно.</p> <p>8. Выгрузка данных вкладки «Документы» на компьютер пользователя в формате .PDF выполняется корректно.</p> <p>9. Печать выбранного документа во вкладке «Документы» с предварительным его просмотром выполняется корректно</p>
4.9	Проверка функционирования Системы в разделе «Мои заявки»	Выполнить действия, описанные в п. 4.5.5 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Мои заявки» отображается список всех заявок, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется</p>

			<p>корректно.</p> <p>3. Страница заявки открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы заявки осуществляются корректно.</p> <p>5. Операция снятия заявки на заключения договора выполняется корректно.</p> <p>6. Опция продления сбора предложение работает корректно.</p> <p>7. Операции подписания/отказ от подписания договора выполняются корректно</p> <p>8. Операции изменения/снятие заявки на выполнение заказа выполняются корректно.</p> <p>9. Операции подписания/отклонения подписания акта оказанных услуг по заказу выполняются корректно</p>
4.10	Проверка функционирования Системы в разделе «Заказы»	Выполнить действия, описанные в п. 4.5.6 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Заказы» отображается список всех заказов, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница заказа открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы заказа осуществляются корректно.</p> <p>5. Печать страницы заказа с предварительным ее просмотром выполняется корректно.</p> <p>6. Печать выбранного документа во вкладке «Документы» с предварительным его просмотром выполняется корректно</p>
4.11	Проверка функционирования	Выполнить действия, описанные в п. 4.5.7	<p>1. В разделе «Рейсы» отображаются сведения о</p>

	Системы в разделе «Рейсы»	документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	запланированных, осуществляющихся или завершенных рейсах транспорта для вывоза отходов или доставки тары. 2. Фильтрация и сортировка данных выполняется корректно. 3. Страница рейса открывается и данные отображаются корректно. 4. Переходы по разделам страницы рейса осуществляются корректно
5	Проверка функционирования Системы при работе под ролью «Перевозчик»		
5.1	Выполнить вход в Систему (процедура аутентификации и авторизации)	Выполнить действия, описанные в п. 4.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	Стартовая страница Системы с дополнительными функциональными элементами открывается
5.2	Проверка функционирования личного кабинета пользователя в разделе «Профиль»	Выполнить действия, описанные в п. 4.6.1.1 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Профиль» отображается информация о пользователе, его организации, контактных данных представителей, лицензиях на деятельность по обращению с отходами. 2. Редактирование данных выполняется корректно
5.3	Проверка функционирования личного кабинета пользователя в разделе «Счет»	Выполнить действия, описанные в п. 4.6.1.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	В разделе «Счет» отображаются два подраздела: «История операций», «Резервирования»
5.4	Проверка функционирования личного кабинета пользователя в разделе «Уведомления»	Выполнить действия, описанные в п. 4.6.1.3 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Уведомления» отображается полный перечень уведомлений. 2. Фильтрация и сортировка данных выполняется корректно
5.5	Проверка функционирования личного кабинета пользователя в разделе «Агентский договор»	Выполнить действия, описанные в п. 4.6.1.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. В разделе «Агентский договор» отображаются два подраздела: «Основное», «Отчетные документы». 2. В подразделе «Основное» печать агентского договора с предварительным его просмотром

			<p>выполняется корректно.</p> <p>3. В подразделе «Отчетные документы» скачивание отчетного документа по агентскому договору на компьютер пользователя выполняется корректно</p>
5.6	Проверка функционирования личного кабинета пользователя в разделе «Настройки»	Выполнить действия, описанные в п. 4.6.1.5 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Настройки» отображаются два подраздела: «Уведомления и рассылки», «Вход и безопасность».</p> <p>2. Редактирование данных выполняется корректно</p>
5.7	Проверка функционирования Системы в разделе «Лоты»	Выполнить действия, описанные в п. 4.6.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Лоты» отображается перечень всех лотов, существующих в Системе в реестре лотов на вывоз отходов.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница лотов открывается, данные отображаются корректно.</p> <p>4. Страница с подробной информацией о выбранном лоте открывается, данные отображаются корректно.</p> <p>5. Переходы по разделам страницы с подробной информацией о выбранном лоте осуществляются корректно.</p> <p>6. Операция подачи предложения о перевозке выполняется корректно</p>
5.8	Проверка функционирования Системы в разделе «Заказы»	<p>1. Ознакомиться с п. 4.6. документа «Руководство оператора» (RU.1107746574308.068-01 34 01).</p> <p>2. Выполнить действия, описанные в п. 4.5.6 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)</p>	<p>1. В разделе «Заказы» отображается список всех заказов, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница заказа открывается, данные отображаются корректно.</p>

			<p>4. Переходы по разделам страницы заказа осуществляются корректно.</p> <p>5. Печать страницы заказа с предварительным ее просмотром выполняется корректно.</p> <p>6. Печать выбранного документа во вкладке «Документы» с предварительным его просмотром выполняется корректно</p>
5.9	Проверка функционирования Системы в разделе «Долгосрочные договоры»	<p>1. Ознакомиться с п. 4.6.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01).</p> <p>2. Выполнить действия, описанные в п. 4.5.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)</p>	<p>1. В разделе «Долгосрочные договоры» отображается список всех долгосрочных договоров, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница долгосрочного договора открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы долгосрочного договора осуществляются корректно.</p> <p>5. Печать страницы долгосрочного договора с предварительным ее просмотром выполняется корректно.</p> <p>6. Операция создания нового заказа к текущему договору через вкладку «Детальная информация» выполняется корректно.</p> <p>7. Выгрузка данных вкладки «Детализация стоимости» на компьютер пользователя в формате .XLSX выполняется корректно.</p> <p>8. Выгрузка данных вкладки «Документы» на компьютер пользователя в формате .PDF выполняется корректно.</p> <p>9. Печать выбранного документа во вкладке «Документы» с предварительным его</p>

			просмотром выполняется корректно
5.10	Проверка функционирования Системы в разделе «Рейсы»	<p>1. Ознакомиться с п. 4.6.5 документа «Руководство оператора» (RU.1107746574308.068-01 34 01).</p> <p>2. Выполнить действия, описанные в п. 4.5.7 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)</p>	<p>1. В разделе «Рейсы» отображаются сведения о запланированных, осуществляющихся или завершенных рейсах транспорта для вывоза отходов или доставки тары.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница рейса открывается и данные отображаются корректно.</p> <p>4. Переходы по разделам страницы рейса осуществляются корректно</p>
5.11	Проверка функционирования Системы в разделе «Мои транспортные средства»	Выполнить действия, описанные в п. 4.6.6 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Мои транспортные средства» отображается перечень транспортных средств, которые использует организация.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница транспортного средства открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы транспортного средства осуществляются корректно.</p> <p>5. Редактирование данных выполняется корректно</p>
5.12	Проверка функционирования Системы в разделе «Настройка телеметрии»	Выполнить действия, описанные в п. 4.6.7 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Настройка телеметрии» перечень подключенных сервисов для передачи телеметрии, подключенных пользователем.</p> <p>2. Доступ к перечню поставщиков выполняется корректно.</p> <p>3. Операция удаления неиспользуемых протоколов выполняется корректно.</p> <p>4. Страница приложения интеграции</p>

			<p>открывается, данные отображаются корректно.</p> <p>5. Фильтрация и сортировка данных выполняется корректно.</p> <p>6. Операция поиска трекера выполняется корректно.</p> <p>7. Трек отдельного транспортного средства на интерактивной карте отображается корректно.</p> <p>8. Привязка трекера к другому транспортному средству выполняется корректно</p>
6	Проверка функционирования Системы при работе под ролью «Переработчик»		
6.1	Выполнить вход в Систему (процедура аутентификации и авторизации)	Выполнить действия, описанные в п. 4.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	Стартовая страница Системы с дополнительными функциональными элементами открывается
6.2	Проверка функционирования личного кабинета пользователя в разделе «Профиль»	Выполнить действия, описанные в п. 4.7.1.1 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Профиль» отображается информация о пользователе, его организации, контактных данных представителей, лицензиях на деятельность по обращению с отходами.</p> <p>2. Редактирование данных выполняется корректно</p>
6.3	Проверка функционирования личного кабинета пользователя в разделе «Счет»	Выполнить действия, описанные в п. 4.7.1.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	В разделе «Счет» отображаются два подраздела: «История операций», «Резервирования»
6.4	Проверка функционирования личного кабинета пользователя в разделе «Уведомления»	Выполнить действия, описанные в п. 4.7.1.3 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Уведомления» отображается полный перечень уведомлений.</p> <p>2. Фильтрация и сортировка данных выполняется корректно</p>
6.5	Проверка функционирования личного кабинета пользователя в разделе «Агентский договор»	Выполнить действия, описанные в п. 4.7.1.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Агентский договор» отображаются два подраздела: «Основное», «Отчетные документы».</p> <p>2. В подразделе «Основное» печать агентского</p>

			<p>договора с предварительным его просмотром выполняется корректно.</p> <p>3. В подразделе «Отчетные документы» скачивания отчетного документа по агентскому договору на компьютер пользователя выполняется корректно</p>
6.6	Проверка функционирования личного кабинета пользователя в разделе «Настройки»	Выполнить действия, описанные в п. 4.7.1.5 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Настройки» отображаются два подраздела: «Уведомления и рассылки», «Вход и безопасность».</p> <p>2. Редактирование данных выполняется корректно</p>
6.7	Проверка функционирования Системы в разделе «Объекты обращения»	Выполнить действия, описанные в п. 4.7.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Объекты обращения» отображается перечень объектов обращения с отходами, которые использует организация.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница объекта обращения открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы объектов обращения осуществляются корректно.</p> <p>5. Редактирование данных выполняется корректно</p>
6.8	Проверка функционирования Системы в разделе «Публичные оферты»	Выполнить действия, описанные в п. 4.7.3 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. В разделе «Публичные оферты» отображается список оферт, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница публичной оферты открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы публичной оферты осуществляются корректно.</p>

			<p>5. Выгрузка данных публичной оферты в формате XLSX выполняется корректно.</p> <p>6. Выгрузка данных вкладки «Документы» на компьютер пользователя в формате .PDF выполняется корректно.</p> <p>7. Печать выбранного документа во вкладке «Документы» с предварительным его просмотром выполняется корректно</p> <p>8. Операция внесения изменений в оферту и отзыв оферты выполняется корректно.</p> <p>9. Операция создания публичной оферты выполняется корректно</p>
6.9	Проверка функционирования Системы в разделе «Заказы»	<p>1. Ознакомиться с п. 4.7.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01).</p> <p>2. Выполнить действия, описанные в п. 4.5.6 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)</p>	<p>1. В разделе «Заказы» отображается список всех заказов, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница заказа открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы заказа осуществляются корректно.</p> <p>5. Печать страницы заказа с предварительным ее просмотром выполняется корректно.</p> <p>6. Печать выбранного документа во вкладке «Документы» с предварительным его просмотром выполняется корректно</p>
6.10	Проверка функционирования Системы в разделе «Долгосрочные договоры»	<p>1. Ознакомиться с п. 4.7.5 документа «Руководство оператора» (RU.1107746574308.068-01 34 01).</p> <p>2. Выполнить действия, описанные в п. 4.5.4 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)</p>	<p>1. В разделе «Долгосрочные договоры» отображается список всех долгосрочных договоров, которые были созданы пользователем.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p>

			<p>3. Страница долгосрочного договора открывается, данные отображаются корректно.</p> <p>4. Переходы по разделам страницы долгосрочного договора осуществляются корректно.</p> <p>5. Печать страницы долгосрочного договора с предварительным ее просмотром выполняется корректно.</p> <p>6. Операция создания нового заказа к текущему договору через вкладку «Детальная информация» выполняется корректно.</p> <p>7. Выгрузка данных вкладки «Детализация стоимости» на компьютер пользователя в формате .XLSX выполняется корректно.</p> <p>8. Выгрузка данных вкладки «Документы» на компьютер пользователя в формате .PDF выполняется корректно.</p> <p>9. Печать выбранного документа во вкладке «Документы» с предварительным его просмотром выполняется корректно</p>
6.11	Проверка функционирования Системы в разделе «Рейсы»	<p>1. Ознакомиться с п. 4.7.6 документа «Руководство оператора» (RU.1107746574308.068-01 34 01).</p> <p>2. Выполнить действия, описанные в п. 4.5.7 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)</p>	<p>1. В разделе «Рейсы» отображаются сведения о запланированных, осуществляющихся или завершенных рейсах транспорта для вывоза отходов или доставки тары.</p> <p>2. Фильтрация и сортировка данных выполняется корректно.</p> <p>3. Страница рейса открывается и данные отображаются корректно.</p> <p>4. Переходы по разделам страницы рейса осуществляются корректно</p>

7.2. Проверка функционирования Системы при работе в мобильном приложении

В таблице 2 порядок выполнения действий для проверки функционирования Системы при работе в мобильном приложении.

Таблица 2 – Порядок проверки функционирования Системы при работе в мобильном приложении

№ шага	Цель проверки	Действие	Ожидаемый результат
1	Развертывание Системы	Выполнить действия, описанные в п. 5.3 настоящего документа	1. Стартовая страница Системы открывается. 2. Переходы по разделам страницы площадки осуществляются корректно
2	Проверка функционирования Системы при работе под ролью «Водитель»		
2.1	Выполнить вход в Систему (процедура аутентификации и авторизации)	Выполнить действия, описанные в п. 5.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. Экран «Рейсы» открывается, данные отображаются корректно 2. Запрос приложения для доступа к местоположению устройства выполняется корректно
2.2	Проверка функционирования Системы в разделе «Рейсы»	Выполнить действия, описанные в п. 5.3.1 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	1. Фильтрация и сортировка данных выполняется корректно. 2. Детальная форма рейса отображается корректно. 3. Операция создания отчета о прибытии на площадку выполняется корректно. 4. Операция создания отчета о завершении погрузки выполняется корректно. 5. Операция создания отчета о прибытии на объект выполняется корректно. 6. Операция создания отчета о завершении выгрузки выполняется корректно. 7. Операция создания отчета о невозможности выполнения рейса выполняется корректно. 8. Операция отмены созданного отчета выполняется корректно
2.3	Проверка функционирования	Выполнить действия, описанные в п. 5.3.2 документа «Руководство оператора»	1. Редактирование данных в подразделе

	Системы в разделе «Еще»	(RU.1107746574308.068-01 34 01)	<p>«Профиль» выполняется корректно.</p> <p>2. Связь со службой поддержкой выполняется корректно.</p> <p>3. Выход из Системы выполняется корректно</p>
3	Проверка функционирования Системы при работе под ролью «Приемщик»		
3.1	Выполнить вход в Систему (процедура аутентификации и авторизации)	Выполнить действия, описанные в п. 5.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. Экран «Рейсы» открывается, данные отображаются корректно</p> <p>2. Запрос приложения для доступа к местоположению устройства выполняется корректно</p>
3.2	Проверка функционирования Системы в разделе «Рейсы»	Выполнить действия, описанные в п. 5.4.1 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. Фильтрация и сортировка данных выполняется корректно.</p> <p>2. Детальная форма рейса отображается корректно.</p> <p>3. Операция создания отчета о принятии рейса выполняется корректно.</p> <p>4. Операция создания отчета об отказе в приемке рейса выполняется корректно.</p> <p>5. Операция отмены созданного отчета выполняется корректно</p>
3.3	Проверка функционирования Системы в разделе «Еще»	Выполнить действия, описанные в п. 5.4.2 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	<p>1. Редактирование данных в подразделе «Профиль» выполняется корректно.</p> <p>2. Связь со службой поддержкой выполняется корректно.</p> <p>3. Выход из Системы выполняется корректно</p>

