

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ МАТЕМАТИКИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

09.03.04 - Программная инженерия

Отчет по учебному курсу «Верификация ПО»  
Тестирование интерфейса информационной системы  
экологического мониторинга

Выполнил:  
Н. И. Жидов, гр. 22407

# Содержание

<b>Описание объекта тестирования</b>	<b>3</b>
<b>1 Стратегия тестирования</b>	<b>4</b>
1.1 Архитектура объекта тестирования . . . . .	4
1.2 Стратегия блочного тестирования . . . . .	9
1.3 Стратегия интеграционного тестирования . . . . .	10
1.4 Стратегия аттестационного тестирования . . . . .	13
1.5 Стратегия тестирования производительности . . . . .	13
1.6 Условия начала, окончания и перехода между этапами тестирования. . . . .	14
1.7 Условия возобновления и приостановки выполнения тестов. . . . .	14
<b>2 Детальный план тестирования</b>	<b>15</b>
2.1 Блочное тестирование . . . . .	15
2.2 Интеграционное тестирование . . . . .	21
2.3 Аттестационное тестирование . . . . .	26
2.4 Тестирование производительности . . . . .	31
<b>3 Примеры тестов</b>	<b>39</b>
<b>4 Журнал тестирования</b>	<b>39</b>
<b>5 Журнал ошибок</b>	<b>41</b>
<b>6 Оценка покрытия кода тестами</b>	<b>41</b>
<b>7 Заключение</b>	<b>41</b>

## Описание объекта тестирования

Информационная система экологического мониторинга представляет собой веб-приложение, разработанное с использованием современных технологий, включая Typescript, библиотеку React для разработки интерфейса, Axios для работы с сервером, Formik для продвинутой работы с формами в React и Zustand для управления состоянием приложения. Основной целью этой системы является сбор и визуализация данных, связанных с экологическим мониторингом, что позволяет пользователям мониторить состояние окружающей среды и устройств-датчиков.

Основные характеристики объекта тестирования:

1. Авторизация и безопасность: Система включает механизм авторизации, валидации данных и управления токенами авторизации, обеспечение разделения доступа к разным частям интерфейса.
2. Основная функциональность системы - отображение данных с устройств-датчиков. Эти данные представлены в виде списка карточек с указанием последних снятых данных, статуса в сети/не в сети и времени снятия данных с учетом часового пояса пользователя и устройства.

# 1 Стратегия тестирования

## 1.1 Архитектура объекта тестирования

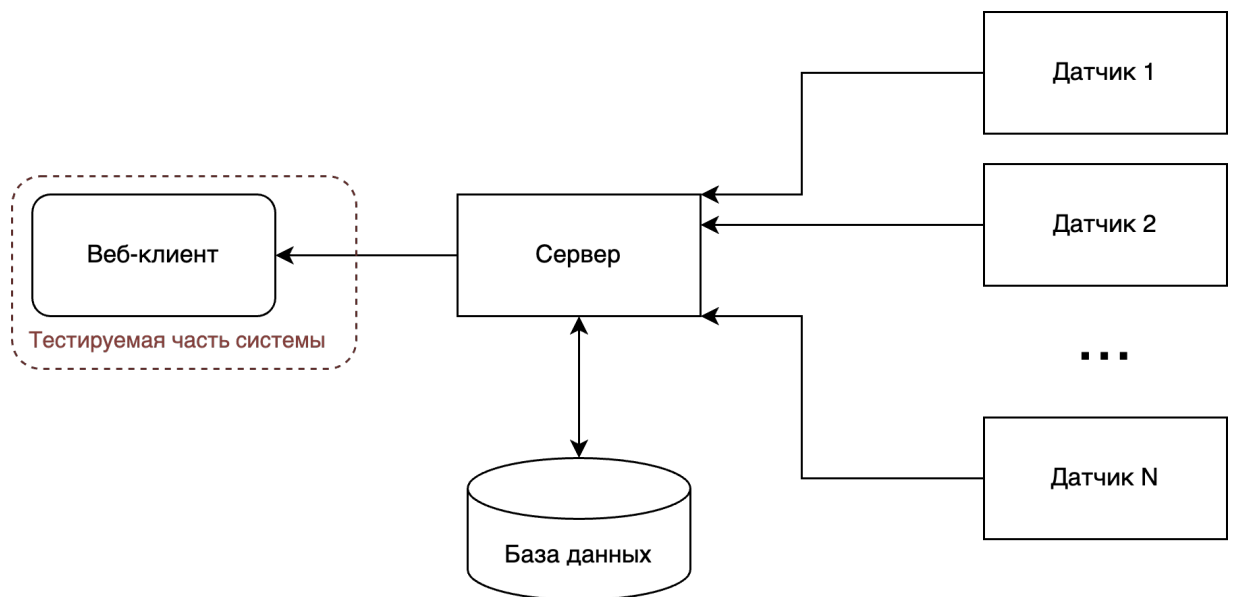


Рис. 1: Высокоуровневая архитектура всей системы

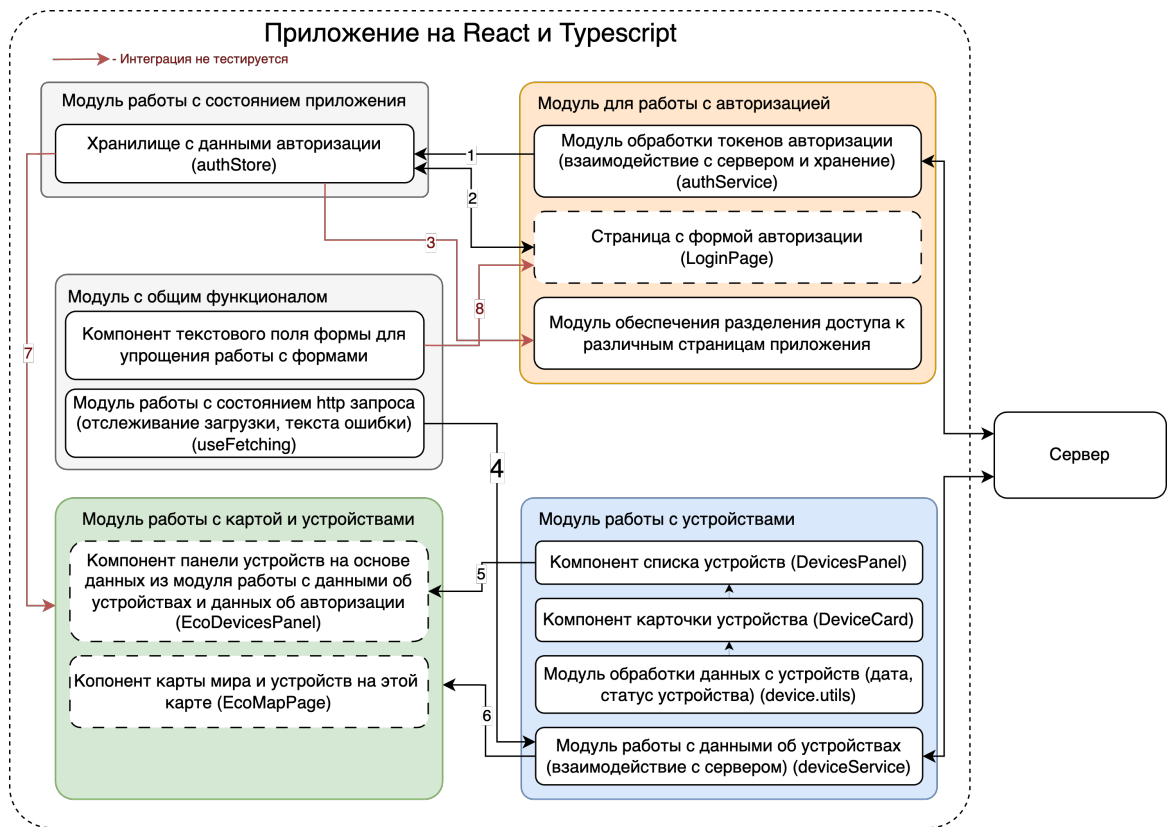


Рис. 2: Архитектура клиентской части системы

Описание модулей в системе:

1. Модуль для работы с авторизацией. Содержит:

- Объект `authService` (Модуль обработки токенов авторизации) с методами:
  - `clearAccessToken()` - удаление токена авторизации из локального хранилища,
  - `saveAccessToken(token)` - сохранение токена авторизации в локальном хранилище,
  - `getAccessToken()` - получение токена авторизации из локального хранилища
- Компонент `LoginPage` (страница с формой авторизации). Содержит поля ввода логина и пароля для входа в систему.
- Функция `authGuardedRoute(component)` (Модуль обеспечения разделения доступа к различным страницам приложения). Не будет тестироваться, так как не используется на данный момент.

2. Модуль работы с состоянием приложения. Представляет из себя объект `authStore`, который содержит:

- флаг `isAuth` - отражает авторизован ли пользователь
- метод `login(login, password)` - осуществляет вход пользователя в систему
- метод `logout()` - осуществляет выход пользователя из системы

### 3. Модуль работы с общим функционалом. Содержит:

- Компонент поля ввода `EcoTextField`. На данный момент не используется, поэтому тестироваться не будет.
- Функция `useFetching(method)` - функция для отслеживания состояния асинхронных операций (например, запрос на сервер).

### 4. Модуль работы с устройствами. Содержит:

- Объект `deviceService` с методом `getAllDevices()` для получения всех устройств в системе.
- Модуль обработки данных с устройств:  
`getDeviceLastDataDate()` - функция получения даты получения последних данных с датчика устройства с учетом часового пояса устройства
- Компонент карточки устройства `DeviceCard`
- Компонент списка устройств `DevicesPanel` (содержит карточки устройств `DeviceCard`)

### 5. Модуль работы с картой и устройствами. Содержит:

- Компонент `EcoDevicesPanel` содержит список устройств доступных в системе.
- Компонент `EcoMapPage` - главная страница в системе, на которой содержатся все устройства в системе и информация о них.

### Описание основных интеграций в системе:

1. Метод `logout()` в Хранилище с данными авторизации использует методы `authService.clearAccessToken()` из Модуля обработки токенов авторизации. Метод `login(token)` использует метод `authService.saveAccessToken(token)` (сохранение `token` в `local storage`). Инициализация флага `isAuth` (в Хранилище) использует метод `authService.getAccessToken()` (получение значения `token` из `local storage`).
2. В зависимости от флага `isAuth` из Хранилища с данными авторизации на Странице с формой авторизации (`LoginPage`) отображается форма входа в систему или сообщение о том, что пользователь уже авторизован. Страница с формой авторизации

использует метод `login(login, password)` из Хранилища для авторизации пользователя.

3. Модуль обеспечения разделения доступа использует флаг `isAuth` из Хранилища данных для авторизации, чтобы определять доступ к частям системы (не тестируется в связи с тем, что не используется на данном этапе разработки системы).
4. Модуль работы данными об устройствах использует функцию `useFetching(method)` из Модуля работы с `http` запросами, чтобы определить состояние (загрузка, ошибка) `http` запроса (передача метода `auth.getAllDevices` в функцию `useFetching` для получение нового объекта, который отражает статус запроса списка устройств).
5. Модуль работы с картой и устройствами (компонент `EcoMapPage`) использует Компонент списка устройств `DevicePanel` для отображения доступных устройств на главной странице (передача списка устройств в компонент `DevicePanel`).
6. Модуль работы с картой и устройствами (компонент `EcoMapPage`) использует метод `deviceService.getAllDevices()` из Модуля работы с данными об устройствах для получения и отображения списка доступных устройств.
7. Модуль работы с картой и устройствами (компонент `EcoMapPage`) использует флаг `isAuth` из Хранилища с данными авторизации, чтобы отображать кнопку "Войти" или "Выйти" (не тестируется в связи с простотой реализации).
8. Компонент текстового поля формы для упрощения работы с формами и Страница с формой авторизации (не тестируется в связи с тем, что большая часть кода - код библиотеки, которая и так протестирована).

Основные части клиентского приложения:

1. Модуль работы с устройствами. Включает в себя все необходимые компоненты для получения данных об устройствах и отображения их в интерфейсе. Необходимо протестировать отображение данных в интерфейсе и обработку данных с устройств.
2. Модуль работы с авторизацией. Включает в себя все компоненты для работы с механизмом авторизации в системе на клиентской стороне. Необходимо протестировать механизм авторизации в приложении и реализацию разделения доступов.
3. Модуль работы с состоянием приложения. Обеспечивает хранение данных, необходимых для функционирования всего приложения. В частности, занимается хранением

данных об авторизации. Необходимо протестировать состояние интерфейса при различных состояниях этого хранилища, а также реакцию приложения на изменение состояния хранилища.

4. Модуль работы с общим функционалом. Обеспечивает клиентское приложение компонентами, упрощающими процесс разработки нового функционала: модуль обработки состояни http запросов и компонент текстового поля формы для упрощения работы с формами. Необходимо протестировать механизм обработки http запросов и работу с формами, используя данный модуль.
5. Модуль работы с картой и устройствами. Модуль отвечает за отображение карты с устройствами на этой карте, а также панель со списком устройств. Работа модуля зависит от всех остальных модулей. Необходимо протестировать отображение устройств при разных состояниях приложения.

Функциональность объекта тестирования:

1. Авторизация в приложении
2. Выход из системы
3. Просмотр списка доступных устройств, и данных с датчиков этих устройств
4. Просмотр списка доступных устройств на карте мира



## 1.2 Стратегия блочного тестирования

Для реализации блочного тестирования будут использованы библиотеки:

1. React Testing Library - это библиотека для тестирования React-приложений, ориентированная на симуляцию взаимодействия пользователя с приложением и проверку его поведения на основе компонентов и пользовательского интерфейса, вместо внутренней логики компонентов.
2. Jest - это библиотека для тестирования JavaScript-кода, предназначенная для написания и выполнения юнит-тестов и интеграционных тестов. Она обеспечивает инструменты для создания и автоматизации тестов в JavaScript-приложениях.

В рамках стратегии блочных тестов каждый модуль клиентского приложения будет подвергнут тестированию с учетом ключевых сценариев использования. Для оценки результатов тестирования будут задействованы специализированные утилиты, предоставляемые вышеупомянутыми библиотеками. Эти утилиты создадут полноценный отчет, отражающий результаты выполнения блочных тестов и обеспечивающий исчерпывающую информацию о качестве приложения.

В рамках блочного тестирования будут проверены следующие части системы:

1. Модуль обработки токенов авторизации: `authService.getAccessToken()` - получение токена из local storage.
2. Модуль обработки токенов авторизации: `authService.saveAccessToken(token: string)` - сохранение токена в local storage.
3. Модуль обработки токенов авторизации: `authService.clearAccessToken()` - очистка токена из local storage
4. Модуль работы с устройствами: `DeviceCard` - компонент карточки устройства.
5. Модуль работы с устройствами: `DevicesPanel` - компонент панели с карточками.
6. Модуль работы с устройствами: `getDeviceLastDataDate()` - функция получения даты получения последних данных с датчика устройства с учетом часового пояса устройства (Модуль обработки данных с устройств).

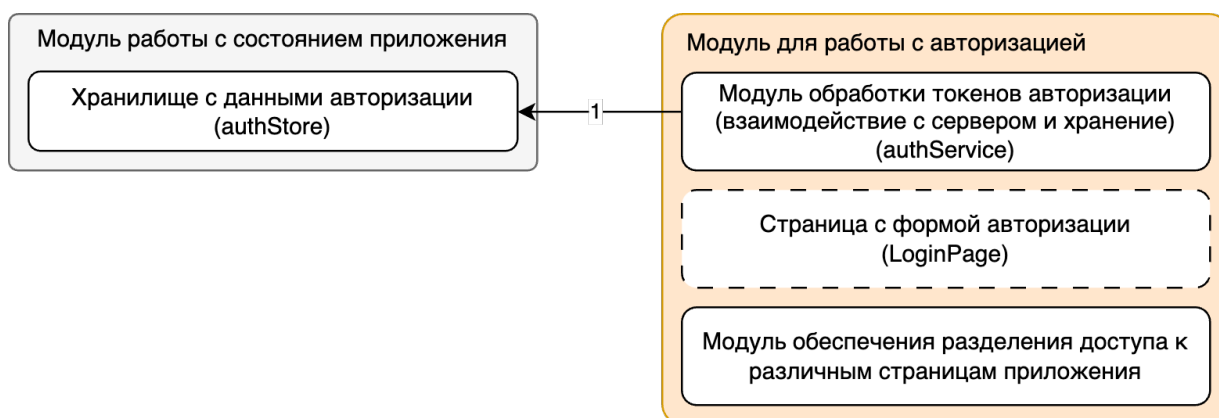
Остальные модули, не упомянутые здесь, либо не будут подвергнуты тестированию (по причине их неприменения или "нечувствительности" к ошибкам), либо будут протестированы в рамках интеграционного тестирования.

### 1.3 Стратегия интеграционного тестирования

Интеграционные тесты будут реализованы с использованием тех же библиотек и окружения, что и для блочного тестирования. Но будет тестироваться взаимодействие между модулями. Интеграция для тестирования будет производиться с учетом ключевых сценариев использования. Оцениваться результаты тестирования будут тем же способом, что и для блочного тестирования.

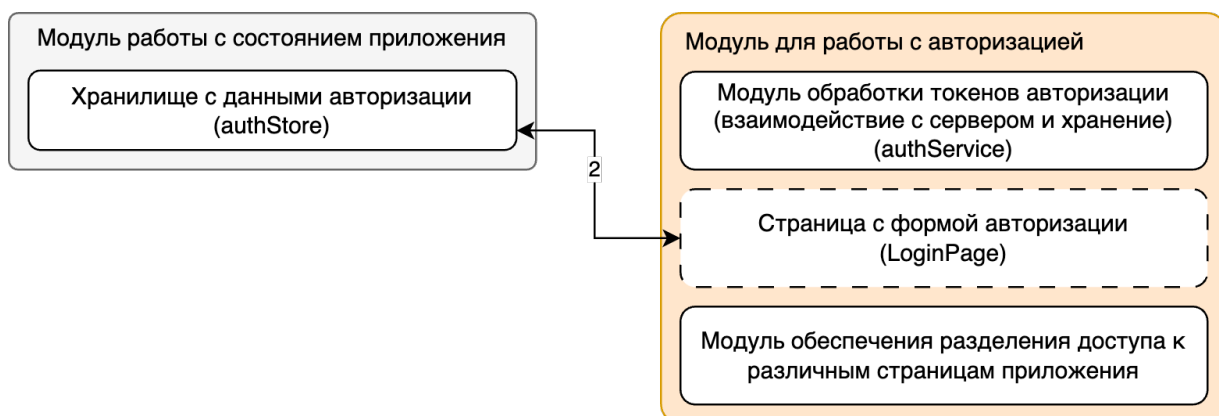
Интеграции в системе:

1. Хранилище авторизации и модуль для работы авторизацией.



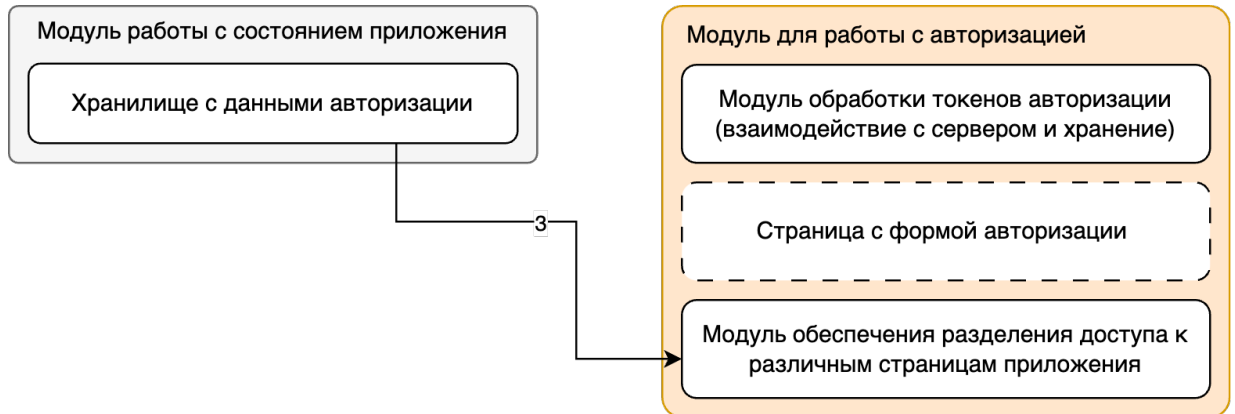
authStore использует метод authService.login(login, password) из Модуля обработки токенов авторизации для получения токена доступа. Используется метод authService.saveAccessToken(token) для сохранения токена доступа в local storage.

2. Хранилище авторизации и модуль для работы авторизацией.



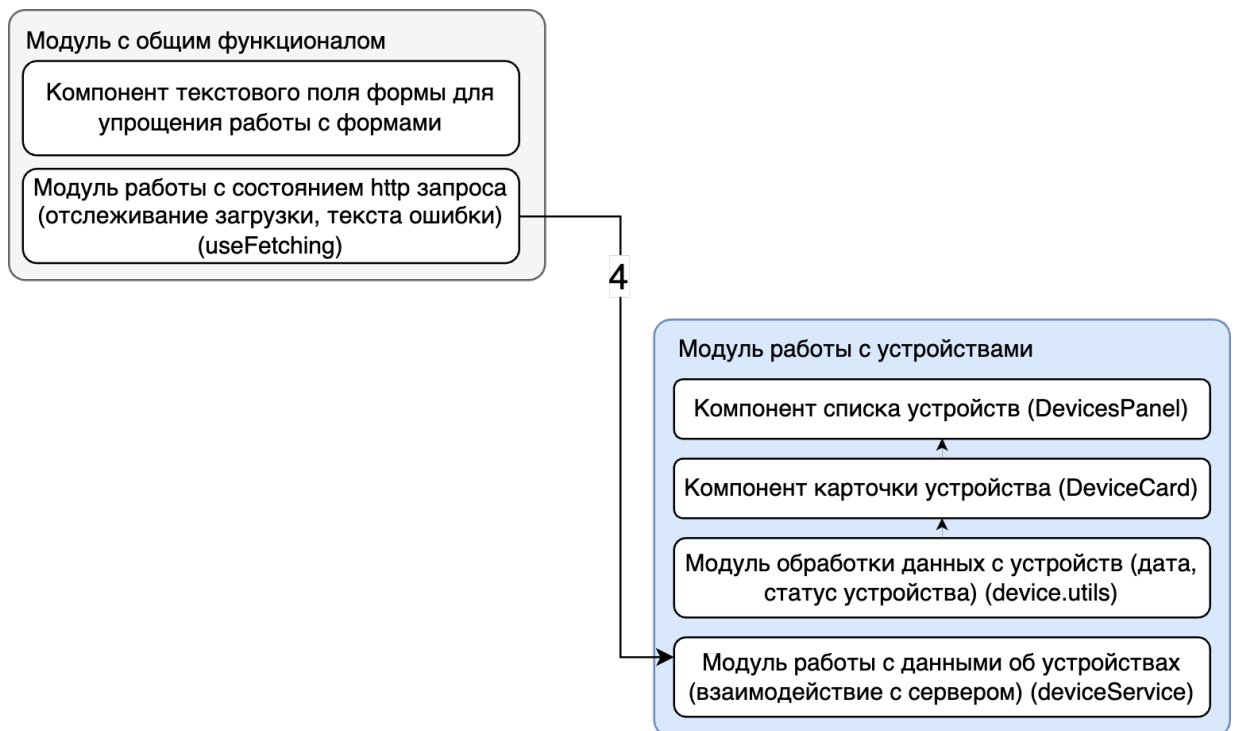
В компоненте LoginPage используется флаг isAuth из Хранилища с данными авторизации (authStore), чтобы отобразить форму входа или сообщение о том, что пользователь уже авторизован.

### 3. Хранилище авторизации и Модуль обеспечения разделения доступа



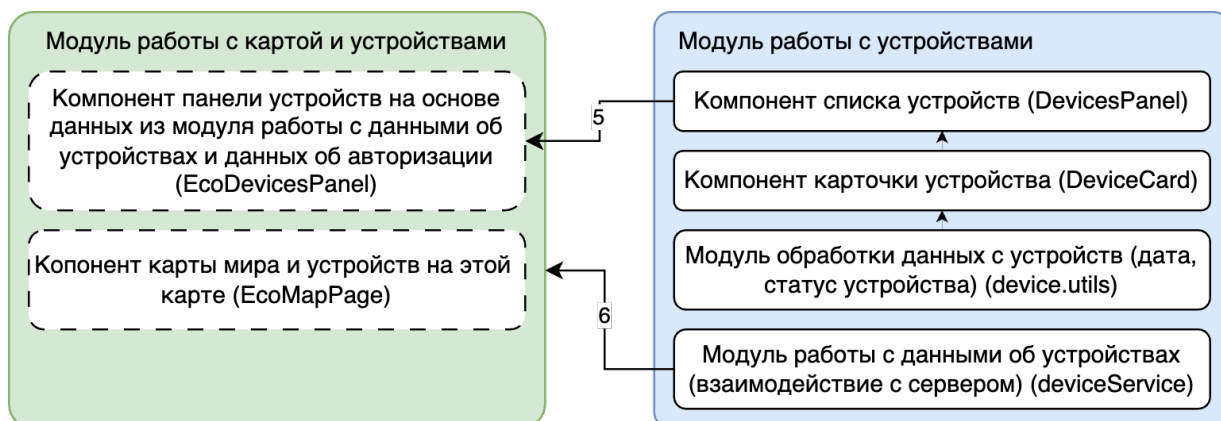
Интеграция не будет протестирована в связи с тем, что на данном этапе разработки системы Модуль обеспечения разделения доступа пока не используется.

### 4. Модуль работы с состоянием http запроса и Модуль работы с данными об устройствах.



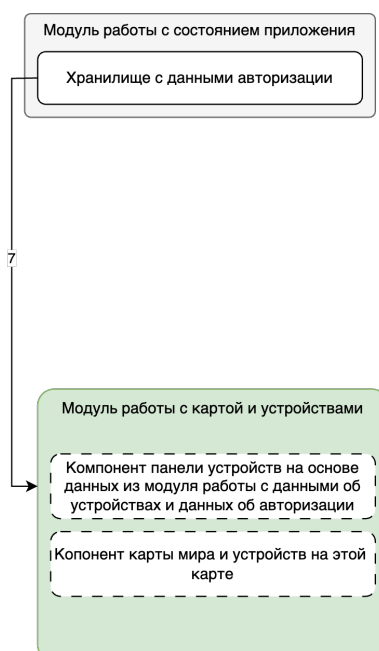
Модуль работы данными об устройствах использует функцию `useFetching(method)` из Модуля работы с http запросами, чтобы определить состояние (загрузка, ошибка) http запроса (передача метода `deviceService.getAllDevices` в функцию `useFetching` для получения нового объекта, который отражает статус запроса списка устройств).

5. Модуль работы с картой и устройствами и Компонент списка устройств.
6. Модуль работы с картой и устройствами и Модуль работы с данными об устройствах.



Модуль работы с картой и устройствами (компонент EcoMapPage) использует Компонент списка устройств DevicePanel для отображения доступных устройств на главной странице (передача списка устройств в компонент DevicePanel) и получения списка этих устройств через метод deviceService.getAllDevices() (модуль работы с данными об устройствах).

7. Хранилище с данными авторизации и Модуль работы с картой и устройствами.



Модуль работы с картой и устройствами (компонент EcoMapPage) использует флаг isAuth из Хранилища с данными авторизации, чтобы отображать кнопку "Войти" или "Выйти". Данная интеграция тестироваться не будет, так как маловероятно, что возникнут неполадки в ходе взаимодействия этих модулей.

## 1.4 Стратегия аттестационного тестирования

В ходе аттестационного тестирования тестируется функциональность объекта тестирования. Для этого определяются основные сценарии использования приложения, которые позволяют произвести необходимые проверки. Тестирование приложения происходит в браузере через интернет на тестовом сервере. Если в ходе выполнения какого-либо сценария выявлено, что объект тестирования не соответствует функциональности, тест считается не пройденным.

Будут протестированы следующие функции:

1. Авторизация в приложении
2. Выход из системы
3. Просмотр списка доступных устройств, и данных с датчиков этих устройств
4. Просмотр списка доступных устройств на карте мира

## 1.5 Стратегия тестирования производительности

В ходе тестирования производительности будет протестирован модуль работы с устройствами. Основной фокус будет сделан на анализ производительности рендера (отрисовки) интерфейса при наличии списка карточек устройств.

Ожидаемый результат скорости рендера для всех случаев: не более 16.7мс для отрисовки 1 кадра (не менее 60 кадров в секунду). Данные на которых будет производиться тест производительности:

1. Список из 0 объектов устройств в компоненте DevicesPanel.
2. Список из 100 объектов устройств в компоненте DevicesPanel.
3. Список из 400 объектов устройств в компоненте DevicesPanel.
4. Список из 800 объектов устройств в компоненте DevicesPanel.
5. Список из 1000 объектов устройств в компоненте DevicesPanel.
6. Список из 2000 объектов устройств в компоненте DevicesPanel.
7. Список из 3500 объектов устройств в компоненте DevicesPanel.
8. Список из 5000 объектов устройств в компоненте DevicesPanel.

## **1.6 Условия начала, окончания и перехода между этапами тестирования.**

1. Этап тестирования считается завершенным, если пройдены все заявленные тесты и получены результаты.
2. Можно начать следующий этап тестирования, только если предыдущий этап полностью завершен.
3. Тест считается успешно пройденным, если полученный результат совпадает с ожидаемым результатом.
4. Если полученный результат не совпадает с ожидаемым, следует проверить правильность теста. Если тест написан правильно, следует зафиксировать ошибку.

## **1.7 Условия возобновления и приостановки выполнения тестов.**

1. Тестирование может быть приостановлено только в случае ошибки, блокирующей выполнение тестирования.
2. Возобновление процедуры тестирования происходит после устранения ошибки, блокирующей выполнение тестирования.

## 2 Детальный план тестирования

### 2.1 Блочное тестирование

№ теста	Б1
Цель теста	Проверить метод по работе с local storage. Метод должен получать строку (токен) из local storage.
Тип теста	Позитивный
Объект тестирования	Метод модуля authService getAccessToken()
Входные параметры	-
Косвенные данные	Ключ по которому хранится token: 'есо-access-token'. Сам token: 'token-example'
Ожидаемый результат	Строка token, находящийся по ключу 'есо-access-token' в local storage

№ теста	Б2
Цель теста	Проверить корректность метода по работе с local storage. Метод должен получать строку (токен) из local storage.
Тип теста	Негативный
Объект тестирования	Метод модуля authService getAccessToken()
Входные параметры	-
Косвенные данные	Ключ по которому хранится token: 'есо-access-token'. По данному ключу нет данных в local storage
Ожидаемый результат	null

№ теста	Б3
Цель теста	Проверить метод по работе с local storage. Метод должен сохранять строку (токен) в local storage.
Тип теста	Позитивный
Объект тестирования	Метод модуля authService saveAccessToken()
Входные параметры	token - строка
Косвенные данные	Ключ по которому хранится token: 'eco-access-token'
Ожидаемый результат	строка token находится в local storage по ключу: 'eco-access-token'

№ теста	Б4
Цель теста	Проверить метод по работе с local storage. Метод должен сохранять строку (токен) в local storage.
Тип теста	Негативный
Объект тестирования	Метод модуля authService saveAccessToken()
Входные параметры	token - строка
Косвенные данные	Ключ по которому хранится token: 'eco-access-token'. Новый токен: 'new-token'. По этому ключу уже записана строка: 'old-token'.
Ожидаемый результат	token 'new-token' находится в local storage по ключу: 'eco-access-token', перезаписав существующую



№ теста	Б5
Цель теста	Проверить метод по работе с local storage. Метод должен удалять строку в local storage.
Тип теста	Позитивный
Объект тестирования	Метод модуля authService clearAccessToken()
Входные параметры	-
Косвенные данные	По ключу 'eco-access-token' хранится token 'token'
Ожидаемый результат	undefined в local storage по ключу: 'eco-access-token'

№ теста	Б6
Цель теста	Проверить, что карточка устройства отображает его имя и заданную дату в формате dd.ММ.yyyy hh:mm
Тип теста	Позитивный
Объект тестирования	Компонент DeviceCard
Входные параметры	{ name: 'device-test'; } - объект устройства, date - JavaScript дата для отображения (1 сентября 2023 года, 13:00)
Ожидаемый результат	Компонент DeviceCard, в котором находится текст: 'device-test' и текст даты: 01.09.2023 13:00

№ теста	Б7
Цель теста	Проверить, что дата карточка устройства отрисовывается, даже если дата не задана
Тип теста	Негативный
Объект тестирования	Компонент DeviceCard
Входные параметры	{ name: 'device-test'; } - объект устройства, date - null
Ожидаемый результат	Компонент DeviceCard, в котором находится текст: 'device-test'.

№ теста	Б8
Цель теста	Проверить, что карточка устройства отображает статус "Не в сети" если устройство не в сети более 600 минут.
Тип теста	Позитивный
Объект тестирования	Компонент DeviceCard
Входные параметры	{ name: 'device', time_not_online: 601 } - объект устройства
Ожидаемый результат	Компонент DeviceCard, в котором находится текст "Не в сети"

№ теста	Б9
Цель теста	Проверить, что карточка устройства отображает статус "В сети" если устройство не в сети менее 600 минут.
Тип теста	Позитивный
Объект тестирования	Компонент DeviceCard
Входные параметры	{ name: 'device', time_not_online: 599 } - объект устройства
Ожидаемый результат	Компонент DeviceCard, в котором находится текст "В сети"

№ теста	Б10
Цель теста	Проверить, что отрисовывается панель с карточками
Тип теста	Позитивный
Объект тестирования	Компонент DevicesPanel
Входные параметры	[[ { name: 'device1' }, { name: 'device2' }, { name: 'device3' } ]] - массив объектов устройств
Ожидаемый результат	Компонент DevicesPanel, в котором находится 3 компонента DeviceCard, с соответствующими названиями внутри карточек: 'device1', 'device2', 'device3'

№ теста	Б11
Цель теста	Проверить, что не отрисовываются карточки устройств, если не заданы устройства. Отображается текст "Нет устройств".
Тип теста	Негативный
Объект тестирования	Компонент DevicesPanel
Входные параметры	[] - пустой массив объектов устройств
Ожидаемый результат	Компонент DevicesPanel, в котором находится текст "Нет устройств".

№ теста	Б12
Цель теста	Проверить функцию получение даты и времени момента, когда были получены последние данные, учитывая часовой пояс устройства.
Тип теста	Позитивный
Объект тестирования	Функция <code>getDeviceLastDataDate()</code>
Входные параметры	'21.09.2023 13:00' - дата в формате строки "dd.ММ.yyyy'.', 5 - часовой пояс устройства в виде числа
Ожидаемый результат	Объект JavaScript Date с датой, метод <code>toISOString()</code> которого возвращает строку '2023-09-21T08:00:00.000Z'.

## 2.2 Интеграционное тестирование

№ теста	И1
Цель теста	Проверить, что если пользователь авторизован, то при переходе на страницу авторизации LoginPage отображается текст "Вы уже авторизованы" и ссылка "Перейти к приложению".
Тип теста	Позитивный
Объекты тестирования	Хранилище данных авторизации, компонент LoginPage
Входные данные	Установить в хранилище авторизации флаг <code>isAuth = true</code> .
Последовательность действий	<ol style="list-style-type: none"><li>1. Создать "заглушку" для <code>authStore</code> с помощью метода <code>jest.mock(...)</code> со значением <code>isAuth: true</code>.</li><li>2. Отрендерить компонент с помощью метода <code>render(&lt;LoginPage/&gt;)</code>.</li><li>3. С помощью метода <code>screen.getBy(...)</code> проверить, что в документе находится текст "Вы уже авторизованы" и ссылка "Перейти к приложению".</li></ol>
Ожидаемый результат	Компонент <code>LoginPage</code> , в котором текст "Вы уже авторизованы" и ссылка, в которой текст "Перейти к приложению".

№ теста	И2
Цель теста	Проверить, что если пользователь не авторизован, то при переходе на страницу авторизации LoginPage отображается текст "Войти в систему" и кнопка "Войти".
Тип теста	Позитивный
Объекты тестирования	Хранилище данных авторизации, компонент LoginPage
Входные данные	Установить в хранилище авторизации флаг isAuth = false.
Последовательность действий	<ol style="list-style-type: none"> <li>1. Создать "заглушку" для authStore с помощью метода jest.mock(...) со значением isAuth: false.</li> <li>2. Отрендерить компонент с помощью метода render(&lt;LoginPage/&gt;).</li> <li>3. С помощью метода screen.getBy(...) проверить, что в документе находится текст "Войти в систему" и кнопка с текстом "Войти".</li> </ol>
Ожидаемый результат	Компонент LoginPage, в котором текст "Войти в систему" и кнопка, в которой текст "Войти".

№ теста	ИЗ
Цель теста	Проверить, что на главной странице отображаются устройства полученные из объекта работы с устройствами deviceService.
Тип теста	Позитивный
Объекты тестирования	Метод deviceService.getAllDevices(), компонент EcoMapPage
Входные данные	Установить "заглушку" для метода deviceService.getAllDevices, которая возвращает - [{ name: 'device1' }, { name: 'device2' }, { name: 'device3' }]
Последовательность действий	<ol style="list-style-type: none"> <li>1. Создать заглушку для метода deviceService.getAllDevices с помощью метода jest.spyOn(...).</li> <li>2. Для созданной заглушки установить возвращаемое значение с помощью метода mockResolvedValue(...)</li> <li>3. Отрендерить компонент с помощью функции render(&lt;EcoMapPage /&gt;).</li> <li>4. С помощью метода screen.getBy(...) для каждого устройства в списке проверить, что имя этого устройства отрисовано в документе.</li> </ol>
Ожидаемый результат	Компонент EcoMapPage, в котором компоненты DeviceCard, в каждой карточек информация о соответствующем устройстве (текст из поля name объекта устройства), полученном через метод "заглушку" deviceService.getAllDevices()

№ теста	И4
Цель теста	Проверить, что во время выполнения метода запроса устройств (возвращает JavaScript объект Promise), модуль работы с http запросами устанавливает значение loading: true, а после выполнения метода loading: false.
Тип теста	Позитивный
Объекты тестирования	Метод deviceService.getAllDevices(), функция useFetching(method).
Последовательность действий	<ol style="list-style-type: none"> <li>1. Создать "заглушку" для метода deviceService.getAllDevices с помощью jest.spyOn(...) и метода mockResolvedValue(new Promise(...)) (объект Promise, который возвращает значение через 200 мс) .</li> <li>2. Вызвать функцию useFetching(deviceService.getAllDevices) для метода. Функция возвращает массив [fetch, loading].</li> <li>3. Проверить, что значение loading = false</li> <li>4. Вызвать функцию fetch(...) и проверить, что значение loading = true.</li> <li>5. С помощью метода jest.advanceTimersByTime(200) подождать 200 мс</li> <li>6. Проверить, что loading = false</li> </ol>
Ожидаемый результат	loading = true во время загрузки. loading = false после окончания загрузки.



№ теста	И5
Цель теста	Проверить, что при нажатии на кнопку "Выйти" пользователь становится неавторизованным.
Тип теста	Позитивный
Объекты тестирования	Хранилище с данными авторизации, компонент EcoMapPage.
Входные данные	Установить значение флага в Хранилище авторизации isAuth = true. Ключ, по которому хранится токен: 'eco-access-token'. В local storage хранится значение 'test-token' по ключу 'eco-access-token'
Последовательность действий	<ol style="list-style-type: none"> <li>1. Установить значение 'test-token' по ключу 'eco-access-token' с помощью метода localStorage.setItem</li> <li>2. Создать "заглушку" для authStore с помощью метода jest.mock(...) со значением isAuth: true.</li> <li>3. Отрендерить компонент с помощью функции render(&lt;LoginPage /&gt;)</li> <li>3. С помощью метода fireEvent.click(screen.getByText('Выйти')) смитировать клик пользователя по кнопке "Выйти"</li> <li>4. Проверить, что значение isAuth = false в authStore</li> <li>5. Проверить, что с помощью метода localStorage.getItem(...) по ключу 'eco-access-token' получаем значение undefined</li> </ol>
Ожидаемый результат	По нажатию на кнопку "Выйти" в компоненте EcoMapPage: флаг isAuth в хранилище с данными авторизации равен false, нет значения (undefined) в local storage по ключу 'eco-access-token'.

## 2.3 Аттестационное тестирование

Данные для аттестационного тестирования:

- Данные для входа

1. Логин: test-user
2. Пароль: test-user-password-2023

- Список устройств для пользователя test-user:

1. { latitude: 61.906633, longitude: 34.192919, name: 'Агро', time\_not\_online: 30 }
2. { latitude: 61.7324, longitude: 34.50461, name: 'Город', time\_not\_online: 30 }
3. { latitude: 59.730036, longitude: 30.38743, name: 'Sokol-M', time\_not\_online: 30 }

№ теста	A1
Имя функции	Авторизация в системе
Тип теста	Позитивный
Цель теста	Проверить, что при вводе существующих логина и пароля, пользователь будет авторизован в системе и попадет на главную страницу.
Входные данные	Существующие в базе данных логин и пароль
Сценарий	Пользователь неавторизован, пользователь заходит на страницу LoginPage вводит существующие логин и пароль для существующего пользователя в базе данных, нажимает кнопку Войти
Ожидаемый результат	Пользователь авторизован и перенаправлен на главную страницу с устройствами. При заходе на страницу входа в систему отображается текст "Вы уже авторизованы".

№ теста	A2
Имя функции	Авторизация в системе
Тип теста	Негативный
Цель теста	Проверить, что при вводе не существующих в базе данных логина и пароля, пользователь не будет авторизован в системе и увидит ошибку "Неверный логин или пароль".
Входные данные	Не существующие в базе данных логин и пароль
Сценарий	Пользователь неавторизован, пользователь заходит на страницу LoginPage вводит не существующие в базе данных логин и пароль, нажимает кнопку Войти
Ожидаемый результат	Пользователь остается на странице LoginPage. На форме логина ошибка "Неверный логин или пароль".

№ теста	А3
Имя функции	Выйти из системы
Тип теста	Позитивный
Цель теста	Проверить, что если пользователь авторизован, то при нажатии на кнопку выход, пользователь становится неавторизованным.
Входные данные	Существующий и валидный (еще не истек) токен доступа в local storage, хранящийся по ключу 'есо-access-token'
Сценарий	Пользователь авторизован, пользователь заходит на главную страницу с устройствами, нажимает кнопку Выйти
Ожидаемый результат	Пользователь неавторизован, вместо кнопки Выйти показывается кнопка Войти. При заходе на страницу входа в систему отображается форма входа в систему.

№ теста	A4
Имя функции	Посмотреть список доступных устройств и данных с датчиков на этих устройствах
Тип теста	Позитивный
Цель теста	Проверить, что если пользователь авторизован и заходит на главную страницу, то отображается список карточек доступных устройств с последними данными с датчиков на них
Входные данные	Существующий и валидный (еще не истек) токен доступа в local storage, хранящийся по ключу 'есо-access-token'
Сценарий	Пользователь авторизован, пользователь заходит на главную страницу
Ожидаемый результат	Пользователь видит список всех устройств, на карточках устройств отображены показания с датчиков на этих устройствах.

№ теста	A5
Имя функции	Посмотреть список доступных устройств и данных с датчиков на этих устройствах
Тип теста	Негативный
Цель теста	Проверить, что если пользователь не авторизован и заходит на главную страницу, то отображается пустой список карточек устройств
Входные данные	-
Сценарий	Пользователь не авторизован, пользователь заходит на главную страницу
Ожидаемый результат	Пользователь видит пустой список устройств

№ теста	A6
Имя функции	Посмотреть список доступных устройств на карте мира
Тип теста	Позитивный
Цель теста	Проверить, что если пользователь авторизован и заходит на главную страницу, то отображается карта мира и значки датчиков на этой карте в соответствии с координатами этих устройств
Входные данные	Существующий и валидный (еще не истек) токен доступа в local storage, хранящийся по ключу 'eco-access-token'
Сценарий	Пользователь авторизован, пользователь заходит на главную страницу
Ожидаемый результат	Пользователь видит список всех устройств на карте мира в соответствии с их координатами (те же самые устройства и в списке карточек).

№ теста	A7
Имя функции	Посмотреть список доступных устройств на карте мира
Тип теста	Негативный
Цель теста	Проверить, что если пользователь не авторизован и заходит на главную страницу, то на карте мира не отображается ни 1 устройство
Входные данные	-
Сценарий	Пользователь не авторизован, пользователь заходит на главную страницу
Ожидаемый результат	Пользователь видит карту мира, на карте нет ни 1 устройства

## 2.4 Тестирование производительности

№ теста	C1
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для пользовательского интерфейса компонента DevicesPanel с 0 устройствами
Входные данные	Список устройств с длиной 0
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс

№ теста	C2
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для компонента DevicesPanel с 100 устройствами
Входные данные	Сгенерированный список устройств с длиной 100 - [{ name: device_1 }, { name: device_2 }, ...].
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс



№ теста	СЗ
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для компонента DevicesPanel с 100 устройствами
Входные данные	Сгенерированный список устройств с длиной 400 - [{ name: device_1 }, { name: device_2 }, ...].
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс

№ теста	C4
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для компонента DevicesPanel с 100 устройствами
Входные данные	Сгенерированный список устройств с длиной 800 - [{ name: device_1 }, { name: device_2 }, ...].
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс

№ теста	C5
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для компонента DevicesPanel с 1000 устройствами
Входные данные	Сгенерированный список устройств с длиной 1000 - [{ name: device_1 }, { name: device_2 }, ...].
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс

№ теста	С6
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для компонента DevicesPanel с 1000 устройствами
Входные данные	Сгенерированный список устройств с длиной 2000 - [{ name: device_1 }, { name: device_2 }, ...].
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс

№ теста	C7
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для компонента DevicesPanel с 3500 устройствами
Входные данные	Сгенерированный список устройств с длиной 1000 - [{ name: device_1 }, { name: device_2 }, ...].
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс

№ теста	C8
Цель теста	Проверить, что скорость отрисовки кадра не более 16.7мс для компонента DevicesPanel с 5000 устройствами
Входные данные	Сгенерированный список устройств с длиной 10000 - [{ name: device_1 }, { name: device_2 }, ...].
Сценарий	Запускается запись информации о процессе отрисовки в инструментах разработчика в браузере (Chrome DevTools). Пользователь заходит на страницу с компонентом DevicesPanel. Затем нажимает кнопку скрыть панель с устройствами. Затем нажимает кнопку раскрыть панель с устройствами. Запись останавливается.
Ожидаемый результат	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7мс

### 3 Примеры тестов

```
describe('authService.getAccessToken()', () => {
  beforeEach(() => {
    localStorage.clear();
  });

  test('возвращает токен из local storage', () => {
    const expectedToken = 'token-example';

    localStorage.setItem('eco-access-token', expectedToken);
    const result = authService.getSavedAccessToken();

    expect(result).toBe(expectedToken);
  });
});
```

Рис. 3: Пример блочного теста

```
jest.mock('./auth.store.ts', () => ({
  useAuthStore: createMockAuthStore(),
}));

describe('LoginPage когда пользователь уже авторизован', () => {
  test('Отображается "Вы уже авторизованы" текст и ссылка "Перейти к приложению"', () => {
    useAuthStore.mockReturnValue({ isAuth: true, login: jest.fn(), logout: jest.fn() });

    render(<LoginPage />);

    const authText = screen.getByText('Вы уже авторизованы');
    expect(authText).toBeInTheDocument();

    const linkToApp = screen.getByText('Перейти к приложению');
    expect(linkToApp).toBeInTheDocument();
  });
});
```

Рис. 4: Пример интеграционного теста

### 4 Журнал тестирования

Тест	Дата	Тестирующий	Результат
Б1	01.12.2023	Жидов Н.И.	Пройден
Б2	01.12.2023	Жидов Н.И.	Пройден
Б3	01.12.2023	Жидов Н.И.	Пройден
Б4	01.12.2023	Жидов Н.И.	Пройден
Б5	01.12.2023	Жидов Н.И.	Пройден
Б6	01.12.2023	Жидов Н.И.	Ошибка №1
Б7	01.12.2023	Жидов Н.И.	Пройден
Б8	01.12.2023	Жидов Н.И.	Пройден
Б9	01.12.2023	Жидов Н.И.	Пройден
Б10	01.12.2023	Жидов Н.И.	Пройден
Б11	01.12.2023	Жидов Н.И.	Ошибка №2
Б12	01.12.2023	Жидов Н.И.	Пройден
И1	01.12.2023	Жидов Н.И.	Ошибка №3
И2	01.12.2023	Жидов Н.И.	Пройден
И3	01.12.2023	Жидов Н.И.	Пройден
И4	01.12.2023	Жидов Н.И.	Пройден
И5	01.12.2023	Жидов Н.И.	Пройден
А1	01.12.2023	Жидов Н.И.	Пройден
А2	01.12.2023	Жидов Н.И.	Пройден
А3	01.12.2023	Жидов Н.И.	Пройден
А4	01.12.2023	Жидов Н.И.	Пройден
А5	01.12.2023	Жидов Н.И.	Пройден
А6	01.12.2023	Жидов Н.И.	Пройден
А7	01.12.2023	Жидов Н.И.	Пройден
С1	01.12.2023	Жидов Н.И.	Пройден
С2	01.12.2023	Жидов Н.И.	Пройден
С3	01.12.2023	Жидов Н.И.	Пройден
С4	01.12.2023	Жидов Н.И.	Пройден
С5	01.12.2023	Жидов Н.И.	Пройден
С6	01.12.2023	Жидов Н.И.	Пройден
С7	01.12.2023	Жидов Н.И.	Пройден
С8	01.12.2023	Жидов Н.И.	Ошибка №4



## 5 Журнал ошибок

Номер	Тест	Полученный результат	Ожидаемый результат
1	Б6	Дата в компоненте карточки DeviceCard в формате: ММ/dd/уууу hh:mm	Дата в формате: dd.ММ.уууу hh:mm
2	Б11	Нет надписи "Нет устройств"	Надпись "Нет устройств" в компоненте DevicesPanel, когда пустой список устройств
3	И1	В компоненте LoginPage нет ссылки с надписью "Перейти к приложению". Только текст "Вы уже авторизованы"	Компонент LoginPage, в котором текст "Вы уже авторизованы" и ссылка, в которой текст "Перейти к приложению".
4	С8	В отчете в инструментах разработчика скорость отрисовки нескольких кадров периодически превосходит 30 мс	В отчете в инструментах разработчика скорость отрисовки 1 кадра не более 16.7м

## 6 Оценка покрытия кода тестами

Общее покрытие кода составляет 84% согласно автоматическому подсчёту средствами библиотеки для тестирования Jest.

## 7 Заключение

Было проведено 32 теста, из них 12 блочных, 5 интеграционных, 7 аттестационных и 8 тестов производительности. За время тестирования выявлено 4 ошибки. Ошибки №1, №3 являются критическими, их необходимо исправить в кратчайшие сроки. Остальные ошибки на данном этапе разработки не являются критическими для работы системы.