

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий

09.03.04 – Программная инженерия

Отчет по дисциплине "Верификация ПО"
ТЕСТИРОВАНИЕ ИНТЕРФЕЙСА МОБИЛЬНОГО ПРИЛОЖЕНИЯ
СИСТЕМЫ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА

Выполнил студент группы 22407:

Беленков Николай Дмитриевич

Преподаватель:

К. А. Кулаков, к.ф.-м.н., доцент

Оценка _____

Дата _____

Петрозаводск 2023 г.

Содержание

1	Объект тестирования	3
2	Стратегия тестирования	4
2.1	Архитектура мобильного приложения	4
2.2	Стратегия блочного тестирования	7
2.2.1	unit-тесты	7
2.2.2	widget-тесты	7
2.3	Стратегия интеграционного тестирования	7
2.4	Стратегия аттестационного тестирования	8
2.5	Стратегия специального тестирования	8
2.6	Условия начала, окончания и перехода между этапами тестирования	9
2.7	Условия возобновления и приостановки выполнения тестов	10
3	Детальный план тестов	10
3.1	Блочное тестирование	10
3.2	Тестирование виджетов	16
3.3	Интеграционное тестирование	18
3.4	Аттестационное тестирование	23
3.5	Специальное тестирование	27
3.6	Примеры тестов	39
4	Журнал тестирования	41
4.1	Блочное тестирование	41
4.2	Тестирование виджетов	41
4.3	Интеграционное тестирование	42
4.4	Аттестационное тестирование	42
4.5	Специальное тестирование	43
5	Журнал найденных ошибок	44
6	Оценка покрытия кода тестами	48
7	Результаты	48

1 Объект тестирования

Информационная система экологического мониторинга представляет собой мобильное приложение, разработанное с использованием современных технологий, в первую очередь фреймворк Flutter для языка Dart. Основной целью этой системы является сбор и визуализация данных, связанных с экологическим мониторингом, что позволяет пользователям контролировать 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.

Функциональные возможности приложения:

- Авторизация пользователя
- Выход пользователя из системы
- Просмотр карты
- Отображение доступных устройств на карте
- Просмотр краткой информации о выбранном устройстве

2 Стратегия тестирования

2.1 Архитектура мобильного приложения

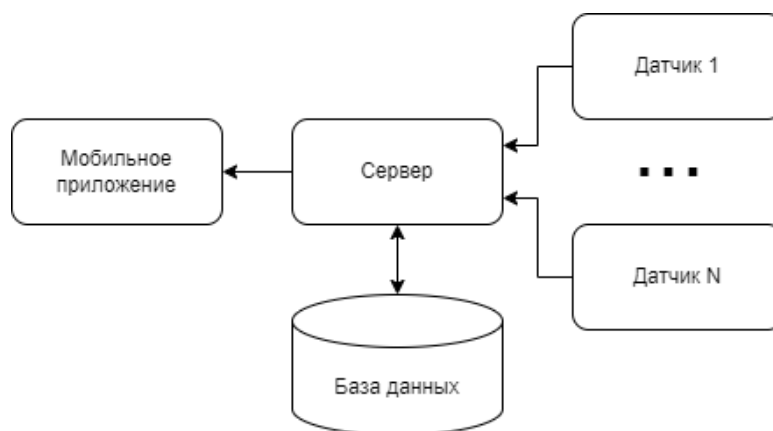


Рис. 1 – Архитектура системы в целом

В рамках данного отчета будет проведено только тестирование самого мобильного приложения ввиду того, что остальные части системы находятся вне контроля автора отчета.



Рис. 2 – Архитектура мобильного приложения

Основные модули мобильного приложения:

1. Экран авторизации(классы LoginPage и _LoginPageState). Данный модуль отвечает за отображение формы авторизации и поддержку взаимодействия пользователя с ней. Содержит следующие методы:

- initState() - создание контроллера анимации
- dispose() - удаление контроллера
- build(BuildContext context) - отображение страницы

Содержит следующие поля:

- late AnimationController controller
- final TextEditingController _emailController
- final TextEditingController _passController
- LoginHelper loginHelper

2. Модуль работы с токенами авторизации(класс LoginHelper). Данный модуль отвечает за получение токенов авторизации от сервера, а также валидацию пользовательского ввода. Содержит следующие методы:

- checkPassword(String user, String pass, http.Client client) - делает запрос к серверу для авторизации пользователя
- parseLoginJson(Map<String, dynamic> json) - разбор ответа сервера
- getToken - геттер для токена авторизации

Содержит следующие поля:

- TokenHelper tokenHelper

3. Хранилище с данными авторизации(класс tokenHelper). Данный модуль отвечает за сохранение токенов авторизации, проверку их наличия и получение их из памяти устройства. Содержит следующие методы:

- setToken(String newToken) - устанавливает значение токена авторизации
- getTokenFromPrefs() - получает значение токена авторизации
- removeToken() - удаляет токен авторизации

Содержит следующие поля:

- bool _isTokenPresent
- bool _isLoading
- String? token

4. Модуль работы с данными от устройствах(класс SensorHelper). Данный модуль отвечает за получение данных устройств от сервера и их хранение. Содержит следующие методы:

- `getSensorsJson(http.Client client)` - делает запрос к серверу для получения списка устройств

Содержит следующие поля:

- `bool _isReady`
- `Sensor? currentSensor`
- `TokenHelper tokenHelper`
- `SensorParser sensorParser`

5. Модуль обработки данных устройств(класс `SensorParser`). Данный модуль отвечает за обработку данных устройств, полученных от сервера. Содержит следующие методы:

- `parseSensorsJson(Map<String, dynamic> json)` - разбор файла со списком устройств, полученного от сервера

Содержит следующие поля:

- `List<Sensor> sensorArray`

6. Экран работы с картой(классы `MapPage` и `_MapPageState`). Данный модуль отвечает за отображение карты, а также устройств на ней, а также обработку действий пользователя по взаимодействию с ними. Содержит следующие методы:

- `initState()` - создание контроллера анимации
- `dispose()` - удаление контроллера
- `build(BuildContext context)` - отображение страницы

Содержит следующие поля:

- `late AnimationController controller`
- `SensorHelper sensorHelper`

Функциональные возможности приложения:

- Авторизация пользователя
- Выход пользователя из системы

- Просмотр карты
- Отображение доступных устройств на карте
- Просмотр краткой информации о выбранном устройстве

2.2 Стратегия блочного тестирования

Блочные тесты проверяют работоспособность отдельных функций программы. Их поведение не зависит от результатов работы других тестируемых функций. Для их проверки в качестве аргументов при вызове функции передаются различные значения, после чего возвращаемый результат функции проверяется на наличие ошибок.

2.2.1 unit-тесты

Тестируются отдельные методы или классы. Проверяется корректность выполнения при различных входных данных. Внешние зависимости классов заменяются mock-ами (Например все http-серверы заменены mock-ами, ответ сервера - stub). Тестирование проводится с помощью библиотеки `flutter_test`.

2.2.2 widget-тесты

Тестируются отдельные виджеты. Проверяется, что виджет выглядит и ведет себя так, как запланировано (появляется на экране, на виджет можно нажать и т.п.). Для тестирования виджетов необходимо среду, в которой виджет будет существовать. Тестирование проводится с помощью библиотеки `flutter_test`.

2.3 Стратегия интеграционного тестирования

Интеграционные тесты проверяют работоспособность целого приложения или больших его частей (объединяющих несколько модулей), правильность их работы друг с другом.

Интеграционное тестирование проводится на реальном устройстве/эмуляторе при помощи библиотеки `integration_test`.

Интеграционное тестирование проводится в несколько этапов (см рис.3):

1. `_LoginPageState` и `LoginHelper` - `LoginHelper` передает в `_LoginPageState` статус текущего токена аутентификации, а также получает от него запрос на удаление текущего токена

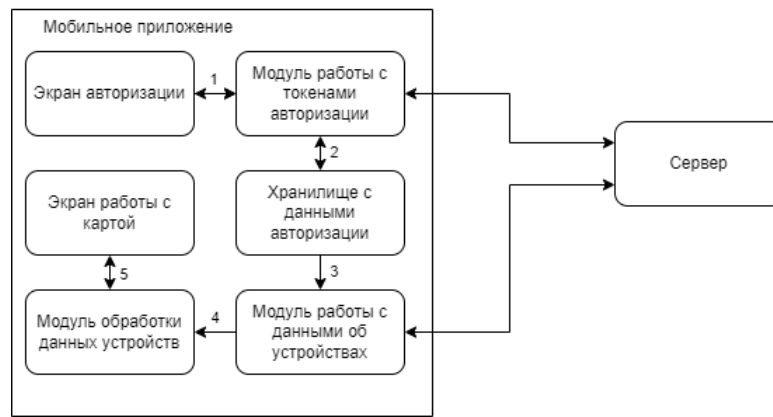


Рис. 3 – Стратегия интеграционного тестирования

2. LoginHelper и TokenHelper - LoginHelper передает в TokenHelper токен аутентификации для сохранения, а также получает от него текущий токен и передает запрос на удаление текущего токена
3. SensorHelper и TokenHelper - TokenHelper передает в SensorHelper токен аутентификации
4. SensorParser и SensorHelper - SensorHelper передает в SensorParser JSON-документ, представленный в виде Map<String, dynamic>
5. _MapPageState и SensorParser - SensorParser передает в _MapPageState список найденных устройств для отображения на карте

2.4 Стратегия аттестационного тестирования

В ходе аттестационного тестирования будет протестирована работоспособность приложения и его возможность осуществлять заявленный функционал.

Аттестационное тестирование будет проводиться методом «живого человека». В роли такого человека выступает сам автор отчета. Тестировщик, по заранее заданным инструкциям (TestCases), производит требуемые действия и сверяется с заранее заданными результатами.

2.5 Стратегия специального тестирования

В ходе специального тестирования будет протестирована возможность приложения выполнять заявленный функционал при различных условиях доступа к мобильной сети, а также при различном уровне загрузки сервера.

Аттестационное тестирование будет проводиться методом «живого человека». В роли такого человека выступает сам автор отчета. Тестировщик, по заранее заданным инструкциям (TestCases), производит требуемые действия и сверяется с заранее заданными результатами.

Тесты запускаются на эмуляторе устройства в рамках AndroidStudio. Симуляция различных условий доступа к мобильной сети производится при помощи настроек эмулятора («Network type» и «Signal strength»). Сервер подменен на локальную, копию, реализованную на Python и Flask, загруженность сервера эмулируется различным уровнем задержки перед ответом на запрос.

В таблице ниже представлено соответствие тестов и их условий:

Загрузка сервера	Доступ к сети			
	LTE, отличный	3G, отличный	3G, плохой	Нет доступа к сети
Сервер не нагружен	C1	C4	C7	C10
Сервер умеренно нагружен	C2	C5	C8	C11
Сервер сильно нагружен	C3	C6	C9	C12

Эмуляция нагрузки на сервер:

- Сервер не нагружен - сервер сразу возвращает ответ после получения запроса
- Сервер умеренно нагружен - сервер возвращает ответ через 2 секунды после получения запроса
- Сервер сильно нагружен - сервер возвращает ответ через 5 секунд после получения запроса

2.6 Условия начала, окончания и перехода между этапами тестирования

Тест считается пройденным если полученный и ожидаемый результат совпадают. Тестирование считается пройденным если во время его прохождения не выявлено критических ошибок и процент пройденных тестов не менее 85%.

2.7 Условия возобновления и приостановки выполнения тестов

Тестирование должно быть остановлено если количество непройденных тестов больше 30% от общего количества, а также при обнаружении критических ошибок сильно влияющих на функциональность приложения.

3 Детальный план тестов

3.1 Блочное тестирование

№ теста	Б1
Тип теста	Позитивный
Описание теста	Проверка корректности отправки запроса аутентификации к серверу.
Объект тестирования	loginHelper : метод checkPassword(String user, String pass, http.Client client)
Входные данные	String user - имя пользователя, String pass - пароль, MockitoClient client - mock сервера
Косвенные входные данные	Сервер готов к работе
Ожидаемый результат	Запрос выполнен, метод вернул значение True

№ теста	Б2
Тип теста	Негативный
Описание теста	Проверка корректности отправки запроса аутентификации к серверу.
Объект тестирования	loginHelper : метод checkPassword(String user, String pass, http.Client client)
Входные данные	String user - имя пользователя, String pass - пароль, MockitoClient client - mock сервера
Косвенные входные данные	Сервер не отвечает
Ожидаемый результат	Exception('http.post error: statusCode=500')

№ теста	Б3
Тип теста	Позитивный
Описание теста	Проверка корректности разбора ответа сервера (Аутентификация).
Объект тестирования	LoginHelper : метод parseLoginJson(Map<String, dynamic> json)
Входные данные	Map<String, dynamic> json - ответ сервера, result="ok"
Косвенные входные данные	-
Ожидаемый результат	Метод вернул значение True

№ теста	Б4
Тип теста	Позитивный
Описание теста	Проверка корректности разбора ответа сервера (Аутентификация).
Объект тестирования	LoginHelper : метод parseLoginJson(Map<String, dynamic> json)
Входные данные	Map<String, dynamic> json - ответ сервера, result != "ok"
Косвенные входные данные	-
Ожидаемый результат	Метод вернул значение False

№ теста	Б5
Тип теста	Позитивный
Описание теста	Проверка корректности сохранения токена авторизации.
Объект тестирования	TokenHelper : setToken(String newToken)
Входные данные	String newToken
Косвенные входные данные	-
Ожидаемый результат	В SharedPreferences есть строка по ключу "access_token"

№ теста	Б6
Тип теста	Позитивный
Описание теста	Проверка корректности удаления токена авторизации.
Объект тестирования	TokenHelper : removeToken()
Входные данные	-
Косвенные входные данные	В SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	В SharedPreferences отсутствует строка по ключу "access_token"

№ теста	Б7
Тип теста	Позитивный
Описание теста	Проверка корректности сохранения токена авторизации.
Объект тестирования	TokenHelper : removeToken()
Входные данные	-
Косвенные входные данные	В SharedPreferences отсутствует строка по ключу "access_token"
Ожидаемый результат	В SharedPreferences отсутствует строка по ключу "access_token"

№ теста	Б8
Тип теста	Позитивный
Описание теста	Проверка корректности сохранения токена авторизации.
Объект тестирования	TokenHelper : getTokenFromPrefs()
Входные данные	-
Косвенные входные данные	В SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	В поле token установлено текущее значение токена, поле _isTokenPresent = true

№ теста	Б9
Тип теста	Позитивный
Описание теста	Проверка корректности сохранения токена авторизации.
Объект тестирования	TokenHelper : getTokenFromPrefs()
Входные данные	-
Косвенные входные данные	В SharedPreferences отсутствует строка по ключу "access_token"
Ожидаемый результат	Поле token = null, поле _isTokenPresent = false

№ теста	Б10
Тип теста	Позитивный
Описание теста	Проверка корректности отправки запроса на получение данных устройств от сервера.
Объект тестирования	SensorHelper : метод getSensorsJson(http.Client client)
Входные данные	MockitoClient client - mock сервера
Косвенные входные данные	Сервер готов к работе, в SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	Запрос выполнен, код ответа - 200, Переменная _isReady данного класса установлена в значение True

№ теста	Б11
Тип теста	Негативный
Описание теста	Проверка корректности отправки запроса на получение данных устройств от сервера.
Объект тестирования	SensorHelper : метод getSensorsJson(http.Client client)
Входные данные	MockitoClient client - mock сервера
Косвенные входные данные	Сервер готов к работе, В SharedPreferences отсутствует строка по ключу "access_token"
Ожидаемый результат	Exception('no access_token found')

№ теста	Б12
Тип теста	Негативный
Описание теста	Проверка корректности отправки запроса на получение данных устройств от сервера.
Объект тестирования	SensorHelper : метод getSensorsJson(http.Client client)
Входные данные	- MockitoClient client - mock сервера
Косвенные входные данные	Сервер не отвечает, в SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	Exception('http.get error: statusCode=500')

№ теста	Б13
Тип теста	Позитивный
Описание теста	Проверка корректности разбора ответа сервера (Устройства).
Объект тестирования	SensorParser : parseSensorJson(Map<String, dynamic> json)
Входные данные	Map<String, dynamic> json - ответ сервера
Косвенные входные данные	-
Ожидаемый результат	Поле sensorArray заполнено данными устройств

3.2 Тестирование виджетов

№ теста	В1
Тип теста	Позитивный
Описание теста	Проверка корректности отображения экрана авторизации.
Объект тестирования	Экран авторизации
Входные данные	-
Косвенные входные данные	Система не знает, авторизован ли пользователь
Ожидаемый результат	На экране отображается текст «Проверяем, есть ли у нас токен», отображается спиннер загрузки

№ теста	В2
Тип теста	Позитивный
Описание теста	Проверка корректности отображения экрана авторизации.
Объект тестирования	Экран авторизации
Входные данные	-
Косвенные входные данные	Пользователь не авторизован
Ожидаемый результат	На экране отображаются поля ввода «Логин» и «Пароль», кнопка «Войти»

№ теста	В3
Тип теста	Позитивный
Описание теста	Проверка корректности отображения экрана авторизации.
Объект тестирования	Экран авторизации
Входные данные	-
Косвенные входные данные	Пользователь авторизован
Ожидаемый результат	На экране отображается текст «Вы уже авторизованы», кнопка «Выйти»

№ теста	В4
Тип теста	Позитивный
Описание теста	Проверка корректности отображения шторки навигации.
Объект тестирования	Шторка навигации
Входные данные	-
Косвенные входные данные	Нажатие на кнопку вызова шторки навигации
Ожидаемый результат	На экране отображается шторка навигации, в ней есть элементы «Профиль», «Карта» и «Настройки»

№ теста	В5
Тип теста	Позитивный
Описание теста	Проверка корректности отображения экрана карты.
Объект тестирования	Экран карты
Входные данные	-
Косвенные входные данные	Пользователь авторизован
Ожидаемый результат	На экране отображается карта, выдвигаемая панель

3.3 Интеграционное тестирование

№ теста	И1
Тип теста	Позитивный
Описание теста	Проверка интеграции классов LoginPageState и LoginHelper
Объект тестирования	Классы LoginPageState и LoginHelper
Входные данные	-
Косвенные входные данные	В SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	LoginPageState получает от LoginHelper токен авторизации, показывается экран с текстом «Вы уже авторизованы», кнопкой «Выйти»

№ теста	И2
Тип теста	Позитивный
Описание теста	Проверка интеграции классов LoginHelper и TokenHelper
Объект тестирования	Классы LoginHelper и TokenHelper
Входные данные	-
Косвенные входные данные	В SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	LoginHelper пытается получить от TokenHelper токен авторизации, LoginHelper.getToken возвращает значение токена

№ теста	И3
Тип теста	Позитивный
Описание теста	Проверка интеграции классов LoginHelper и TokenHelper
Объект тестирования	Классы LoginHelper и TokenHelper
Входные данные	-
Косвенные входные данные	В SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	LoginHelper отправляет TokenHelper запрос на удаление токена, LoginHelper.getToken возвращает null

№ теста	И4
Тип теста	Позитивный
Описание теста	Проверка интеграции классов LoginHelper и TokenHelper
Объект тестирования	Классы LoginHelper и TokenHelper
Входные данные	-
Косвенные входные данные	В SharedPreferences нет строки по ключу "access_token"
Ожидаемый результат	LoginHelper пытается получить от TokenHelper токен авторизации, LoginHelper.getToken возвращает null

№ теста	И5
Тип теста	Позитивный
Описание теста	Проверка интеграции классов LoginHelper и TokenHelper
Объект тестирования	Классы LoginHelper и TokenHelper
Входные данные	-
Косвенные входные данные	В SharedPreferences нет строки по ключу "access_token"
Ожидаемый результат	LoginHelper отправляет TokenHelper запрос на установку токена, LoginHelper.getToken возвращает значение токена

№ теста	И6
Тип теста	Позитивный
Описание теста	Проверка интеграции классов SensorHelper и TokenHelper
Объект тестирования	Классы SensorHelper и TokenHelper
Входные данные	-
Косвенные входные данные	В SharedPreferences есть строка по ключу "access_token"
Ожидаемый результат	SensorHelper пытается получить от TokenHelper токен авторизации, SensorHelper.getToken возвращает значение токена

№ теста	И7
Тип теста	Позитивный
Описание теста	Проверка интеграции классов SensorHelper и TokenHelper
Объект тестирования	Классы SensorHelper и TokenHelper
Входные данные	-
Косвенные входные данные	В SharedPreferences нет строки по ключу "access_token"
Ожидаемый результат	SensorHelper пытается получить от TokenHelper токен авторизации, SensorHelper.getToken возвращает null

№ теста	И8
Тип теста	Позитивный
Описание теста	Проверка интеграции классов SensorParser и SensorHelper
Объект тестирования	Классы SensorParser и SensorHelper
Входные данные	-
Косвенные входные данные	SensorHelper получил json для парсинга
Ожидаемый результат	SensorHelper отправляет SensorParser json для парсинга, SensorParser имеет поле SensorArray - не пустой список

№ теста	И9
Тип теста	Позитивный
Описание теста	Проверка интеграции классов MapPageState и SensorParser
Объект тестирования	Классы MapPageState и SensorParser
Входные данные	-
Косвенные входные данные	SensorParser имеет поле SensorArray - не пустой список
Ожидаемый результат	MapPageState получает от SensorParser SensorArray, на экране есть текст "Всего сенсоров 1"

3.4 Аттестационное тестирование

№ теста	A1
Тип теста	Позитивный
Описание теста	Проверка функциональной возможности авторизации в системе
Объект тестирования	Функция авторизации
Входные данные	Существующие в базе данных логин и пароль
Косвенные входные данные	Пользователь не авторизован
Сценарий	Пользователь находится на экране авторизации, вводит логин и пароль, нажимает на кнопку «Войти»
Ожидаемый результат	Пользователь авторизован. При повторном заходе на экран авторизации отображается текст «Вы уже авторизованы», кнопка «Выйти»

№ теста	A2
Тип теста	Негативный
Описание теста	Проверка функциональной возможности авторизации в системе
Объект тестирования	Функция авторизации
Входные данные	Несуществующие в базе данных логин и пароль
Косвенные входные данные	Пользователь не авторизован
Сценарий	Пользователь находится на экране авторизации, вводит логин и пароль, нажимает на кнопку «Войти»
Ожидаемый результат	Пользователь неавторизован. На экране авторизации появляется сообщение «Login failed»

№ теста	А3
Тип теста	Позитивный
Описание теста	Проверка функциональной возможности выхода из системы
Объект тестирования	Функция выхода
Входные данные	-
Косвенные входные данные	Пользователь авторизован
Сценарий	Пользователь находится на экране авторизации, нажимает на кнопку «Выйти»
Ожидаемый результат	Пользователь неавторизован. Отображается экран авторизации с формой входа

№ теста	А4
Тип теста	Позитивный
Описание теста	Проверка функциональной возможности просмотра карты
Объект тестирования	Функция просмотра карты
Входные данные	-
Косвенные входные данные	Пользователь авторизован
Сценарий	Пользователь находится на экране карты
Ожидаемый результат	Отображается экран карты, на карте нет маркеров

№ теста	A5
Тип теста	Позитивный
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств», ожидает 1 минуту
Ожидаемый результат	Отображается экран карты, на карте отображаются маркеры доступных устройств

№ теста	A6
Тип теста	Негативный
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь не авторизован
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств», ожидает 1 минуту
Ожидаемый результат	Отображается экран карты, на карте нет маркеров. На экране карты появляется сообщение «Нет доступных данных, авторизуйтесь для продолжения»

№ теста	A7
Тип теста	Позитивный
Описание теста	Проверка функциональной возможности просмотра краткой информации о выбранном устройстве
Объект тестирования	Функция отображения просмотра краткой информации о выбранном устройстве
Входные данные	-
Косвенные входные данные	Пользователь авторизацию, выполнил запрос данных устройств, который прошел успешно (устройства отобразились на карте)
Сценарий	Пользователь находится на экране карты, нажимает на маркер, расположенный на Кубе
Ожидаемый результат	Отображается экран карты, на карте отображаются маркеры доступных устройств. На экране карты, в области выдвижной панели, выведен 2 блока текста «Сюда вставляем краткую информацию о датчике» и «Например, сейчас выбран датчик с названием Большая Лепринда»

3.5 Специальное тестирование

№ теста	C1
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загрузенности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - LTE, уровень сигнала - отличный, сервер не нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 3 секунд

№ теста	C2
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - LTE, уровень сигнала - отличный, сервер умеренно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 5 секунд

№ теста	СЗ
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - LTE, уровень сигнала - отличный, сервер сильно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 10 секунд

№ теста	С4
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загрузенности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - отличный, сервер не нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 5 секунд

№ теста	С5
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - отличный, сервер умеренно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 7 секунд

№ теста	С6
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - отличный, сервер сильно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 12 секунд

№ теста	C7
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - плохой, сервер не нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 7 секунд

№ теста	С8
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - плохой, сервер умеренно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 10 секунд

№ теста	С9
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загрузенности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - плохой, сервер сильно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 15 секунд

№ теста	C10
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - отсутствует, сервер не нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	После 20 секунд ожидания появляется сообщение «Нет доступных данных, сервер не отвечает»

№ теста	C10
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загруженности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - отсутствует, сервер умеренно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	После 20 секунд ожидания появляется сообщение «Нет доступных данных, сервер не отвечает»

№ теста	C10
Описание теста	Проверка функциональной возможности отображения доступных устройств на карте при различных условиях доступа к сети и загрузенности сервера
Объект тестирования	Функция отображения доступных устройств на карте
Входные данные	-
Косвенные входные данные	Пользователь авторизован, тип подключения - UMTS (3G), уровень сигнала - отсутствует, сервер сильно нагружен
Сценарий	Пользователь находится на экране карты, нажимает на кнопку «Запросить данные устройств»
Ожидаемый результат	После 20 секунд ожидания появляется сообщение «Нет доступных данных, сервер не отвечает»

3.6 Примеры тестов

```
test('tokenHelper set test', () async {
  SharedPreferences.setMockInitialValues({});

  final tokenHelper = TokenHelper();

  await tokenHelper.setToken("tok");

  final SharedPreferences prefs = await SharedPreferences.getInstance();
  String? token = prefs.getString("access_token");

  expect(token, "tok");
});
```

Лист. 1 – Пример блочного теста

```
testWidgets('Navigation drawer test', (WidgetTester tester) async {
  await tester.pumpWidget(
    const MaterialApp(home: MyHomePage(title: "Flutter Map Demo",)));

  expect(find.text('Flutter Map Demo'), findsOneWidget);
  expect(find.byType(DrawerButton), findsWidgets);
  await tester.tap(find.byType(DrawerButton));
  await tester.pump();

  expect(find.text('Профиль'), findsOneWidget);
  expect(find.text('Карта'), findsOneWidget);
  expect(find.text('Настройки'), findsOneWidget);
});
```

Лист. 2 – Пример виджет теста

```
IntegrationTestWidgetsFlutterBinding.ensureInitialized();
```

```

group('end-to-end test', () {
  group('LoginPageState - LoginHelper integration', () {
    testWidgets('Login got token test', (WidgetTester tester) async {
      SharedPreferences.setMockInitialValues({"access_token" : "tok"});
      await tester.pumpWidget(const MyApp());

      await tester.pump();
      expect(find.text('Flutter Map Demo'), findsOneWidget);

      expect(find.text('Вы уже авторизованы'), findsOneWidget);

      expect (find.ancestor(
        of: find.text('Выйти'),
        matching: find.byType(TextButton),
      ), findsWidgets,);

    });
  });
});

```

Лист. 3 – Пример интеграционного теста

4 Журнал тестирования

4.1 Блочное тестирование

Тест	Дата	Тестирующий	Попытки	Результат	Отчет об ошибке
Б1	30.11.2023	Беленков Н. Д.	2	Пройден	Отчет 1
Б2	30.11.2023	Беленков Н. Д.	2	Пройден	Отчет 2
Б3	30.11.2023	Беленков Н. Д.	2	Пройден	Отчет 3
Б4	30.11.2023	Беленков Н. Д.	1	Пройден	
Б5	30.11.2023	Беленков Н. Д.	1	Пройден	
Б6	30.11.2023	Беленков Н. Д.	1	Пройден	
Б7	30.11.2023	Беленков Н. Д.	1	Пройден	
Б8	30.11.2023	Беленков Н. Д.	1	Пройден	
Б9	30.11.2023	Беленков Н. Д.	1	Пройден	
Б10	30.11.2023	Беленков Н. Д.	1	Пройден	
Б11	30.11.2023	Беленков Н. Д.	1	Пройден	
Б12	30.11.2023	Беленков Н. Д.	1	Пройден	
Б13	30.11.2023	Беленков Н. Д.	1	Пройден	

4.2 Тестирование виджетов

Тест	Дата	Тестирующий	Попытки	Результат	Отчет об ошибке
В1	30.11.2023	Беленков Н. Д.	1	Пройден	
В2	30.11.2023	Беленков Н. Д.	1	Пройден	
В3	30.11.2023	Беленков Н. Д.	1	Пройден	
В4	30.11.2023	Беленков Н. Д.	2	Пройден	Отчет 4
В5	30.11.2023	Беленков Н. Д.	2	Пройден	Отчет 5

4.3 Интеграционное тестирование

Тест	Дата	Тестирующий	Попытки	Результат	Отчет об ошибке
И1	30.11.2023	Беленков Н. Д.	2	Пройден	Отчет 6
И2	30.11.2023	Беленков Н. Д.	1	Пройден	
И3	30.11.2023	Беленков Н. Д.	1	Пройден	
И4	30.11.2023	Беленков Н. Д.	1	Пройден	
И5	30.11.2023	Беленков Н. Д.	1	Пройден	
И6	30.11.2023	Беленков Н. Д.	1	Пройден	
И7	30.11.2023	Беленков Н. Д.	1	Пройден	
И8	30.11.2023	Беленков Н. Д.	1	Пройден	
И9	30.11.2023	Беленков Н. Д.	2	Пройден	Отчет 7

4.4 Аттестационное тестирование

Тест	Дата	Тестирующий	Попытки	Результат	Отчет об ошибке
А1	30.11.2023	Беленков Н. Д.	1	Пройден	
А2	30.11.2023	Беленков Н. Д.	1	Пройден	
А3	30.11.2023	Беленков Н. Д.	1	Пройден	
А4	30.11.2023	Беленков Н. Д.	1	Пройден	
А5	30.11.2023	Беленков Н. Д.	1	Пройден	
А6	30.11.2023	Беленков Н. Д.	1	Пройден	
А7	30.11.2023	Беленков Н. Д.	1	Пройден	

4.5 Специальное тестирование

Тест	Дата	Тестирующий	Попытки	Результат	Отчет об ошибке
C1	13.12.2023	Беленков Н. Д.	2	Пройден	Отчет 8
C2	13.12.2023	Беленков Н. Д.	1	Пройден	
C3	13.12.2023	Беленков Н. Д.	1	Пройден	
C4	13.12.2023	Беленков Н. Д.	1	Пройден	
C5	13.12.2023	Беленков Н. Д.	1	Пройден	
C6	13.12.2023	Беленков Н. Д.	1	Пройден	
C7	13.12.2023	Беленков Н. Д.	1	Пройден	
C8	13.12.2023	Беленков Н. Д.	1	Пройден	
C9	13.12.2023	Беленков Н. Д.	1	Пройден	
C10	13.12.2023	Беленков Н. Д.	1	Пройден	
C11	13.12.2023	Беленков Н. Д.	1	Пройден	
C12	13.12.2023	Беленков Н. Д.	1	Пройден	

5 Журнал найденных ошибок

№ отчета об ошибке	1
Дата составления отчета	30.11.2023
Тест	Б1
Объект тестирования	loginHelper : метод checkPassword(String user, String pass, http.Client client)
Ожидаемый результат	Запрос выполнен, метод вернул значение True
Фактический результат	Запрос не выполнен, тест вернул ошибку MissingStubError
Решение	Добавления стаба в тест - when(client.post()).thenAnswer((_) async => http.Response());
Статус	Решена

№ отчета об ошибке	2
Дата составления отчета	30.11.2023
Тест	Б2
Объект тестирования	loginHelper : метод checkPassword(String user, String pass, http.Client client)
Ожидаемый результат	Exception('http.post error: statusCode=500')
Фактический результат	Тест не поймал исключение и был провален
Решение	Добавление строки expect(loginHelper.checkPassword(args) ,throwsException) в тест
Статус	Решена

№ отчета об ошибке	3
Дата составления отчета	30.11.2023
Тест	Б3
Объект тестирования	LoginHelper : метод parseLoginJson(Map<String, dynamic> json)
Ожидаемый результат	Метод вернул значение True
Фактический результат	Тест вернул ошибку «Binding has not yet been initialized»
Решение	Добавление строки IntegrationTestWidgetsFlutterBinding.ensureInitialized(); перед выполнением тестов
Статус	Решена

№ отчета об ошибке	4
Дата составления отчета	30.11.2023
Тест	В4
Объект тестирования	Шторка навигации
Ожидаемый результат	На экране отображается шторка навигации, в ней есть элементы «Профиль», «Карта» и «Настройки»
Фактический результат	Тест не пройден, элементы «Профиль», «Карта» и «Настройки» не найдены
Решение	Добавление поиска кнопки открытия шторки и нажатия на нее в тест
Статус	Решена

№ отчета об ошибке	5
Дата составления отчета	30.11.2023
Тест	B5
Объект тестирования	Экран карты
Ожидаемый результат	На экране отображается карта, выдвигаемая панель
Фактический результат	Тест вернул исключение ClientException
Решение	Замена адреса с 127.0.0.1 на 10.0.2.2 (адрес хоста для эмулятора)
Статус	Решена

№ отчета об ошибке	6
Дата составления отчета	30.11.2023
Тест	И1
Объект тестирования	Классы LoginPageState и LoginHelper
Ожидаемый результат	LoginHelper пытается получить от TokenHelper токен авторизации, LoginHelper.getToken возвращает значение токена
Фактический результат	Тест вернул исключение MissingPluginException
Решение	Добавление инициализации SharedPreferences в тест - SharedPreferences.setMockInitialValues()
Статус	Решена

№ отчета об ошибке	7
Дата составления отчета	30.11.2023
Тест	И9
Объект тестирования	Классы MapPageState и SensorParser
Ожидаемый результат	MapPageState получает от SensorParser SensorArray, на экране есть текст «Всего сенсоров 1»
Фактический результат	Тест не пройден, текст «Всего сенсоров 1» не найден
Решение	Добавление ожидания и обновления интерфейса в тест
Статус	Решена

№ отчета об ошибке	8
Дата составления отчета	13.12.2023
Тест	С1
Объект тестирования	Функция отображения доступных устройств на карте
Ожидаемый результат	От нажатия кнопки до появления маркеров на карте проходит не более 3 секунд
Фактический результат	Запрос провалился с исключением ClientException
Решение	Замена адреса с 127.0.0.1 на 10.0.2.2 (адрес хоста для эмулятора)
Статус	Решена

6 Оценка покрытия кода тестами

Общее покрытие кода тестами составляет 77.1% по оценкам автоматического тестирования с помощью библиотеки «Flutter Test».

7 Результаты

В результате данной работы было проведено 46 тестов, среди них 13 блочных, 5 тестов виджетов, 9 интеграционных, 7 аттестационных, а также 12 специальных тестов. В ходе проведения тестов были выявлены не критичные ошибки в работе системы.

После разработки тестов было решено добавить обработку условий негативных тестов в исходный код программы для повышения стабильности работы программы.

Многие из непройденных с первой попытки тестов были провалены ввиду ошибок в реализации самих тестов, малого опыта работы со средствами эмуляции Android. В ходе работы над их исправлением навыки в работе с тестами для flutter были улучшены, тестирование новых функций системы будет возможно провести в более короткие сроки.