

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Петрозаводский государственный университет»

Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Сысоев Максим Федорович

Отчет по дисциплине «Верификация программного обеспечения»

Направление 09.04.02 — Информационные системы и технологии
Программа «Управление данными»

Преподаватель: к.ф.-м.н., доцент К.А. Кулаков

Петрозаводск
2020

Содержание

Содержание	2
Объект тестирования	3
Описание программы	3
Архитектура системы	3
Описание тестируемого модуля	4
Описание тестируемого функционала	5
Стратегия тестирования	5
Блочное тестирование	5
Метод PostNewValue	5
Метод HaveAccessToMeteringDevice	7
Метод CheckNewValue	8
Метод AssessChanges	9
Интеграционное тестирование	10
Аттестационное тестирование	12
Нагрузочное тестирование	12
Детальный план тестов	12
Блочное тестирование	12
Заглушки	12
Метод AssessChanges	13
Метод CheckNewValue	15
Метод HaveAccessToMeteringDevice	17
Метод PostNewValue	18
Интеграционное тестирование	20
Аттестационное тестирование	22
Нагрузочное тестирование	24
Покрытие кода тестами	25
Примеры реализации тестов	26
Журнал тестирования	27
Журнал найденных ошибок	28
Результаты	29

Объект тестирования

Описание программы

В рамках курса «Верификация программного обеспечения» будет рассмотрена система по обеспечению жильцов возможностью взаимодействовать с поставщиками услуг ЖКХ.

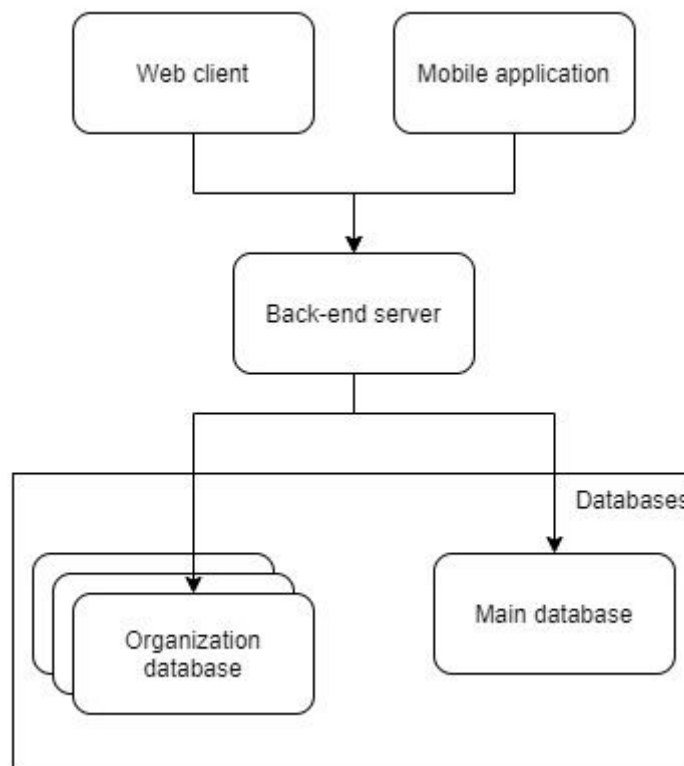
Данная система предоставляет пользователю возможности:

- Авторизации
- Создания списка адресов
- Привязки к адресу лицевого счета организации
- Передача показаний счетчиков
- Просмотр и оплату квитанций
- Просмотр новых событий по адресу

Архитектура системы

Система состоит из следующих модулей:

1. Приложение пользователя. Этот модуль предоставляет пользовательский интерфейс и отвечает за взаимодействие с back-end сервером;
2. Back-end сервер. Этот модуль отвечает за обработку запросов пользовательского приложения и взаимодействия с базой данных;
3. База данных. Этот модуль отвечает за хранение данных и передачу их в back-end сервер.



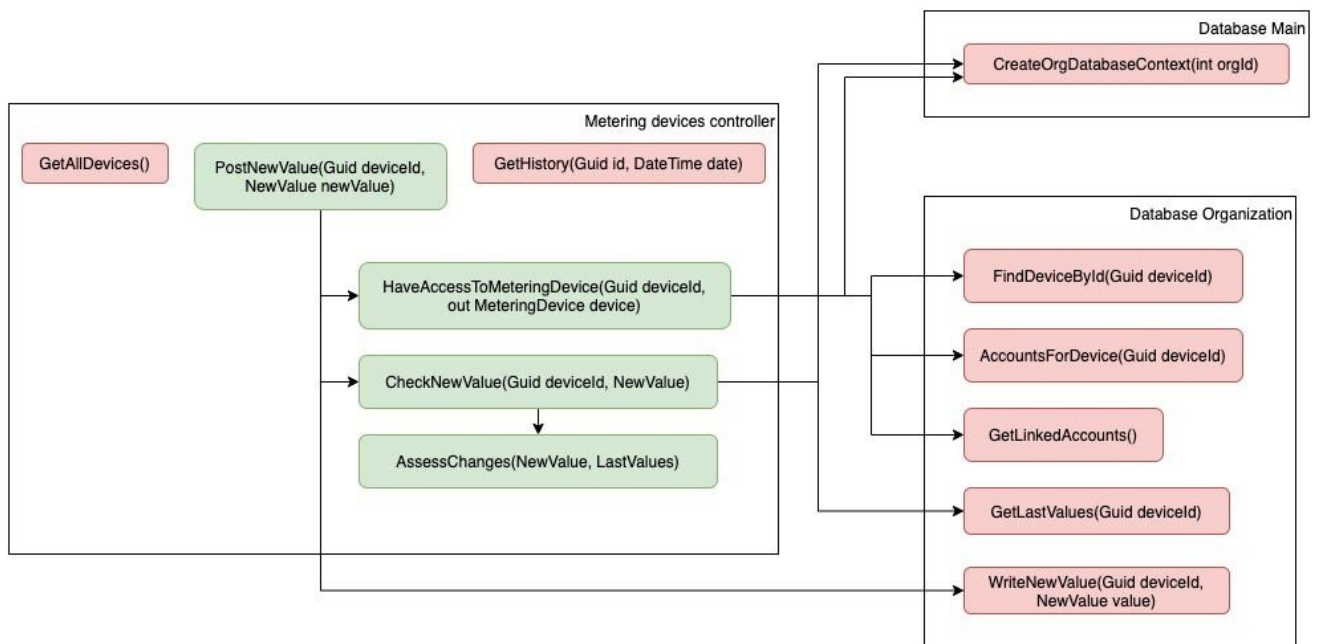
Используемые технологии в системе:

1. Angular версии 9 для создания пользовательского приложения в web
2. Flutter для создания кроссплатформенного мобильного приложения
3. ASP.NET для обеспечения обработки запросов из приложения пользователя
4. Entity Framework для взаимодействия с базой данных из back-end сервера
5. PostgreSQL для создания базы данных

Описание тестируемого модуля

Тестироваться будет back-end сервер, который состоит из модулей:

- Database.Main – модуль отвечающий за взаимодействие с основной базой данных, её создание, миграцию.
- Database.Organization – модуль отвечающий за взаимодействие с базами данных организаций, их создание, миграции.
- ClientApi – модуль отвечающий за обработку запросов с приложения пользователя.



В рамках дисциплины будет частично протестирована функциональная возможность ClientApi, а именно возможность передачи показаний счетчика в организацию и промежуточная проверка введенных данных.

Данные методы не будут проверяться в ходе тестирования и считается что они работают корректно:

- GetAllDevices – возвращает список всех приборов учета для аккаунта
- GetHistory – возвращает историю показаний для прибора учета

- CreateOrgDatabaseContext – создает DbContext для взаимодействия с базой данных организации
- FindDeviceById – возвращает прибор учета по id
- AccountsForDevices – возвращает аккаунты привязанные к прибору учета
- GetLinkedAccounts – возвращает связанные лицевые счета
- GetLastValues – возвращает последние показания прибора учета
- WriteNewValue – записать новое показание прибора учета

Работа нижеперечисленных методы будет проверена в ходе тестирования:

- PostNewValue – производит проверку передаваемых значений, записывает новые показания и возвращает ответ клиенту
- HasAccountsToMeteringDevices – производит проверку, есть ли доступ до прибора учета у пользователя и возвращает объект прибора учета
- CheckNewValue – производит проверку новых значений на
- AssessChanges – проверяет, попадает ли новое значение в оценочный интервал

Описание тестируемого функционала

Перечень требований, предполагаемых к тестированию:

1. Если происходит попытка записи показания на несуществующий прибор учета должна возвращаться ошибка BadRequest;
2. Если пользователь пытается передать показания для чужого прибора учета, то возвращается ошибка AccessDenied;
3. Программа должна возвращать сообщение о возможной ошибке в передаваемых показаниях, если они не попадают в оценочный интервал;
4. Программа должна записывать показания приборов учета, если они прошли проверку.

Стратегия тестирования

Блочное тестирование

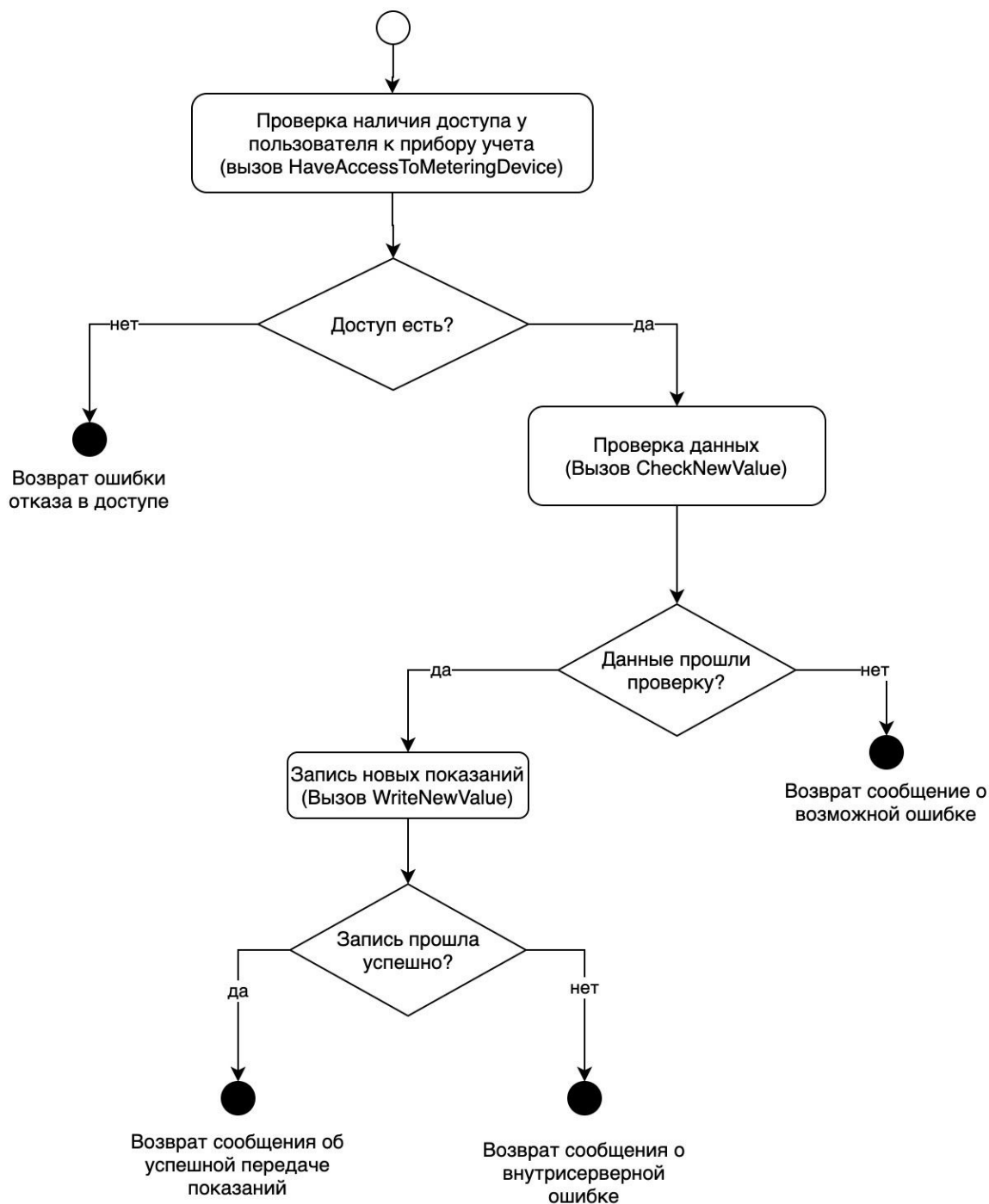
Блочное тестирование будет проводиться методом автоматизированного тестирования с использованием библиотеки Xunit. Для проведения блочных тестов должны использоваться заглушки в репозитории баз данных.

Блочное тестирование должно быть применено к следующим методам: PostNewValue, HasAccountsToMeteringDevices, CheckNewValue, AssessChanges. Для каждого пути в блок-схеме метода должен быть разработан как минимум один тест.

Метод PostNewValue

Объект	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
--------	---

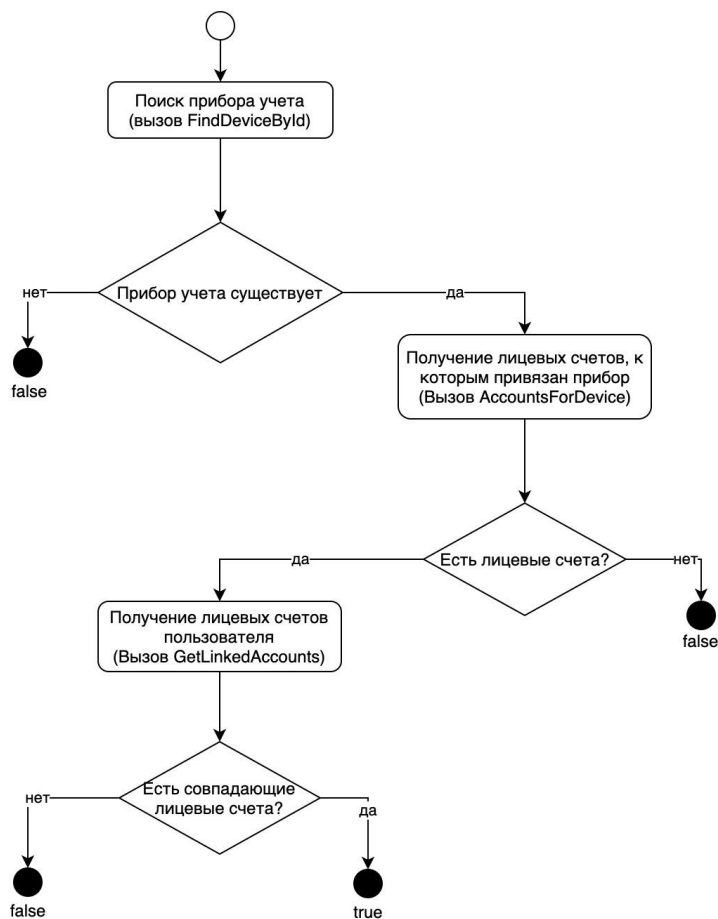
Расположение	ClientApi.Controllers.MeteringDevicesController
Назначение	Точка приема запроса для передачи показаний приборов учета. Запускается при получении запроса на роуте "api/v1.0/metering-devices/readings-input".
Входные параметры	<ul style="list-style-type: none"> • deviceId – идентификатор прибора учета для которого передаются показания. Тип данных OrganizationEntityId. • newValue – объект содержащий новые показания приборов учета. Получается из тела post запроса. Тип данных NewValue.
Выходные параметры	<ul style="list-style-type: none"> • Ответ со статусом "Access denied" – если у пользователя нет доступа к данному прибору учета • Ответ со статусом "Internal server error" – если произошла ошибка при обращении к базе данных • Ответ со статусом успешно – если данные были записаны • Ответ с сообщением о возможной ошибке при вводе показаний – если передаваемые данные не прошли проверку



Метод HaveAccessToMeteringDevice

Объект	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Расположение	ClientApi.Controllers.MeteringDevicesController
Назначение	Проверяет, есть ли у пользователя доступ до счетчика с данным идентификатором. Если доступ есть, то возвращается объект

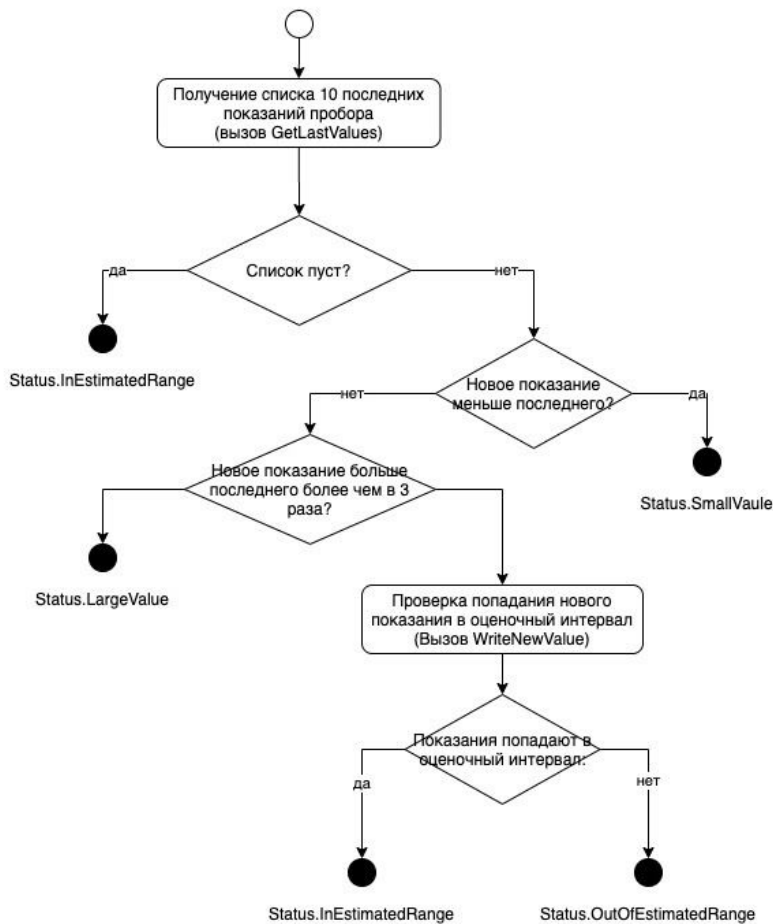
	прибора учета.
Входные параметры	<ul style="list-style-type: none"> • orgDbContext - контекст базы данных организации. Тип данных DbContext. • deviceId – идентификатор прибора учета для которого передаются показания. Тип данных OrganizationEntityId. • linkedOrganizationAccountsReader – репозиторий для взаимодействия с базой данных организации. Получается из dependency injection. Тип данных ILinkedOrganizationAccountsReader.
Выходные параметры	<ul style="list-style-type: none"> • false – если у пользователя нет доступа до прибора учета с данным идентификатором, если прибора учета с таким идентификатором не существует • true – если доступ есть, в таком случае также заполняется аргумент device найденным прибором учета



Метод CheckNewValue

Объект	private Status CheckNewValue(orgDbContext, device, newValue)
Расположение	ClientApi.Controllers.MeteringDevicesController

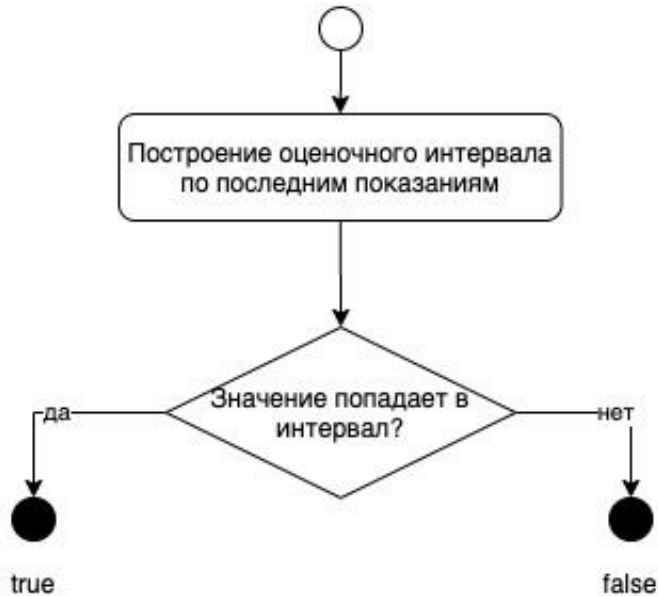
Назначение	Проверка переданных показаний на корректность
Входные параметры	<ul style="list-style-type: none"> • orgDbContext - контекст базы данных организации. Тип данных DbContext. • device – прибор учета для которого передаются показания. Тип данных MeteringDevice. • newValue – объект содержащий новые показания приборов учета. Тип данных NewValue.
Выходные параметры	<ul style="list-style-type: none"> • Status.SmallValue – возвращается, если переданное показание меньше последнего • Status.LargeValue – возвращается, если значение больше предыдущего более чем в 3 раза • Status.OutOfEstimatedRange – возвращается, если значение не попадает в интервал предполагаемого значения • Status.InEstimatedRange – возвращается, если значение попадает в интервал предполагаемого значения



Метод AssessChanges

Объект	private bool AssessChanges(newValue, lastValues)
Расположение	ClientApi.Controllers.MeteringDevicesController

Назначение	Проверка переданных показаний на попадание в оценочный интервал
Входные параметры	<ul style="list-style-type: none"> newValue – объект содержащий новые показания приборов учета. Тип данных NewValue. lastValues – список предыдущих показаний. Должен быть не пустым. Тип данных List<Value>.
Выходные параметры	<ul style="list-style-type: none"> true – показание попало в интервал false – показание не попало в интервал



Интеграционное тестирование

Интеграционное тестирование будет проводиться методом автоматизированного тестирования с использованием библиотеки Xunit. Для проведения интеграционных тестов должна создаваться временная база данных и заполняться данными для тестирования. База должна удаляться после прохождения тестов. Интеграционное тестирование будет проверять корректность работы баз данных.

Таблицы должны быть заполнены такими данными:

PersonalAccounts		
id	name	loginType
1	Сысоев Максим	vkontakte

PersonalAddresses

id	personalAccountId	fias
1	1	5ee84ac0-eb9a-4b42-b814-2f5f7c27c255

Accounts		
id	number	pinCodeHash
1	001	202cb962ac59075b964b07152d234b70

LinkedAccounts	
accountId	personalAddressId
1	1

MeteringDevices		
id	number	name
'505934a3-2b29-11eb-826b-7085c291e054'	1111	ГBC
'7085c291e-2b29-11eb-826b-7085c291e054'	1112	ГBC

MeteringDeviceToAccountAssign	
meteringDeviceId	accountId
'505934a3-2b29-11eb-826b-7085c291e054'	1

MeteringDeviceValues			
meteringDeviceId	value	month	tariffZoneId
'505934a3-2b29-11eb-826b-7085c291e054'	140	2020-11	1
'505934a3-2b29-11eb-826b-7085c291e054'	130	2020-10	1

'505934a3-2b29-11eb-826b-7085c291e054'	120	2020-09	1
--	-----	---------	---

Аттестационное тестирование

Аттестационное тестирование будет проводиться методом ручного тестирования с использованием инструмента Fiddler 4. Для проведения аттестационных тестов должна быть использована тестовые базы данных, заполненной данными из интеграционного тестирования.

Нагрузочное тестирование

Нагрузочное тестирование будет проводиться методом автоматизированного тестирования с использованием Apache JMeter. Для проведения нагрузочных тестов должна быть использована тестовая база данных, содержащая примерно 300 лицевых счетов, 450 приборов учета, 4500 показаний.

В рамках нагрузочного тестирования будет проверяться скорость работы метода PostNewValue. При тестировании будет выполняться 100, 300, 1000 запросов и оцениваться время выполнения всех запросов.

Детальный план тестов

Блочное тестирование

Для проведения блочного тестирования необходимо создать заглушки на основе интерфейсов:

- OrganizationAccountsRepository
- IOrganizationDbContextFactory
- MeteringDeviceValuesRepository
- MeteringDeviceRepository

Заглушки

ILinkedOrganizationAccountsRepository:

- Метод getLinkedAccounts возвращает [

```
{
    id: 1,
}
```

]

MeteringDeviceValuesRepository:

- Метод getLastValues возвращает: [

```
{
    month: 2020-11,
```

```

        tariffZoneld: 1,
        value: 140,
    },
    {
        month: 2020-10,
        tariffZoneld: 1,
        value: 130,
    },
    {
        month: 2020-19,
        tariffZoneld: 1,
        value: 120,
    },
]

```

- Метод writeValue используется для детекции вызова

MeteringDeviceRepository:

- FindDeviceById
 - при аргументе равном '505934a3-2b29-11eb-826b-7085c291e054' возвращает: {
 - id: '505934a3-2b29-11eb-826b-7085c291e054',
 - name: 'ГВС'
 - при аргументе равном '7085c291e-2b29-11eb-826b-7085c291e054' возвращает: {
 - id: '7085c291e-2b29-11eb-826b-7085c291e054',
 - name: 'ГВС'
 - при других аргументах возвращает null
- AccountsForDevice
 - при аргументе равном '505934a3-2b29-11eb-826b-7085c291e054' возвращает: [
 - {
 - id: '1',
 - при других аргументах []

Метод AssessChanges

№	Б1
Описание	Общий, положительный Проверяется правильность оценки соответствия нового показания оценочному интервалу.

Объект тестирования	private bool AssessChanges(newValue, lastValues)
Входные данные	<ul style="list-style-type: none"> • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneld: 1, value: 150, • lastValues: [<ul style="list-style-type: none"> { <ul style="list-style-type: none"> month: 2020-11, tariffZoneld: 1, value: 140, }, { <ul style="list-style-type: none"> month: 2020-10, tariffZoneld: 1, value: 130, }, { <ul style="list-style-type: none"> month: 2020-19, tariffZoneld: 1, value: 120, },
Ожидаемый результат	true

№	Б2
Описание	Краевой, положительный Проверяется правильность оценки соответствия нового показания оценочному интервалу.
Объект тестирования	private bool AssessChanges(newValue, lastValues)
Входные данные	<ul style="list-style-type: none"> • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneld: 1, value: 140, • lastValues: [<ul style="list-style-type: none"> { <ul style="list-style-type: none"> month: 2020-11, tariffZoneld: 1, value: 140, }, { <ul style="list-style-type: none"> month: 2020-10, tariffZoneld: 1,

	<pre> value: 130, }, { month: 2020-19, tariffZoneld: 1, value: 120, },] </pre>
Ожидаемый результат	false

№	БЗ
Описание	Краевой, положительный Проверяется правильность оценки соответствия нового показания оценочному интервалу.
Объект тестирования	private bool AssessChanges(newValue, lastValues)
Входные данные	<ul style="list-style-type: none"> • newValue: { <pre> month: 2020-12, tariffZoneld: 1, value: 165, } </pre> • lastValues: [<pre> { month: 2020-11, tariffZoneld: 1, value: 140, }, { month: 2020-10, tariffZoneld: 1, value: 130, }, { month: 2020-19, tariffZoneld: 1, value: 120, },] </pre>
Ожидаемый результат	false

Метод CheckNewValue

№	Б5
Описание	Краевой, положительный Проверка работы с большим новым показанием
Объект тестирования	private Status CheckNewValue(orgDbContext, device, newValue)
Входные данные	<ul style="list-style-type: none"> • orgDbContext – mock • device: { id: '505934a3-2b29-11eb-826b-7085c291e054', } • newValue: { month: 2020-12, tariffZoneld: 1, value: 1500, }
Ожидаемый результат	Status.LargeValue

№	Б6
Описание	Краевой, положительный Проверка работы с новым показанием меньше предыдущего
Объект тестирования	private Status CheckNewValue(orgDbContext, device, newValue)
Входные данные	<ul style="list-style-type: none"> • orgDbContext – mock • device: { id: '505934a3-2b29-11eb-826b-7085c291e054', } • newValue: { month: 2020-12, tariffZoneld: 1, value: 139, }
Ожидаемый результат	Status.SmallValue

№	Б7
Описание	Общий, положительный Проверка работы с новым показанием не попадающим в оценочный интервал
Объект тестирования	private Status CheckNewValue(orgDbContext, device, newValue)

Входные данные	<ul style="list-style-type: none"> • orgDbContext – mock • device: { id: '505934a3-2b29-11eb-826b-7085c291e054', } • newValue: { month: 2020-12, tariffZoneld: 1, value: 170, }
Ожидаемый результат	Status.OutOfEstimatedRange

Метод HaveAccessToMeteringDevice

№	Б8
Описание	Общий, положительный Проверка работы с прибором учета, до которого есть доступ
Объект тестирования	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Входные данные	<ul style="list-style-type: none"> • orgDbContext – mock • linkedOrganizationAccountsReader – mock • deviceId: '505934a3-2b29-11eb-826b-7085c291e054'
Ожидаемый результат	true

№	Б9
Описание	Общий, негативный Проверка работы если прибора учета не существует
Объект тестирования	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Входные данные	<ul style="list-style-type: none"> • orgDbContext – mock • linkedOrganizationAccountsReader – mock • deviceId: '00000000-2b29-11eb-826b-7085c291e054'
Ожидаемый результат	false

№	Б10
Описание	Общий, положительный

	Проверка работы если прибора учета не привязан к лицевому счету
Объект тестирования	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Входные данные	<ul style="list-style-type: none"> • orgDbContext – mock • linkedOrganizationAccountsReader – mock • deviceId: '7085c291-2b29-11eb-826b-7085c291e054'
Ожидаемый результат	false

№	Б11
Описание	Общий, положительный Проверка работы если у пользователя нет лицевого счета с таким счетчиком
Объект тестирования	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Входные данные	<ul style="list-style-type: none"> • orgDbContext – mock • linkedOrganizationAccountsReader – mock • deviceId: '7085c291-2b29-11eb-826b-7085c291e054'
Ожидаемый результат	false

Метод PostNewValue

№	Б12
Описание	Общий, положительный Проверка записи подходящего показания
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneld: 1, value: 150,
Ожидаемый результат	Вызов метода записи нового показания в БД {

	<pre> success: true, } </pre>
--	-------------------------------

№	Б13
Описание	Общий, положительный Проверка записи показания, которое меньше предыдущего
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 139, <pre> } </pre>
Ожидаемый результат	<pre> { success: false, message: "Переданное показание меньше последнего переданного", } </pre>

№	Б14
Описание	Общий, положительный Проверка записи показания, которое больше предыдущего в 10 раз
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 1400, <pre> } </pre>
Ожидаемый результат	<pre> { success: false, message: "Переданное показание слишком большое", } </pre>

№	Б15
Описание	Общий, положительный Проверка записи показания, если значение имеет повышенную динамику изменений

Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 250,
Ожидаемый результат	{ <ul style="list-style-type: none"> success: false, message: "Показания сильно изменились, вы уверены что они верны?" }

Интеграционное тестирование

№	И1
Описание	Общий, положительный Проверка записи подходящего показания
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 150,
Ожидаемый результат	В БД появилась новая запись

№	И2
Описание	Общий, негативный Проверка возврата ошибки, если нет доступа до базы данных организации
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: {

	<pre> month: 2020-12, tariffZoneld: 1, value: 150, } </pre>
Ожидаемый результат	<pre> { success: false, error: internalServerError, } </pre>

№	ИЗ
Описание	Общий, негативный Проверка возврата ошибки, если нет доступа до main базы данных
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <pre> month: 2020-12, tariffZoneld: 1, value: 150, } </pre>
Ожидаемый результат	<pre> { success: false, error: internalServerError, } </pre>

№	И4
Описание	Общий, негативный Проверка работы с прибором учета, которого не существует
Объект тестирования	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Входные данные	<ul style="list-style-type: none"> • deviceId: '00000000-2b29-11eb-826b-7085c291e054'
Ожидаемый результат	ArgumentException

№	И5
Описание	Общий, положительный

	Проверка работы с прибором учета, до которого нет доступа
Объект тестирования	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Входные данные	<ul style="list-style-type: none"> • deviceId: '7085c291-2b29-11eb-826b-7085c291e054'
Ожидаемый результат	false

№	И6
Описание	Общий, положительный Проверка работы с прибором учета, до которого есть доступ
Объект тестирования	private bool HaveAccessToMeteringDevice(orgDbContext, deviceId, linkedOrganizationAccountsReader, out device)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054'
Ожидаемый результат	true

Аттестационное тестирование

№	A1
Описание	Проверка записи подходящего показания
Функциональные требования	4
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 150,
Ожидаемый результат	При обращении по пути 'api/v1/send-values' запрос обрабатывается и данные записываются в БД

№	A2
----------	-----------

Описание	Проверка отказа в доступе к прибору учета
Функциональные требования	2
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '7085c291-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 150,
Ожидаемый результат	Возврат access denied (403 номер ошибки)

№	A3
Описание	Проверка отказа в доступе если не указан хэдер 'Authorization'
Функциональные требования	2
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 150,
Ожидаемый результат	Возврат unauthorized (401 номер ошибки)

№	A4
Описание	Общий, положительный Проверка возврата ошибки, если переданного id нет прибора учета в базе данных
Функциональные требования	1
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)

Входные данные	<ul style="list-style-type: none"> • deviceId: '00000000-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneld: 1, value: 150,
Ожидаемый результат	BadRequest

№	A5
Описание	Общий, положительный Проверка возврата ошибки, при попытке записать показания к чужому прибору учета
Функциональные требования	3
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneld: 1, value: 180,
Ожидаемый результат	{ <ul style="list-style-type: none"> success: false, message: "Показания сильно изменились, вы уверены что они верны?", }

Нагрузочное тестирование

№	H1
Описание	Общий, нагрузочный Проверка времени выполнения при отправке 100 запросов
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: {

	<pre> month: 2020-12, tariffZoneld: 1, value: 180, } </pre>
Ожидаемый результат	время выполнения каждого запроса не превышает одну секунду

№	Н2
Описание	Общий, нагрузочный Проверка времени выполнения при отправке 300 запросов
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <pre> month: 2020-12, tariffZoneld: 1, value: 180, } </pre>
Ожидаемый результат	время выполнения каждого запроса не превышает три секунды

№	Н3
Описание	Общий, нагрузочный Проверка времени выполнения при отправке 1000 запросов
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <pre> month: 2020-12, tariffZoneld: 1, value: 180, } </pre>
Ожидаемый результат	время выполнения каждого запроса не превышает 10 секунд

Покрытие кода тестами

Расчет тестового покрытия относительно исполняемого кода программного обеспечения проводится по формуле:

$Tcov = Ltc / Lcode * 100\%$, где
Tcov - тестовое покрытие;
Ltc - количество строк кода, покрытых тестами;
Lcode - общее количество строк кода.

Ltc (MeteringDevicesController) = 302
Lcode (MeteringDevicesController) = 721

Тогда Tcov (MeteringDevicesController) = $(202/721) * 100\% = 41\%$

Такой показатель был получен, поскольку в модуле остались не тестируемые функции.

Примеры реализации тестов

Пример реализации блочных тестов для функции AssetsValue

```
private NewValue largeNewValue = new NewValue() { Value = 180, TariffZoneId = 1, Month = "12-2020" };
private NewValue smallNewValue = new NewValue() { Value = 140, TariffZoneId = 1, Month = "12-2020" };
private NewValue inRangeNewValue = new NewValue() { Value = 150, TariffZoneId = 1, Month = "12-2020" };
private List<MeteringDeviceValue> lastValues = List<MeteringDeviceValue> {
    new MeteringDeviceValue() { Value = 140, TariffZoneId = 1, Month = "11-2020" },
    new MeteringDeviceValue() { Value = 130, TariffZoneId = 1, Month = "10-2020" },
    new MeteringDeviceValue() { Value = 120, TariffZoneId = 1, Month = "09-2020" },
};
```

```
[Theory]
[InlineData(largeNewValue, lastValues, false)] // Б2
[InlineData(smallNewValue, lastValues, false)] // Б3
[InlineData(inRangeNewValue, lastValues, false)] // Б1
public void AssestsValueTest(NewValue newValue, List<MeteringDeviceValue> lastValues,
bool expectedValue)
{
    var result = AssetsValue(newValue, lastValues);
    Assert.Equal(expectedValue, result);
}
```

Пример реализации интеграционного теста И1:

```
/* И1 */
[Fact]
public void WriteValueTest()
{
    var deviceId = "505934a3-2b29-11eb-826b-7085c291e054";
    var newValue = new NewValue() { Value = 180, TariffZoneId = 1, Month = "12-2020"};
```

```

var result = CreateService(AliceId, AliceOffset).PostNewValue(deviceId, newValue);

Assert.NotNull(GetMonthValuesForDevice(deviceId, newValue.month));
}

```

Журнал тестирования

№ теста	Результат	Номер ошибки в журнале
Б1	Пройден	
Б2	Пройден	
Б3	Пройден	
Б4	Пройден	
Б5	Пройден	
Б6	Пройден	
Б7	Не пройден	1
Б8	Пройден	
Б9	Пройден	
Б10	Пройден	
Б11	Пройден	
Б12	Пройден	
Б13	Пройден	
Б14	Пройден	
Б15	Пройден	
И1	Пройден	
И2	Пройден	
И3	Пройден	
И4	Пройден	
И5	Пройден	
И6	Пройден	
А1	Пройден	

A2	Пройден	
A3	Пройден	
A4	Пройден	
A5	Пройден	
H1	Пройдет	
H2	Не пройден	2
H3	Не пройден	3

Журнал найденных ошибок

№ отчета об ошибке	1
Номер теста	Б7
Объект тестирования	private Status CheckNewValue(orgDbContext, device, newValue)
Входные параметры	<ul style="list-style-type: none"> • orgDbContext – mock • device: { <ul style="list-style-type: none"> id: '505934a3-2b29-11eb-826b-7085c291e054', • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 170,
Ожидаемый результат	Status.OutOfEstimatedRange
Фактический результат	Status.InEstimatedRange
Воспроизводимость	Всегда
Статус	Не решено

№ отчета об ошибке	2
Номер теста	H2
Описание	Общий, нагрузочный Проверка времени выполнения при отправке 300 запросов

Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 180,
Ожидаемый результат	время выполнения каждого запроса не превышает три секунды
Фактический результат	Запросы с 150 возвращают InternalServerError
Воспроизводимость	Всегда
Статус	Не решено

№ отчета об ошибке	3
Номер теста	НЗ
Описание	Общий, нагрузочный Проверка времени выполнения при отправке 1000 запросов
Объект тестирования	public Response<ReadingsInputResult> PostNewValue(deviceId, newValue)
Входные данные	<ul style="list-style-type: none"> • deviceId: '505934a3-2b29-11eb-826b-7085c291e054' • newValue: { <ul style="list-style-type: none"> month: 2020-12, tariffZoneId: 1, value: 180,
Ожидаемый результат	время выполнения каждого запроса не превышает 10 секунд
Фактический результат	Запросы с 150 возвращают InternalServerError
Воспроизводимость	Всегда
Статус	Не решено

Результаты

В рамках курса «Верификация программного обеспечения» была протестирована функциональная возможность ClientApi, а именно возможность передачи показаний счетчика в организацию и промежуточная проверка введенных данных.

Блочное и интеграционное тестирование проводилось методом автоматизированного тестирования с использованием библиотеки Xunit. Нагрузочное тестирование проводилось с помощью программы Apache JMeter.