

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

Направления подготовки бакалавриата

09.03.02 - Информационные системы и технологии

09.03.04 - Программная инженерия

Отчет по дисциплине «Верификация программного обеспечения»

ИНФОРМАЦИОННАЯ СИСТЕМА «RFID MOBILE»

Выполнил:

студент 4 курса группы 22407

Р. Ю. Русаков

подпись

Преподаватель:

К.А. Кулаков, к.ф.-м.н., доцент

подпись

Содержание

1	Описание объекта тестирования	3
1.1	Функциональные возможности объекта тестирования	3
1.2	Структура объекта тестирования	6
2	Стратегия тестирования	9
2.1	Стратегия модульного тестирования	9
2.2	Стратегия интеграционного тестирования	10
2.3	Стратегия аттестационного тестирования	11
2.4	Стратегия тестирования безопасности	11
2.5	Критерии прохождения тестирования	11
3	Детальный план тестирования	12
3.1	Описание тестовых данных	12
3.2	Модульное тестирование	13
3.2.1	Метод filterObject()	13
3.2.2	Метод deleteObject()	17
3.2.3	Метод updateObject()	18
3.2.4	Метод newRent()	19
3.2.5	Метод newObject()	19
3.2.6	Метод findObject()	20
3.2.7	Метод infoRental()	21
3.2.8	Метод returnObject()	22
3.2.9	Метод rentalObject()	22
3.2.10	Метод rfidRead()	23
3.3	Интеграционное тестирование	24
3.3.1	Методы классов Logic и Activiti	24
3.4	Аттестационное тестирование	26
3.5	Тестирование безопасности	34
3.6	Примеры реализации тестов	37
3.7	Журнал тестирования	38
3.8	Найденные ошибки	41
3.9	Покрытие кода тестами	43
3.10	Результаты тестирования	44

1 Описание объекта тестирования

Объектом тестирования является информационная система "RFID mobile" разработанная для работников складских помещений г. Петрозаводска. Данная система предназначена для хранения и обработки данных о товарах, хранящихся на складе.

Информационная система реализована с помощью фреймворка Java для мобильных устройств (версии 13.1) и фреймворка Python (версии 3.11) с использованием объектно-реляционной базы данных SQLite (версии 3.39.4). Система является распределённой, групповой, автоматизированной и имеет клиент-серверную составляющую.

1.1 Функциональные возможности объекта тестирования

Информационная система "RFID mobile" выполняет следующие функции:

1. Работа серверной части системы:
 - (a) считывание данных с RFID-метки;
 - (b) получение данные объекта по RFID-метке;
 - (c) добавление данных объекта в систему;
 - (d) изменение данных объекта в системе;
 - (e) удаление данных объекта из системы;
 - (f) просмотр данных об объектах системы
 - (g) изменение статуса аренды объекта в системе;
 - (h) добавление данных об аренде объекта в систему;
 - (i) фильтрация объектов:
 - по наличию;
 - по категории;
 - по по названию.
2. Работа клиентской части системы:
 - (a) получение данных с сервера;
 - (b) отправка данных на сервера;
 - (c) визуализация данных для пользователя;

(d) вход в систему;

(e) получение данных от пользователя;

Среди пользователей информационной системы "RFID Mobile" выделены следующие категории:

1. менеджер предприятия - пользователь прошедший процедуру авторизации, имеющий права менеджер;
2. работник склада - пользователь прошедший процедуру авторизации, имеющий права работник;
3. администратор системы - пользователь, прошедший процедуру авторизации. Сотрудник предприятия с правами администратор системы;

На рисунке 1 приведены установленные права доступа для каждой категории пользователей.

менеджер	Доступ к информации об объектах системы
работник	Доступ к информации об объектах системы, изменение, удаление, добавление объектов системы
администратор системы	Доступ к информации о пользователях системы, изменение, удаление, добавление информации о пользователях системы

Рис. 1 – Установленные права доступа

1.2 Структура объекта тестирования

Информационная система "RFID Mobile" представляет собой распределённую систему с архитектурой клиент-сервер.

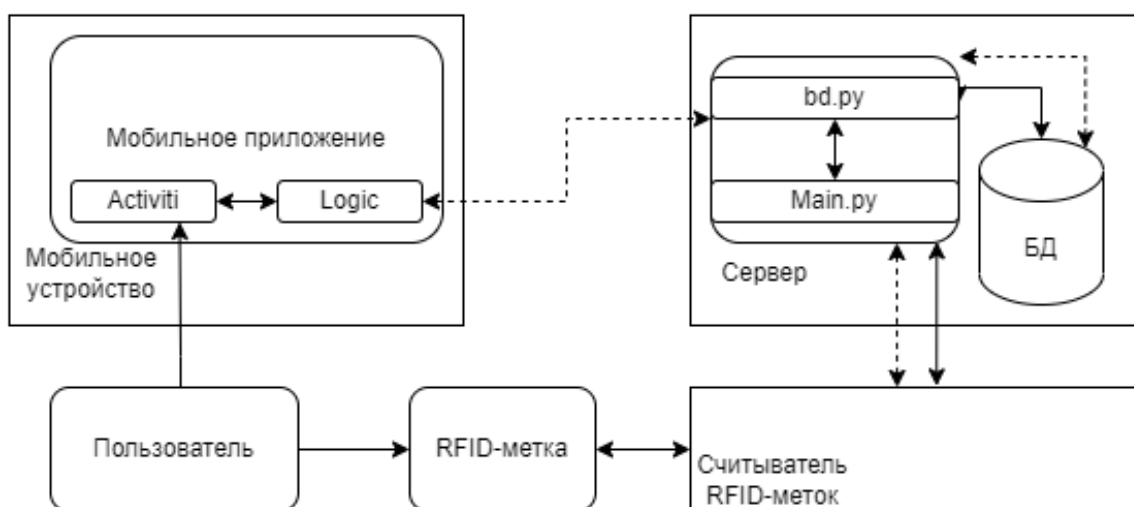


Рис. 2 – схема взаимодействия компонент информационной системы

Пользователь, использует мобильное приложение для работы с системой. Мобильное приложение состоит из графических элементов и логики клиент-серверной составляющей. Мобильное приложение взаимодействует с удалённым сервером, заполняя графические элементы полученной информацией, а так-же отправляя введённую пользователем информацию на сервер, по протоколу TCP/IP. Сервер состоит из скриптов, с помощью которых получает, обрабатывает и отправляет информацию, а так-же взаимодействует с реляционной базой данных. Сервер подключён к считывателю RFID-меток, который не имеет логики, его функционал реализуют соответствующие скрипты на сервере.

Основным классом в мобильном приложении является класс **Logic** который состоит из следующих методов:

1. `scanRfid()` - Передаёт строку "checkRfid" в функцию `con()`, возвращает строку от `connect()`;
2. `getObjectById(strng id)` - Отправляет строку "findObject|" + id в функцию `connect()`,

- разбивает полученную из функции connect() строку и заполняет экземпляр объекта, возвращает заполненный экземпляр объекта;
3. saveEdit(ObjectClass obj) - Отправляет строку "updateObject|" + obj.id + " + obj.name + " + obj.description + " + obj.category в функцию connect(), возвращает True или False;
 4. returnObject(String id) - Меняет статус объекта с "ареновано" на "на складе" и наоборот, отправляя строку "returnObject|" + id в функцию connect(), возвращает True или False;
 5. newRental(RentalClass obj) - Добавляет информацию об аренде, отправляя строку "newRent|" + obj.name + " + obj.startDate + " + obj.endDate + " + obj.objectID в функцию connect(), возвращает True или False;
 6. infoRental(String id) - Получает информацию об аренде объекта по id, отправляя строку "infoRental|" + id в функцию connect(), разбивает полученную из функции connect() строку и заполняет экземпляр объекта, возвращает заполненный экземпляр объекта;
 7. deleteObject(String id) - Удаляет объект, отправляя строку "deleteObject|" + id в функцию connect(), возвращает True или False;
 8. addObject(ObjectClass obj) - Добавляет новый объект с заданной информацией, отправляя строку "newObject|" + obj.id + " + obj.name + " + obj.description + " + obj.category в функцию connect(), возвращает True или False;
 9. getCategories() - Формирует и возвращает список заданных категорий.
 10. sort(String status, List<Boolean> category, String name) - Сортирует объекты по заданным фильтрам, формируя и отправляя строку "filterObject|" + status + " + (ArrayList) + (ArrayList)names, где categories и names имеют вид "item1item2item3...item(n)" разбивает полученную из функции connect() строку и заполняет лист экземпляров объектов, возвращает заполненный лист экземпляров объектов;
 11. connect(String msg) - Обеспечивает обмен информацией с микрокомпьютером, используя сокет, отправляет строку байтов на сервер и получает строку в ответ, возвращает декодированную строку ответа.

Сервер состоит двух модулей содержащих следующие функции:

1. Модуль `bd.py`:

- (a) `findObject(String arg1)` - Возвращает строку информации об объекте по его ID, обращаясь к базе данных;
- (b) `infoRental(String arg1)` - Возвращает строку информации об аренде объекта по его ID, обращаясь к базе данных;
- (c) `newObject(String arg1)` - Добавляет новый объект в базу данных, вызывает функцию `findObject(arg1)`;
- (d) `updateObject(String arg1, String arg2, String arg3, String arg4)` - Обновляет информацию об объекте в базе данных, вызывает функцию `findObject(arg1)`;
- (e) `newRent(String arg1, String arg2, String arg3, String arg4)` - Добавляет новую аренду объекта в базу данных, вызывает функцию `infoRental(arg1)`;
- (f) `returnObject(String arg1)` - Меняет статус объекта с "арендован" на "на складе" обращаясь к базе данных, вызывает функцию `findObject(arg1)`;
- (g) `rentalObject(String arg1)` - Меняет статус объекта с "на складе" на "арендован" обращаясь к базе данных, вызывает функцию `findObject(arg1)`;
- (h) `deleteObject(String arg1)` - Удаляет объект из базы данных, вызывает функцию `filterObject(| "`;
- (i) `filterObject(String arg1, String arg2, String arg3)` - Возвращает список объектов в базе данных, отфильтрованный по заданным критериям.

2. Модуль `main.py`:

- (a) `handle()` - Отвечает за обмен информацией с мобильным приложением, создаёт TCP сервер, ожидает запроса клиента, обрабатывает запрос вызывая функции из списка, отправляет клиенту ответ из функции;
- (b) `rfidRead()` - Ожидает приложение RFID-метки к считывателю, возвращает строку `"True|"+id`, если метка есть в базе данных, `"False|"+id`, если метка отсутствует в базе данных.

2 Стратегия тестирования

2.1 Стратегия модульного тестирования

Блочное тестирование системы будет проведено с помощью метода автоматизированного тестирования в встроенном в Android Studio фреймворке JUnit 5.0. Для заполнения системы тестовыми данными будет использована тестовая база данных и использоваться тестовая RFID-метка с идентификатором "344642296637".

Блочное тестирование не будет применено к следующим методам:

1. `getView();`
2. `onClick();`
3. `onCreate();`
4. `onActivityResult();`
5. `openObject();`
6. `openEmptyObject();`

Данные методы реализованы на базе готовых инструментов среды разработки и не нуждаются в тестировании. Оставшиеся методы будут подвержены тестированию.

2.2 Стратегия интеграционного тестирования

Интеграционное тестирование системы будет проведено с помощью метода автоматизированного тестирования в встроенном в Android Studio фреймворке JUnit 5.0. Для заполнения системы тестовыми данными будет использована тестовая база данных.

Будет рассмотрена интеграция методов классов Logic и Activiti, за исключением методов:

1. `getView();`
2. `onClick();`
3. `onCreate();`
4. `onActivityResult();`
5. `openObject();`
6. `openEmptyObject();`

Данные методы реализованы на базе готовых инструментов среды разработки и не нужны в тестировании.

Будет протестирована интеграция следующих методов:

1. `MainActivity.OnClick() -> Logic.sort();`
2. `ObjectActivity.OnClick() -> Logic.deleteObject();`
3. `ObjectActivity.OnCreate() -> Logic.getObjectById();`
4. `ObjectActivity.OnClick() -> Logic.returnObject();`
5. `FormActivity.OnClick() -> Logic.saveEdit();`
6. `FormActivity.OnClick() -> Logic.newRental();`
7. `FormActivity.OnClick() -> Logic.rentalObject();`
8. `FormActivity.OnClick() -> Logic.addObject();`

2.3 Стратегия аттестационного тестирования

Аттестационное тестирование будет проводиться посредством ручного тестирования в мобильном приложении, установленном на операционной системе Android 11.0. Для заполнения системы тестовыми данными будет использована тестовая база данных.

В результате аттестационного тестирования будет проверена корректность функционирования создания, изменения, фильтрации, аренды, просмотра и удаления объектов системы, работы RFID считывателя. В ходе тестирования будут проверены все функциональные возможности. (Информация о функциональных возможностях системы была представлена в разделе 1.1 «Функциональные возможности объекта тестирования», в пунктах 1(a)-1(i)).

2.4 Стратегия тестирования безопасности

Тестирование безопасности системы будет проведено с помощью метода автоматизированного тестирования в встроенном в Android Studio фреймворке JUnit 5.0. Для заполнения системы тестовыми данными будет использована тестовая база данных.

В результате тестирования безопасности системы должно быть проверено, что возможность неавторизованного доступа к системе и как следствие получение информации третьими лицами полностью исключена.

2.5 Критерии прохождения тестирования

Тест считается пройденным, если ожидаемый результат совпадает с фактическим. Тестирование системы будет считаться успешно пройденным, если количество пройденных тестов составляет не менее 80% от общего числа, а так же отсутствуют критические ошибки, приводящие систему или её часть в нерабочее состояние.

3 Детальный план тестирования

3.1 Описание тестовых данных

Перед выполнением тестирования таблицы tblObjects и tblRent были заполнены тестовыми данными.

Object				
ID	Name	Status	Description	Category
1	JBL Charge 3	0	music column JBL Charge 3, подключение по Bluetooth, порт AUX	musicColumn
2	JBL Club one	0	Беспроводные наушники harman cardoon, 10-20000 Гц, активное	Headphones
3	Shure SM58	1	Проводной микрофон компании SHURE, 20-25000 Гц, динамический	Microphone
4	XLR-XLR 5м.	1	Провод акустический, балансный, 5м.	Wires
5	RCA-RCA 10м.	0	Провод акустический, небалансный, 10м.	Wires
6	Jack-Jack 5м.	1	Провод акустический, небалансный, 5м	Wires

Рис. 3 – Таблица Object.

Rent				
ID	FIO	Date_rental	Date_return	Objectid
1	Филармонов Артём Павлович	10.09.2019	20.09.20219	5
2	Клависинов Геннадий Владимирович	13.06.2020	15.06.2020	2
3	Дудкин Виталий Ренатович	13.12.2022	30.12.2022	3
4	Пилатов Маврентий Драгунович	05.02.2022	12.02.2022	1
5	Безымянный Павел Сергеевич	12.12.2022	16.12.2022	6
6	Никаноров Роман Павлович	13.12.2022	20.12.2022	2

Рис. 4 – Таблица Rent.

3.2 Модульное тестирование

3.2.1 Метод filterObject()

№ теста	V1
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет вывод без указанных фильтров.
Входные данные	status = 'both', category = [], name = '. В статусе выбраны оба варианта (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список объектов, где Object.id[] = [1,2,3,4,5,6]

№ теста	B2
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку объектов со статусом "на складе"
Входные данные	status = 'false', category = [], name = '' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список объектов, которые сейчас есть в наличии на складе, где Object.id[] = [1,2,5]

№ теста	B3
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по категории
Входные данные	status = 'both', category = ['wires'], name = '' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список всех объектов которые имеют категорию 'wires', где Object.id[] = [4,5,6]

№ теста	B4
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по нескольким категориям
Входные данные	status = 'both', category = ['wires','microphone'], name = '' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список всех объектов которые имеют категории 'wires' или 'microphone', где Object.id[] = [3,4,5,6]

№ теста	В5
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по всем категориям
Входные данные	status = 'both', category = ['wires', 'microphone', 'headphones', 'musicColumn'], name = '' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список всех объектов, где Object.id[] = [1,2,3,4,5,6]

№ теста	В6
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по названию
Входные данные	status = 'both', category = [], name = 'Shure' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список объектов, в названии которых есть подстрока 'Shure', где Object.id[] = [3]

№ теста	В7
Тип теста	Негативный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по несуществующему имени
Входные данные	status = 'both', category = [], name = '(\$@&(\$)' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает пустой список

№ теста	B8
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по статусу и категории
Входные данные	status = 'false', category = ['headphones'], name = "" (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список объектов, которые имеют категорию 'headphones' и сейчас есть в наличии на складе, где Object.id[] = [2]
№ теста	B9
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по категории и имени
Входные данные	status = 'both', category = ['headphones'], name = 'JBL' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список объектов, которые имеют категорию 'headphones' и в их названии есть слово "JBL", где Object.id[] = [2]
№ теста	B10
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по статусу и имени
Входные данные	status = 'false', category = [], name = 'JBL' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список объектов, которые сейчас есть в наличии на складе и в их названии есть слово "JBL", где Object.id[] = [1,2]

№ теста	B11
Тип теста	Позитивный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку по статусу, категории и имени
Входные данные	status = 'false', category = ['headphones'], name = 'JBL' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает список объектов, которые сейчас есть в наличии на складе, имеют категорию "headphones" и в их названии есть слово "JBL", где Object.id[] = [2]

3.2.2 Метод deleteObject()

№ теста	B12
Тип теста	Позитивный
Объект тестирования	deleteObject(int objectId)
Описание теста	Проверяет удаление объекта из системы
Входные данные	id = (id существующего объекта)
Ожидаемый результат	Удаляет объект из базы данных. Возвращает значение True.

№ теста	B13
Тип теста	Негативный
Объект тестирования	deleteObject(int objectId)
Описание теста	Проверяет удаление несуществующего объекта системы
Входные данные	id = (id несуществующего объекта)
Ожидаемый результат	Возвращает значение True (т.к. не найдет ни одной строки в бд, функция ничего не удалит и вернет True, что позволит удалить отображение несуществующего объекта у клиента)

3.2.3 Метод updateObject()

№ теста	B14
Тип теста	Позитивный
Объект тестирования	updateObject(int objectId, String name, String description, String category)
Описание теста	Проверяет изменение данных объекта системы
Входные данные	objectId=3, name='JBL Charge 3', category = 'musicColumn', description='music column JBL Charge 3, подключение по Bluetooth, порт AUX'
Ожидаемый результат	Обновляет значение объекта в базе данных. Возвращает значение True

№ теста	B15
Тип теста	Негативный
Объект тестирования	updateObject(int objectId, String name, String description, String category)
Описание теста	Проверяет изменение данных объекта системы с пустым полем 'name'
Входные данные	objectId=3, name='', category = 'musicColumn', description='music column JBL Charge 3, подключение по Bluetooth, порт AUX'
Ожидаемый результат	Возвращает значение False (пустое поле 'name')

№ теста	B16
Тип теста	Негативный
Объект тестирования	updateObject(int objectId, String name, String description, String category)
Описание теста	Проверяет изменение данных объекта системы с пустым полем 'description'
Входные данные	objectId=3, name='JBL Charge 3', category = 'musicColumn', description=''
Ожидаемый результат	Возвращает значение False (пустое поле 'description')

3.2.4 Метод newRent()

№ теста	B17
Тип теста	Позитивный
Объект тестирования	newRent(String FIORental, String dateRental, String dateReturn, int objectId)
Описание теста	Проверяет добавление записи об аренде в систему
Входные данные	FIORental='Максим Дмитриевич', dateRental='2022-01-10', dateReturn='2022-03-10', objectId=3
Ожидаемый результат	Сохраняет данные об аренде в базу данных. Возвращает значение True

№ теста	B18
Тип теста	Негативный
Объект тестирования	newRent(String FIORental, String dateRental, String dateReturn, int objectId)
Описание теста	Проверяет добавление записи об аренде в систему с пустым полем 'name'
Входные данные	FIORental="", dateRental='2022-01-10', dateReturn='2022-03-10', objectId=3
Ожидаемый результат	Возвращает значение False (пустое поле 'name')

3.2.5 Метод newObject()

№ теста	B19
Тип теста	Позитивный
Объект тестирования	newObject(int objectId, String name, String description, String category)
Описание теста	Проверяет создание нового объекта системы
Входные данные	objectId=4, name='JBL Charge 3', category = 'musicColumn', description='music column JBL Charge 3, подключение по Bluetooth, порт AUX'
Ожидаемый результат	Добавляет новый объект с заданной информацией в базу данных. Возвращает значение True

№ теста	B20
Тип теста	Негативный
Объект тестирования	<code>newObject(int objectId, String name, String description, String category)</code>
Описание теста	Проверяет создание нового объекта системы с пустым полем 'name'
Входные данные	<code>objectId=4, name="", category = 'musicColumn', description='music column JBL Charge 3, подключение по Bluetooth, порт AUX'</code>
Ожидаемый результат	Возвращает значение False (пустое поле 'name')

№ теста	B21
Тип теста	Негативный
Объект тестирования	<code>newObject(int objectId, String name, String description, String category)</code>
Описание теста	Проверяет создание нового объекта системы с пустым полем 'description'
Входные данные	<code>objectId=4, name='JBL Charge 3', category = 'musicColumn', description=""</code>
Ожидаемый результат	Возвращает значение False (пустое поле 'description')

3.2.6 Метод `findObject()`

№ теста	B22
Тип теста	Позитивный
Объект тестирования	<code>findObject(int objectId)</code>
Описание теста	Проверяет поиск объекта системы
Входные данные	<code>int RfidId = "344642296637"</code>
Ожидаемый результат	Возвращает всю информацию об объекте с меткой RfidId, где <code>Object.id[] = [1]</code>

№ теста	B23
Тип теста	Негативный
Объект тестирования	findObject(int objectId)
Описание теста	Проверяет поиск несуществующего объекта системы
Входные данные	objectId = "0000000000000000"
Ожидаемый результат	Возвращает значение False

3.2.7 Метод infoRental()

№ теста	B24
Тип теста	Позитивный
Объект тестирования	infoRental(int objectId)
Описание теста	Проверяет получение информации об аренде объекта системы
Входные данные	objectId = 1, идентификатор, аренда на который есть в БД
Ожидаемый результат	Возвращает информацию об аренде объекта с меткой objectId, где Rent.id[] = [4]

№ теста	B25
Тип теста	Негативный
Объект тестирования	infoRental(int objectId)
Описание теста	Проверяет получение информации об аренде несуществующего объекта системы
Входные данные	objectId = 7, идентификатор, аренды на который нет в БД
Ожидаемый результат	Возвращает значение False

3.2.8 Метод returnObject()

№ теста	B26
Тип теста	Позитивный
Объект тестирования	returnObject(int objectId)
Описание теста	Проверяет изменение статуса объекта
Входные данные	objectId = 3 идентификатор объекта, который есть в БД и арендован
Ожидаемый результат	Возвращает значение True
Соответствие ФТ	Тест удовлетворяет ФТ-7

№ теста	B27
Тип теста	Негативный
Объект тестирования	returnObject(int objectId)
Описание теста	Проверяет изменение статуса несуществующего объекта
Входные данные	objectId = 7, идентификатор объекта, которого нет в БД
Ожидаемый результат	Возвращает значение False

3.2.9 Метод rentalObject()

№ теста	B28
Тип теста	Позитивный
Объект тестирования	rentalObject(int objectId)
Описание теста	Проверяет изменение статуса объекта
Входные данные	objectId = 1, идентификатор объекта, который есть в БД
Ожидаемый результат	Возвращает значение True

№ теста	B29
Тип теста	Негативный
Объект тестирования	rentalObject(int objectId)
Описание теста	Проверяет изменение статуса несуществующего объекта
Входные данные	objectId = 7, идентификатор объекта, которого нет в БД
Ожидаемый результат	Возвращает значение False

3.2.10 Метод rfidRead()

№ теста	B30
Тип теста	Позитивный
Объект тестирования	rfidRead()
Описание теста	Проверяет получение id метки приложенной к считывателю, вызывает функцию con(String request)
Входные данные	String request = "RFID _R EAD"
Ожидаемый результат	Возвращает String "344642296637"

№ теста	B31
Тип теста	Негативный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку объектов со статусом "abcdefg"
Входные данные	status = 'abcdefg', category = [], name = '' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает пустой список

№ теста	B32
Тип теста	Негативный
Объект тестирования	filterObject(Bool status, String[] category, String name)
Описание теста	Проверяет сортировку объектов с категорией "abcdefg"
Входные данные	status = 'both', category = ["abcdefg"], name = '' (в базе данных имеются объекты только перечисленных в массиве category категорий)
Ожидаемый результат	Возвращает пустой список

3.3 Интеграционное тестирование

3.3.1 Методы классов Logic и Activiti

№ теста	I1
Тип теста	Позитивный
Описание теста	Проверяет вызов метода sort() класса Logic из метода onClick() класса MainActivity
Входные данные	Массив выбранных статусов <ArrayList> statusCheckboxes = [TRUE, FALSE], массив выбранных категорий <ArrayList> categoryCheckboxes = ['Wires'], строка имени объекта поиска TextInputLayout nameView = ['']
Ожидаемый результат	вызывается метод sort() из класса Logic, обновляется список объектов в каталоге, где Object.id[] = [4,5,6]

№ теста	I2
Тип теста	Позитивный
Описание теста	Проверяет вызов метода deleteObject() класса Logic из метода onClick() класса ObjectActivity
Входные данные	Идентификатор данного объекта id = 2
Ожидаемый результат	вызывает метод deleteObject() класса Logic

№ теста	I3
Тип теста	Позитивный
Описание теста	Проверяет вызов метода getObjectById() класса Logic из метода onCreate() класса ObjectActivity
Входные данные	Идентификатор данного объекта id = 4
Ожидаемый результат	вызывает метод getObjectById() класса Logic, возвращающий данные об объекте, где Object.category = "Wires"

№ теста	I4
Тип теста	Позитивный
Описание теста	Проверяет вызов метода returnObject() класса Logic из метода onClick() класса ObjectActivity
Входные данные	Идентификатор данного объекта id = 3
Ожидаемый результат	вызывает метод returnObject() класса Logic

№ теста	I5
Тип теста	Позитивный
Описание теста	Проверяет вызов метода saveEdit() класса Logic из метода onClick() класса FormActivity
Входные данные	ObjectClass(id, name, description, status, category), где id = 1, name = "JBL Charge 4 description = "Music status = FALSE, category = "musicColumn"
Ожидаемый результат	вызывает метод saveEdit() класса Logic

№ теста	I6
Тип теста	Позитивный
Описание теста	Проверяет вызов метода newRental() класса Logic из метода onClick() класса FormActivity
Входные данные	RentalClass(name, startDate, endDate, id), где name = "Ибрагим Васильевич Зурин startDate = "10.09.2022 endDate = "12.09.2022 id = 2
Ожидаемый результат	вызывает метод newRental() класса Logic

№ теста	I7
Тип теста	Позитивный
Описание теста	Проверяет вызов метода rentalObject() класса Logic из метода OnClick() класса FormActivity
Входные данные	Идентификатор данного объекта id = 2
Ожидаемый результат	вызывает метод rentalObject() класса Logic

№ теста	I8
Тип теста	Позитивный
Описание теста	Проверяет вызов метода addObject() класса Logic из метода OnClick() класса FormActivity
Входные данные	ObjectClass(id, name, description, status, category), где id = 7, name = "JBL PRX description = "Professional Column status = False, category = "musicColumn"
Ожидаемый результат	вызывает метод addObject() класса Logic

3.4 Аттестационное тестирование

№ теста	A1
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет считывание RFID-метки существующего объекта
Действие:	Пользователь подносит зарегистрированную RFID-метку к считывателю
Ожидаемый результат:	Показывается информация, привязанная к метке, и кнопки “Удалить”, “Редактировать”, “Взять в аренду”/”Вернуть на склад”

№ теста	A2
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет считывание RFID-метки несуществующего объекта
Действие:	Пользователь подносит незарегистрированную RFID-метку к считывателю
Ожидаемый результат:	Показывается информация о том, что к метке не привязан никакой объект и кнопка “Добавить объект”

№ теста	A3
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение, приложил метку несуществующего объекта к считывателю и находится на странице добавления объекта
Описание теста	Тест проверяет привязывание объекта к RFID-метке
Действие	Пользователь заполняет поля “Название”, “Категория”, “Описание” не пустыми строками и нажимает кнопку “Добавить”
Ожидаемый результат	Пользователю отображается окно об успешном добавлении. Объект добавляется в базу данных

№ теста	A4
Тип теста	Негативный
Начальное состояние	Пользователь открыл приложение, приложил метку несуществующего объекта к считывателю и находится на странице добавления объекта
Описание теста	Тест проверяет привязывание объекта к RFID-метке с пустым полем 'название'
Действие	Пользователь заполняет поля "Название", "Категория", "Описание" и нажимает кнопку "Добавить". Поле "Название" - пустое
Ожидаемый результат	Пользователю выводится сообщение о том, что введены неверные данные

№ теста	A5
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение, нажал на область объекта, нажал на кнопку 'редактировать' и находится на странице изменения данных объекта
Описание теста	Тест проверяет изменение информации об объекте
Действие	Пользователь заполняет поля "Название", "Категория", "Описание" не пустыми строками и нажимает на кнопку "Сохранить изменения"
Ожидаемый результат	Пользователю выводится сообщение о том, что данные успешно изменены. Информация об объекте изменяется в базе данных

№ теста	A6
Тип теста	Негативный
Начальное состояние	Пользователь открыл приложение, нажал на область объекта, нажал на кнопку 'редактировать' и находится на странице изменения данных объекта
Описание теста	Тест проверяет изменение информации об объекте с пустым полем "Название"
Действие	Пользователь заполняет поля "Название", "Категория", "Описание" оставляя поле "Название" пустым и нажимает на кнопку "Сохранить изменения"
Ожидаемый результат	пользователю выводится сообщение "введены некорректные данные"

№ теста	A7
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение, нажал на область объекта, и находится на странице объекта
Описание теста	Тест проверяет удаление объекта
Действие	Пользователь нажимает на кнопку "Удалить"
Ожидаемый результат	Объект удаляется из базы данных. Пользователю выводится сообщение о том, что объект успешно удален

№ теста	A8
Тип теста	Позитивный
Начальное состояние	Пользователь находится в каталоге приложений своего мобильного устройства
Описание теста	Тест проверяет запуск приложения и отображение каталога объектов
Действие	Пользователь нажимает на иконку приложения "RFID-mobile"
Ожидаемый результат	Приложение запускается, открывается экран с каталогом объектов

№ теста	A9
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение, и находится на главной странице
Описание теста	Тест проверяет просмотр подробной информации по конкретному объекту
Действие	Пользователь нажимает на поле объекта
Ожидаемый результат	Открывается страница с информацией о нажатом объекте

№ теста	A10
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение, приложил RFID-метку и находится на странице объекта, ему показаны кнопки “Удалить”, “Редактировать”, “Взять в аренду”/”Вернуть на склад”
Описание теста	Тест проверяет изменение состояния объекта
Действие	Пользователь нажимает на кнопку “Взять в аренду”/”Вернуть на склад”
Ожидаемый результат	Состояние объекта изменяется в базе данных и выводится сообщение об изменении состояния объекта

№ теста	A11
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет фильтрацию объектов по статусу
Действие	Пользователь нажимает на кнопку фильтрации, выбирает статус объекта (выбирает один или несколько вариантов) и нажимает кнопку “Найти”
Ожидаемый результат	Появляется каталог с объектами, удовлетворяющими заданному фильтру.

№ теста	A12
Тип теста	Негативный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет фильтрацию объектов по статусу
Действие	Пользователь нажимает на кнопку фильтрации, оставляет пустыми все варианты статуса и нажимает кнопку “Найти”
Ожидаемый результат	Появляется каталог, со всеми объектами в базе данных.

№ теста	A13
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет фильтрацию объектов по категории
Действие	Пользователь нажимает на кнопку фильтрации, выбирает категории объекта (выбирает один или несколько вариантов) и нажимает кнопку “Найти”
Ожидаемый результат	Появляется каталог с объектами, удовлетворяющими заданному фильтру

№ теста	A14
Тип теста	Негативный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет фильтрацию объектов по категории, когда пользователь оставляет пустыми все варианты фильтра.
Действие	Пользователь нажимает на кнопку фильтрации, оставляет пустыми все варианты и нажимает кнопку “Найти”
Ожидаемый результат	Появляется каталог, со всеми объектами в базе данных.

№ теста	A15
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет фильтрацию объектов по названию
Действие	Пользователь нажимает на кнопку фильтрации, вводит название объектов, существующих в базе данных, и нажимает кнопку “Найти”
Ожидаемый результат	Появляется каталог с объектами, удовлетворяющими заданному фильтру

№ теста	A16
Тип теста	Негативный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет фильтрацию по названию объектов, которых нет в базе данных
Действие	Пользователь нажимает на кнопку фильтрации, вводит название объектов, которых нет в базе данных, и нажимает кнопку “Найти”
Ожидаемый результат	Появляется пустой каталог

№ теста	A17
Тип теста	Негативный
Начальное состояние	Пользователь открыл приложение и находится на главной странице
Описание теста	Тест проверяет фильтрацию по названию объектов, когда пользователь оставляет пустым поле ввода
Действие	Пользователь нажимает на кнопку фильтрации, оставляет пустым поле ввода и нажимает кнопку “Найти”
Ожидаемый результат	Появляется каталог, со всеми объектами в базе данных

№ теста	A18
Тип теста	Позитивный
Начальное состояние	Пользователь открыл приложение, приложил RFID-метку и находится на странице объекта, ему показаны кнопки “Удалить”, “Редактировать”, “Взять в аренду”/”Вернуть на склад”
Описание теста	Тест проверяет добавление данных об аренде объекта
Действие	Пользователь нажимает на кнопку “Взять в аренду”, вносит не пустые данные об аренде и нажимает на кнопку “Арендовать”
Ожидаемый результат	Данные об аренде вносятся в БД и открывается сообщение об успешной аренде

№ теста	A19
Тип теста	Негативный
Начальное состояние	Пользователь открыл приложение, приложил RFID-метку и находится на странице объекта, ему показаны кнопки “Удалить”, “Редактировать”, “Взять в аренду”/”Вернуть на склад”
Описание теста	Тест проверяет добавление данных об аренде объекта
Действие	Пользователь нажимает на кнопку “Взять в аренду”, вносит данные об аренде , где дата возврата раньше даты взятия в аренду и нажимает на кнопку “Арендовать”
Ожидаемый результат	Открывается сообщение о том, что пользователь ввел некорректные данные

3.5 Тестирование безопасности

№ теста	S1
Тип теста	Позитивный
Объект тестирования	tryLoginIn(String login, String Hashpass)
Описание теста	Проверяет вход пользователя в систему
Входные данные	String login = "admin String pass = DigestUtils.md5Hex("admin")
Ожидаемый результат	Возвращает значение True

№ теста	S2
Тип теста	Негативный
Объект тестирования	tryLoginIn(String login, String Hashpass)
Описание теста	Проверяет вход пользователя в систему без использования хеширования
Входные данные	String login = "admin String Hashpass = "admin"
Ожидаемый результат	Возвращает значение False

№ теста	S3
Тип теста	Позитивный
Объект тестирования	isLogIn(String page)
Описание теста	Проверяет получение доступа к редактированию данных пользователем, который зарегистрирован в системе и имеет права администратора.
Входные данные	String page = "editObject String login = "admin String Hashpass = DigestUtils.md5Hex("admin")
Ожидаемый результат	Возвращает значение True

№ теста	S4
Тип теста	Негативный
Объект тестирования	isLogIn(String page)
Описание теста	Проверяет получение доступа к редактированию данных пользователем, который зарегистрирован в системе и имеет права менеджера.
Входные данные	String page = "editObject String login = "manager String Hashpass = DigestUtils.md5Hex("manager")
Ожидаемый результат	Возвращает значение False

№ теста	S5
Тип теста	Позитивный
Объект тестирования	isLogIn(String page)
Описание теста	Проверяет получение доступа к просмотру данных пользователем, который зарегистрирован в системе и имеет права менеджера.
Входные данные	String page = "mainPage String login = "manager String Hashpass = DigestUtils.md5Hex("manager")
Ожидаемый результат	Возвращает значение True

№ теста	S6
Тип теста	Позитивный
Объект тестирования	isLogIn(String page)
Описание теста	Проверяет получение доступа к просмотру данных пользователем, который зарегистрирован в системе и имеет права администратора.
Входные данные	String page = "mainPage String login = "admin String Hashpass = DigestUtils.md5Hex("admin")
Ожидаемый результат	Возвращает значение True

№ теста	S7
Тип теста	Негативный
Объект тестирования	tryLoginIn(String login, String Hashpass)
Описание теста	Проверяет попытку инъекции в базу данных при входе
Входные данные	String login = "admin ИЛИ manager String Hashpass = DigestUtils.md5Hex("manager")
Ожидаемый результат	Возвращает значение False

3.6 Примеры реализации тестов

```
public class ExampleUnitTest {
    @Test
    public void rfidReadPositive()
    {
        String testMarkId = "344642296637";
        String answer = Logic.scanRfid()[1];
        assertEquals(testMarkId, answer);
    }

    @Test
    public void rentalObjectPositive()
    {
        String testObjectId = "344642296637";
        Boolean answer = Logic.rentalObject(testObjectId);
        assertEquals(Boolean.TRUE, answer);
    }
}
```

Рис. 5 – Пример кода тестов чтения идентификатора метки и аренды объекта.

```

@Test
public void returnObjectPositive()
{
    String testObjectId = "344642296637";
    Boolean answer = Logic.returnObject(testObjectId);
    assertEquals(Boolean.TRUE, answer);
}

@Test
public void newObjectNegative()
{
    ObjectClass testObject = new ObjectClass(id: "344642296622", name: "", description: "description", Boolean.FALSE, category: "music_column");
    Boolean answer = Logic.addObject(testObject);
    assertEquals(Boolean.FALSE, answer);
}

```

Рис. 6 – Пример кода тестов возврата объекта на склад и попытки создания объекта без имени.

3.7 Журнал тестирования

№ Теста	Дата	Результат	№ Ошибки
A1	15.11.2022	Пройден	-
A2	15.11.2022	Пройден	-
A3	15.11.2022	Пройден	-
A4	15.11.2022	Пройден	-
A5	15.11.2022	Пройден	-
A6	15.11.2022	Пройден	-
A7	15.11.2022	Пройден	-
A8	15.11.2022	Пройден	-
A9	15.11.2022	Пройден	-
A10	15.11.2022	Не пройден	E1
A11	15.11.2022	Пройден	-
A12	15.11.2022	Пройден	-
A13	15.11.2022	Пройден	-
A14	15.11.2022	Пройден	-
A15	15.11.2022	Пройден	-
A16	15.11.2022	Не пройден	E2
A17	15.11.2022	Пройден	-
A18	15.11.2022	Пройден	-
A19	15.11.2022	Пройден	-

№ Теста	Дата	Результат	№ Ошибки
B1	22.11.2022	Пройден	-
B2	22.11.2022	Пройден	-
B3	22.11.2022	Пройден	-
B4	22.11.2022	Пройден	-
B5	22.11.2022	Пройден	-
B6	22.11.2022	Пройден	-
B7	22.11.2022	Пройден	-
B8	22.11.2022	Пройден	-
B9	22.11.2022	Пройден	-
B10	22.11.2022	Пройден	-
B11	22.11.2022	Пройден	-
B12	22.11.2022	Пройден	-
B13	22.11.2022	Пройден	-
B14	22.11.2022	Пройден	-
B15	22.11.2022	Пройден	-
B16	22.11.2022	Не пройден	E3
B17	22.11.2022	Не пройден	E4
B18	22.11.2022	Пройден	-
B19	22.11.2022	Не пройден	E5
B20	22.11.2022	Пройден	-
B21	22.11.2022	Пройден	-
B22	22.11.2022	Пройден	-
B23	22.11.2022	Пройден	-
B24	22.11.2022	Пройден	-
B25	22.11.2022	Пройден	-
B26	22.11.2022	Пройден	-
B27	22.11.2022	Пройден	-
B28	22.11.2022	Пройден	-
B29	22.11.2022	Пройден	-
B30	22.11.2022	Пройден	-
B31	22.11.2022	Пройден	-
B32	22.11.2022	Пройден	-

№ Теста	Дата	Результат	№ Ошибки
I1	22.11.2022	Пройден	-
I2	22.11.2022	Пройден	-
I3	22.11.2022	Пройден	-
I4	22.11.2022	Пройден	-
I5	22.11.2022	Пройден	-
I6	22.11.2022	Пройден	-
I7	22.11.2022	Пройден	-
I8	22.11.2022	Пройден	-

№ Теста	Дата	Результат	№ Ошибки
S1	22.11.2022	Пройден	-
S2	22.11.2022	Пройден	-
S3	22.11.2022	Пройден	-
S4	22.11.2022	Пройден	-
S5	22.11.2022	Пройден	-
S6	22.11.2022	Пройден	-
S7	22.11.2022	Пройден	-

3.8 Найденные ошибки

№ ошибки	E1
№ теста	A10
Описание	После взятия объекта в аренду выводится сообщение об ошибке
Дата обнаружения	20.10.2022
Исправлено	Да

№ ошибки	E2
№ теста	A16
Описание	Поиск объекта по названию крашится, если вводить несуществующее в БД слово или название
Дата обнаружения	20.10.2022
Исправлено	Да

№ ошибки	E3
№ теста	B16
Описание	База данных не выдаёт исключение, когда обновляется объект без названия (При создании таблицы свойство not null)
Дата обнаружения	20.10.2022
Исправлено	Да

№ ошибки	E4
№ теста	B17
Описание	При обновлении объекта база данных не выдаёт исключение, когда обновленный объект заносится без описания(при создании таблицы свойство not null)
Дата обнаружения	20.10.2022
Исправлено	Да

№ ошибки	Е5
№ теста	В19
Описание	При вводе в базу данных нового объекта, база данных не выдаёт исключение при не указанном названии объекта (при создании таблицы свойство not null)
Дата обнаружения	20.10.2022
Исправлено	Да

3.9 Покрывтие кода тестами

Покрывтие кода будет рассчитано по формуле:

$$\frac{[\text{Количество строк кода, покрывтое тестами}]}{[\text{Общее количество строк кода}]} * 100\%$$

В результате получены значения:

- Всего строк кода: 682
- Строк кода покрывтых тестами: 589
- Процент покрывтия: 86,36%

3.10 Результаты тестирования

По результатам тестирования было выявлено 5 ошибок, которые удалось исправить. Остальные функции системы протестированы и исправно работают согласно функциональным требованиям.