

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО
ОБЕСПЕЧЕНИЯ

Отчёт по курсу
«Верификация ПО»

Выполнил:
студент группы 22608 В. Н.
Хаймоев
Преподаватель:
к.ф-м.н., доцент К. А. Кулаков

Петрозаводск
2016

Оглавление

Согласование проекта	3
Описание функциональности	3
Архитектура	4
План тестирования	6
Описание входных данных для тестирования	8
Детальный план тестов	12
Блочное тестирование.....	12
Интеграционное тестирование:.....	18
Стресс тестирование.....	20
Реализации тестов	21
Журнал тестирования:	22
Журнал найденных ошибок	25
Методы покрытия	28

Согласование проекта

Проект представляет собой мобильное приложение для автоматизации складского учета.

Мобильное приложение синхронизируется с основным программой 1С:Управление торговлей и поддерживает основные операции для складского учета: комплектация, консолидация. Мобильное приложение предназначено для ОС Android. Тестироваться мобильное приложение будем совместно с основным приложением для проверки синхронизации.

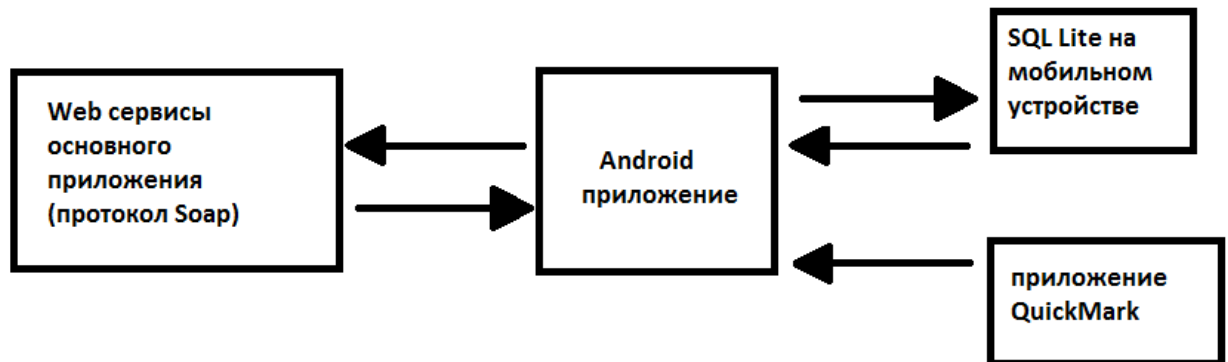
Описание функциональности

1. Проверка авторизации в приложение ввод логина и пароля, также проверка авторизации с использованием QR кода.
2. Получение задач «Комплектация» и «Консолидация» от основного приложения через web сервис.
3. Сохранение, обновление и удаление документов в базе данных SQL Lite в мобильном приложении.
4. Сканирование наклеек товаров используя внешнее приложение QuickMark и отметка нахождения в документе.
5. Синхронизация с Web сервисом документа «Комплектация» в статусе «Проблема» при этом задач остается в приложении и синхронизируется с основным через web сервис.
6. Синхронизация с Web сервисом документа «Консолидация» в статусе «Проблема» при этом задач остается в приложении и синхронизируется с основным через web сервис.
7. Синхронизация с Web сервисом документа «Комплектация» в статусе «Исполнена» при этом задача синхронизируется с основным через web сервис и удаляется из приложения.
8. Синхронизация с Web сервисом документа «Консолидация» в статусе «Исполнена» при этом задача синхронизируется с основным через web сервис и удаляется из приложения.
9. Отправка статуса задачи «Исполняется» в Web сервис задачи «Комплектации» и «Консолидации».

Ограничения

Вышеперечисленные функции должны быть доступны для мобильных телефонов с операционной системой Android версии 4.4.0 и старше.

Архитектура



1. Архитектура

При создании приложения использовалась среда разработки Android Studio.

Android приложение получает данные от Web сервиса основного приложения с использованием формата xml по протоколу soap. Приложение разбирает полученные задачи, записывает их во внутреннюю базу данных SQL lite. При работе с документами используется стороннее приложение QuickMark для считывания QR кодов, которые после передаются обратно строку для получения информации о товаре.

После окончания работы с документами производится синхронизация с Web сервисами основного приложения с помощью формата xml, которое формирует само андроид приложение на основе данных документа.

Перечень классов и модулей

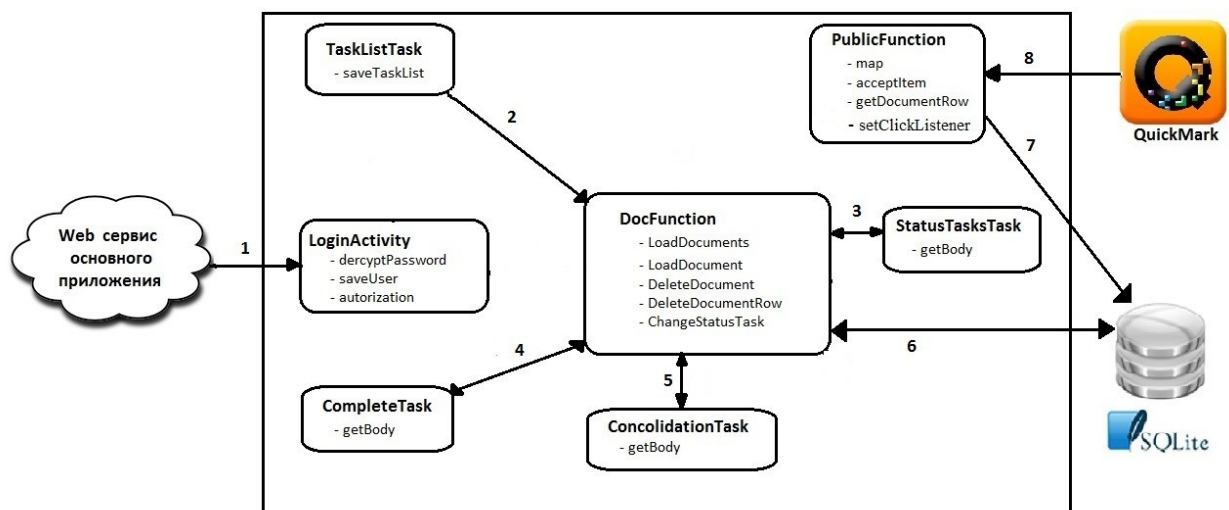


рис. 2. Структура классов приложения

Классы:

1. Класс: LoginActivity – осуществляющий пользовательскую Авторизацию с целью получения права доступа к Web сервису.

- 1.1. decryptPassword – расшифровка пароля
- 1.2. saveUser – сохранение пользователя
- 1.3. authorization – процедура авторизации с основным приложением
2. Класс: PublicFunctions – класс, реализующий функции, используемые во всем приложении.
Функции:
 - 2.1. Map – получение параметров из http ссылки
 - 2.2. acceptItem – отметка наличия товара в строке
(делает отметку наличия товара в строке в базе данных)
 - 2.3. getDocumentRow – поиск строки товара для отметки
 - 2.4. setClickListener – запуск приложения QuickScan
3. Класс: StatusTasksTask получение задач от основного приложения
 - 3.1. getBody – формирование xml файла для синхронизации статуса
исполняется с Web сервисом
4. Класс: TaskListTask
Функции:
 - 4.1. saveTaskList – чтение xml документа и сохранение документов в базу
данных
5. Класс: CompleteTask
 - 5.1. getBody – формирование xml документа для синхронизации задачи
«Комплектаии» с web сервисом основного приложения.
6. Класс: ConsolidationTask
getBody – формирование xml документа для синхронизации задачи
«Консолидации» с web сервисом основного приложения.
7. Класс: DocFunctions – класс, реализующий функции работы с
документами
Функции:
 - 7.1. loadDocuments – функция загрузки документов в список задач из базы
данных
 - 7.2. loadDocument – функция получения выбранного документа в форму
документа из базы данных
 - 7.3. deleteDocument – удаление документа из базы
 - 7.4. deleteDocumentRow – удаление строк таблицы документа из базы
 - 7.5. changeStatusTask – смена статуса документа в базе данных

Меню приложения:

1. Страница входа в приложение.
2. Страница списка задач кладовщика.
Кнопки:
 - 2.1. Загрузка задач
 - 2.2. Принятие задачи к исполнению
3. Форма документа
 - 3.1. Кнопка «Меню» содержит
 - 3.1.1. Проблема
 - 3.1.2. Исполнена

4. Кнопка вызова камеры для сканирования наклейки

Соответствие меню и функциональности приложения:

1. – 1.
- 2.1. – 2,3.
- 2.2. – 9.
- 3.1.1. – 5,6.
- 3.1.2. – 6,7.
4. – 4.

План тестирования

1. Блочное тестирование

Тестированию подлежат методы следующих вышеописанных классов:
Классы:

1. Класс: LoginActivity
исключая метод authorization – не имеет логики, стандартные механизмы.
2. Класс: PublicFunctions
3. Класс: StatusTasksTask
4. Класс: TaskListTask
5. Класс: CompleteTask
6. Класс: ConsolidationTask
7. Класс: DocFunctions

2. Интеграционное тестирование

На рис. 3 стрелками обозначены направления передачи данных между модулями. Цифры над стрелками показывают последовательность интеграционного тестирования модулей.

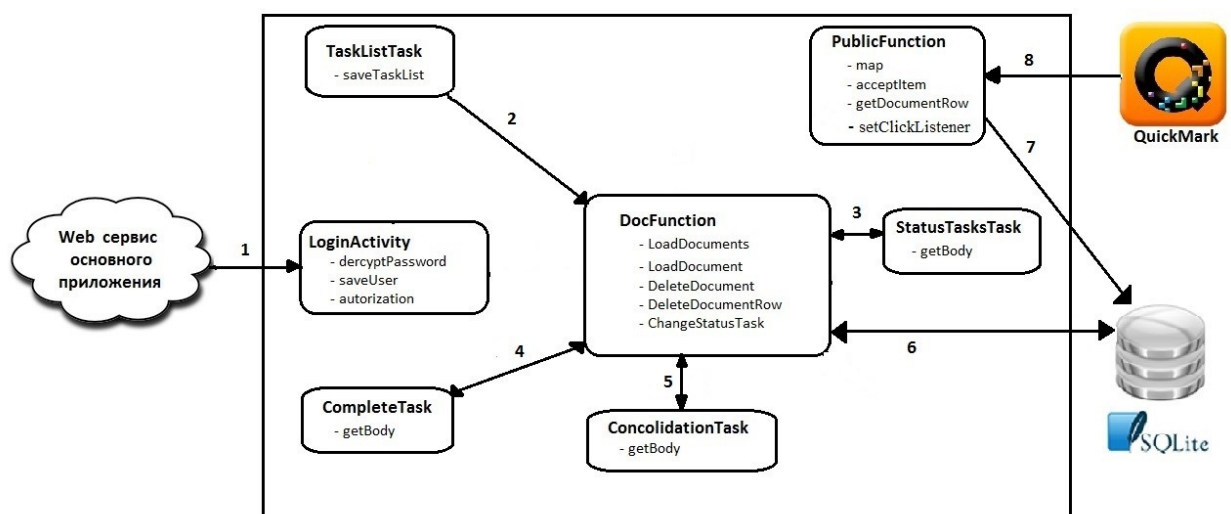


рис. 3 Архитектура приложения

Интеграционные тесты проверяют взаимодействие классов:

1. Web сервис и LoginActivity
Взаимодействие не тестируется, используются стандартные механизмы авторизации.
2. TaskListTask и DocFunction
функции saveTaskList и LoadDocuments – выполняются последовательно.
3. StatusTaskTasks и DocFunction
функции getBody и changeStatusTask выполняются последовательно.
4. ConcolidationTask и DocFunction
функция getBody и changeStatusTask выполняются последовательно
функции getBody, changeStatusTask, deleteDocumentRow и deleteDocument выполняются последовательно.
5. CompleteTask и DocFunction
функция getBody, changeStatusTask выполняются последовательно.
функции getBody changeStatusTask, deleteDocumentRow и deleteDocument выполняются последовательно.
6. Взаимодействие класса DocFunction с базой данных производится с использованием стандартных механизмов.
7. Взаимодействие класса PublicFunction с базой данных производится с использованием стандартных механизмов.
8. Взаимодействие между PublicFunction с QuickMark производится с использованием стандартных механизмов.

3. Нагруженное тестирование

К стресс тестам данного приложения можно отнести проверку поведения приложения при больших объемах данных.

1. Увеличение количества задач
2. Увеличение количества товаров в задаче

Обычный объем данных:

1. Малый склад – до 5 задач, до 10 штук товара в задаче.
2. Средний склад – до 30 задач, до 50 штук товара в задаче.
3. Большой склад – от 30 задач, от 50 штук товара в задаче.

Стрестесты проводятся для модулей:

Класс: TaskListTask

функция – saveTaskList

Класс: FunctionDocument

функция – LoadDocuments

4. Аттестационное тестирование

Используемые окружение и инструменты: мобильный телефон с ОС Android, Web сервис для синхронизации с основным приложением, приложение QuickMark.

Проверка основных функций приложения:

1. Проверка авторизации в приложение ввод логина и пароля, также проверка авторизации с использованием QR кода.
2. Получение задач «Комплектация» и «Консолидация» от основного приложения через web сервис.
3. Сохранение, обновление и удаление документов в базе данных SQL Lite в мобильном приложении.
4. Сканирование наклеек товаров используя внешнее приложение QuickMark и отметка нахождения в документе.
5. Синхронизация с Web сервисом документа «Комплектация» в статусе «Проблема» при этом задач остается в приложении и синхронизируется с основным через web сервис.
6. Синхронизация с Web сервисом документа «Консолидация» в статусе «Проблема» при этом задач остается в приложении и синхронизируется с основным через web сервис.
7. Синхронизация с Web сервисом документа «Комплектация» в статусе «Исполнена» при этом задача синхронизируется с основным через web сервис и удаляется из приложения.
8. Синхронизация с Web сервисом документа «Консолидация» в статусе «Исполнена» при этом задача синхронизируется с основным через web сервис и удаляется из приложения.
9. Отправка статуса задачи «Исполняется» в Web сервис задачи «Комплектации» и «Консолидации».

Описание входных данных для тестирования

1. xml файл списка задач кладовщика

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header/>
  <soap:Body>
    <m:TaskListResponse xmlns:m="http://www.w3.org/2001/XMLSchema">
      <m:return xsi:type="m:string"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</m:return>
      <m:List xsi:type="m:string"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><![CDATA[<?xml
version="1.0" encoding="windows-1251"?>
<ListTask Version="14">
  <Task StatusTask="Ожидает исполнения" View="Консолидация"
NumberTask="17640">
    <ListProduct>
      <Number>1</Number>
      <Article>128130</Article>
      <NameProduct>Микрофон Audio Technica
АТМ75сW/Микрофон</NameProduct>
```



```
<Unit>шт</Unit>
<SumProduct>1</SumProduct>
<Zakaz>788042</Zakaz>
<Fact>Нет</Fact>
<IdSticker>145861</IdSticker>
</ListProduct>
<ListProduct>
  <Number>1</Number>
  <Article>120767</Article>
  <NameProduct>USB кабель Prolink PB458-0150 (USB - micro
USB 3.0 (AM-BM), 1,5м.)</NameProduct>
  <Unit>шт</Unit>
  <SumProduct>1</SumProduct>
  <Zakaz>789776</Zakaz>
  <Fact>Нет</Fact>
  <IdSticker>146105</IdSticker>
</ListProduct>
</Task>
<Task StatusTask="Ожидает исполнения" View="Комплектация"
NumberTask="17643" Comment="" Date="2016-11-30" Driver="">
  <ListProduct>
    <Number>1</Number>
    <Article>135174</Article>
    <NameProduct>Аксессуар для винила Michell Engineering
Record Clamp Standart</NameProduct>
    <Unit>шт</Unit>
    <SumProduct>1</SumProduct>
    <Zakaz>788995</Zakaz>
    <Fact>Нет</Fact>

  </ListProduct>
  <ListProduct>
    <Number>1</Number>
    <Article>128114</Article>
    <NameProduct>Микрофон Arthur Forty AF-905s
PSC</NameProduct>
    <Unit>шт</Unit>
    <SumProduct>1</SumProduct>
    <Zakaz>786946</Zakaz>
    <Fact>Нет</Fact>
    <IdSticker/>
  </ListProduct>
  <ListProduct>
    <Number>2</Number>
    <Article>128114</Article>
```

```

        <NameProduct>Микрофон Arthur Forty AF-905s
PSC</NameProduct>
        <Unit>шт</Unit>
        <SumProduct>1</SumProduct>
        <Zakaz>786946</Zakaz>
        <Fact>Нет</Fact>
        <IdSticker/>
    </ListProduct>
</Task>
</ListTask>]]></m:List>
    </m:TaskListResponse>
</soap:Body>
</soap:Envelope>

```

2. описание xml файла для синхронизации с web сервисом при принятии задачи к исполнению

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <soap:Header/>
    <soap:Body>
        <xs>StatusTasks>
            <xs>StatusTask>Статус задачи</xs>StatusTask>
            <xs:Number>Номер задачи</xs:Number>
        </xs>StatusTasks>
    </soap:Body>
</soap:Envelope>

```

3. описание xml файла для синхронизации с web сервисом при исполнении задачи комплектации

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <soapenv:Header/>
    <soapenv:Body>
        <xs:Complete>
            <xs:List>
                <![CDATA[<?xml version="1.0" encoding="windows-1251"?>
                <Task>
                    <ListProduct>
                        <Number>2</Number>
                        <Article>128114</Article>
                        <SumProduct>1</SumProduct>
                        <Fact>Есть</Fact>
                        <IdSticker>225583</IdSticker>
                    </ListProduct>
                </Task>

```

```

]]>
</xs:List>
<xs:Number>17643</xs:Number>
<xs:View>Комплектация</xs:View>
<xs:StatusTask>Исполнена</xs:StatusTask>
</xs:Complete>
</soapenv:Body>
</soapenv:Envelope>

```

4. описание xml файла для синхронизации с web сервисом при выполнении задачи консолидации

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<soapenv:Header/>
<soapenv:Body>
<xs:Consolidation>
<xs:List>
<![CDATA[<?xml version="1.0" encoding="windows-1251"?>
<Task>
<ListProduct>
<Number>1</Number>
<Article>128130</Article>
<Status>Есть</Status>
<IdSticker>145861</IdSticker>
<Fact>Есть</Fact>
</ListProduct>
]]>
</xs:List>
<xs:Number>17640</xs:Number>
<xs:View>Консолидация</xs:View>
<xs:StatusTask>Исполнена</xs:StatusTask>
</xs:Consolidation>
</soapenv:Body>
</soapenv:Envelope>

```

Ограничения данных тегов:

- Number – число идентифицирующее строку в документе (от 0 до 9 999 999)
- Article – артикул, строка (до 8 символов)
- NameProduct – наименование товара, строка (до 300 символов)
- Unit – единица измерения товара, имеет определенный список значений (шт, пара, комплект, м)
- SumProduct – количество товара, число (от 0.00 до 9 999 999)
- Fact – наличие товара, список значений (есть, нет)
- IdSticker – номер наклейки, число (от 0 до 99 999 999)

StatusTask - Статус задачи – список значений (ожидает исполнения, проблема, исполняется)

Number - Номер задачи – число (от 1 до 999 999)

View – вид задачи, список значений (комплектация, консолидация)

Детальный план тестов

Блочное тестирование

Класс: LoginActivity

Функции

decryptPassword – расшифровка пароля (вход. параметры строка с паролем) пароль состоит из цифр, в пароль добавлены сгенерированные цифры, первая, последняя и каждая вторая с 3-й цифры также случайная цифра.

№	Условие	Ожидаемый результат	Тип теста
1	132549243	3594	Общий
2	пустая строка	пустая строка	Общий
3	999999999999	99999	Общий
4	РОРПРРвсдвос	Пустая строка	Негативный

saveUser – сохранение пользователя базу данных в таблицу users_list (вх. параметр строка с логином)

№	Условие	Ожидаемый результат	Тип теста
5	Иванов Иван	Запись в таблице users_list «Иванов Иван»	Общий
6	пустая строка	Отсутствие записи в таблице users_list	Общий
7	Hrfuhreuhuergurehguhruh fjrhefreherhgerghregjregergergb frejherhg284543b542nnb ;;;343 5' “/242359@\$%#^&\$*(&\$#@	Запись в таблице users_list	Общий

Класс: Public Function

Map – получение параметров из http ссылки

<http://qr.pult.ru/?A=138282&P=4&N=145656>

A – номер заказа, P – номер в документе, N – номер наклейки
получение параметров ссылки

Вход. параметры – ссылка.

Выходные – структура параметров и значений

№	Условие	Ожидаемый результат	Тип теста
---	---------	---------------------	-----------

8	http://qr.pult.ru/?A=138282&P=4&N=145656	Структура вида (А – 138282, Р – 4, N - 145656)	Общий
9	пустая строка	пустая структура	Общий
10	рварукакгуукпу	Пустая структура	Негативный
11	http://qr.234/?A=138282&P=4&N=325253	Пустая структура	Негативный

getRow – поиск строки наклейки.

Вход. параметры – id номер документа, номер наклейки, которую нужно найти в таблице «».

Выходные – строка найденной наклейки в документе

id номер документа – берется из базы данных (от 1 до 65535)

номер наклейки – берется из таблицы документа в базе данных (от 1 до 99 999 999)

№	Условие	Ожидаемый результат	Тип теста
12	Номер документа, который есть в базе Номер наклейки, которая есть в документе	Найденная строка товара	Общий
13	Номер документа, который нет в базе	Пустая строка	Негативный
14	Номер документа, который есть в базе Номер наклейки, которой нет в документе	Пустая строка	Общий
15	Номер документа 0, наклейка 0	Пустая строка	Крайние значения
16	Номер документа 65560, наклейка 100 000 000	Пустая строка	Негативный

AcceptedItem – отметка наличия товара в строке.

id строки (от 1 до 65535)

Вход. параметры – id строки таблицы базы данных.

Выходные – измененная строка таблицы базы данных.

В таблице документа n строк.

№	Условие	Ожидаемый результат	Тип теста
---	---------	---------------------	-----------

17	1	значение fact = true в строке	Общий
18	n	значение fact = true в строке	Общий
19	n + 1	ничего не произошло	Негативный
20	oao	ничего не произошло	Негативный

Класс: TaskListTask получение задач от основного приложения

saveTaskList – разбор и сохранение задачи в базе данных.

Вход. данные – xml файл списка задач кладовщика

Вых. результат – записи в базе данных в таблицах.

№	Условие	Ожидаемый результат	Тип теста
21	xml файл списка задач кладовщика с данными в пределах описанных ограничений	записи в базе данных в соответствии с количеством отправленных задач и количеством товаров	Общий
22	xml файл списка задач кладовщика с данными за пределами описанных ограничений	отсутствие записи в базе данных	Негативный
23	xml файл списка задач кладовщика с данными на границах описанных ограничений	записи в базе данных в соответствии с количеством отправленных задач и количеством товаров	Краевой

Класс: StatusTasksTask (Вход. параметры id документа)

getBody – формирование xml файла для синхронизации статуса исполняется с Web сервисом

Вход. данные - id документа для синхронизации.

Вых. результат - xml файла для синхронизации с web сервисом при принятии задачи к исполнению.

Данные для тестирования – наличие в базе данных документов комплектации и консолидации для тестирования.

№	Условие	Ожидаемый результат	Тип теста
24	id документа для синхронизации, который есть в таблице	xml файла для синхронизации с web сервисом при принятии задачи к исполнению.	Общий
25	id документа для синхронизации (последний по номеру	xml файла для синхронизации с web сервисом при принятии	Краевой

	документ в таблице)	задачи к исполнению.	
26	id документа, которого нет в таблице	пустая xml	Негативный
27	вместо id уукрпупоеуплру	пустая xml	Негативный

Класс: CompleteTask

getBody – формирование xml документа для синхронизации задачи «Комплектации» с web сервисом основного приложения.

Вход. данные - id документа для синхронизации.

Вых. результат - xml файла для синхронизации с web сервисом при исполнении задачи комплектации

№	Условие	Ожидаемый результат	Тип теста
28	id документа для синхронизации, который есть в таблице	xml файла для синхронизации с web сервисом при исполнении задачи комплектации	Общий
29	id документа для синхронизации (последний по номеру документ в таблице)	xml файла для синхронизации с web сервисом при исполнении задачи комплектации	Краевой
30	id документа, которого нет в таблице	пустая xml	Негативный
31	вместо id уукрпупоеуплру	пустая xml	Негативный

Класс: ConsolidationTask

getBody – формирование xml документа для синхронизации задачи «Консолидации» с web сервисом основного приложения.

Вход. данные - id документа для синхронизации.

Вых. результат - xml файла для синхронизации с web сервисом при исполнении задачи консолидации.

№	Условие	Ожидаемый результат	Тип теста
32	id документа для синхронизации, который есть в таблице	xml файла для синхронизации с web сервисом при исполнении задачи консолидации	Общий

33	id документа для синхронизации (последний по номеру документ в таблице)	xml файла для синхронизации с web сервисом при исполнении задачи консолидации	Краевой
34	id документа, которого нет в таблице	пустая xml	Негативный
35	вместо id уукрпупоеуплру	пустая xml	Негативный

Класс: DocFunctions

Функции:

loadDocuments – функция загрузки документов в список задач из базы данных

№	Условие	Ожидаемый результат	Тип теста
36	массив id документов, которые есть в таблице	полученные таблицы с выбранными документами	Общий
37	массив содержащий id документа (первого и последнего в таблице)	полученные таблицы с выбранными документами	Краевой
38	id документа, которого нет в таблице	Пустая таблица	Негативный
39	вместо id уукрпупоеуплру	Пустая таблица	Негативный

loadDocument – функция получения выбранного документа в форму документа из базы данных

№	Условие	Ожидаемый результат	Тип теста
40	id документа для синхронизации, который есть в таблице	полученные таблицы с выбранными документами	Общий
41	id документа для синхронизации (последний по номеру документ в таблице)	полученные таблицы с выбранными документами	Краевой
42	id документа, которого нет в таблице	Пустая таблица	Негативный
43	вместо id уукрпупоеуплру	Пустая таблица	Негативный

deleteDocument – удаление документа из базы

Вход. данные - id документа для удаления.

Вых. результат – отсутствие соответствующей записи в базе данных

№	Условие	Ожидаемый результат	Тип теста
44	id документа для удаления, который есть в таблице	отсутствие соответствующей записи в базе данных	Общий
45	id документа для удаления (последний по номеру документ в таблице)	отсутствие соответствующей записи в базе данных	Краевой
46	id документа, которого нет в таблице	ничего не изменилось	Негативный
47	вместо id уукрпупоеуплру	ничего не изменилось	Негативный

deleteDocumentRows – удаление строки таблицы документа из базы данных

Вход. данные - id документа, id строки для удаления

Вых. результат – отсутствие соответствующей записи в базе данных

№	Условие	Ожидаемый результат	Тип теста
48	id документа, который есть в базе, id строки	отсутствие соответствующей записи в базе данных	Общий
49	id документа (последний по номеру документ в таблице), id строки последняя в таблице	отсутствие соответствующей записи в базе данных	Краевой
50	id документ, которого нет в базе, id строки, которой нет в базе	ничего не изменилось	Негативный
51	вместо id документа уукрпупоеу, вместо id строки «оаорро»	ничего не изменилось	Негативный

changeStatusTask – смена статуса документа в базе данных

Вход. данные - id документа изменения статуса

статус задачи из списка (ожидает исполнения, проблема, исполняется)

Вых. результат – изменение статуса у документа в базе

№	Условие	Ожидаемый результат	Тип теста
52	id документа у	изменение	Общий

	которого изменяется статус, статус из списка статусов	соответствующего статуса у документа	
53	id документа (последний по номеру документ в таблице), статус из списка статусов	изменение соответствующего статуса у документа	Краевой
54	id документ, которого нет в базе, статус которого нет в списке	ничего не изменилось	Негативный
55	вместо id документа уукрупноеу, вместо статус любая строка	ничего не изменилось	Негативный

Интеграционное тестирование:

1. TaskListTask и DocFunction

функции saveTaskList и LoadDocuments – выполняются последовательно.

№	Условие	Ожидаемый результат	Тип теста
56	xml файл списка задач кладовщика с данными в пределах описанных ограничений	полученные таблицы с загруженными документами в базе данных	Общий
57	xml файл списка задач кладовщика с данными за пределами описанных ограничений	Пустая таблица	Негативный
58	xml файл списка задач кладовщика с данными на границах описанных ограничений	полученные таблицы с загруженными документами в базе данных	Краевой
59	вместо id документа уукрупноеу, вместо статус любая строка	Пустая таблица	Негативный

2. StatusTaskTasks и DocFunction

функции getBody и changeStatusTask выполняются последовательно.

№	Условие	Ожидаемый результат	Тип теста
60	Принятие задачи «Консолидации» к	Сформированная и отправлен xml файл для	Общий

	исполнению	смены статуса в web сервисе, поменян статус задачи на «Исполняется»	
61	Принятие задачи «Комплектации» к исполнению	Сформированная и отправлен xml файл для смены статуса в web сервисе, поменян статус задачи на «Исполняется»	Общий

функции `getBody` и `changeStatusTask`, `deleteDocumentRow` и `deleteDocument` выполняются последовательно.

№	Условие	Ожидаемый результат	Тип теста
62	Действие на присвоение статуса «Исполнена» в задаче комплектации	Сформированная и отправлен xml файл для синхронизации в web сервисе, удалена задача из базы	Общий
63	Действие на присвоение статуса «Исполнена» в задаче консолидации	Сформированная и отправлен xml файл для смены статуса в web сервисе, удалена задача из базы	Общий

3. `ConcolidationTask` и `DocFunction`

функция `getBody` и `changeStatusTask` выполняются последовательно

№	Условие	Ожидаемый результат	Тип теста
63	Действие на присвоение статуса «Проблема» в задаче консолидации	Сформированная и отправлен xml файл для синхронизации задачи консолидации, смена статуса на «Проблема» в документе в базе	Общий

функции `getBody`, `changeStatusTask`, `deleteDocumentRow` и `deleteDocument` выполняются последовательно.

№	Условие	Ожидаемый результат	Тип теста
64	Действие на присвоение статуса «Исполнена» в задаче консолидации	Сформированная и отправлен xml файл для синхронизации задачи консолидации, удаление задачи из базы данных	Общий

4. `CompleteTask` и `DocFunction`

функция `getBody`, `changeStatusTask` выполняются последовательно.

№	Условие	Ожидаемый результат	Тип теста
65	Действие на присвоение статуса «Проблема» в задаче комплектации	Сформированная и отправлен xml файл для синхронизации задачи комплектации, смена статуса на «Проблема» в документе в базе	Общий

функции `getBody`, `changeStatusTask`, `deleteDocumentRow` и `deleteDocument` выполняются последовательно.

№	Условие	Ожидаемый результат	Тип теста
66	Действие на присвоение статуса «Исполнена» в задаче комплектации	Сформированная и отправлен xml файл для синхронизации задачи комплектации, , удаление задачи из базы данных	Общий

Стресс тестирование

Стрестесты проводятся для модулей:

Класс: `TaskListTask`

функция – `saveTaskList`

№	Условие	Ожидаемый результат	Тип теста
67	Загрузка 5 задач по 15 товару	Все загружено в БД	Стресс тест
68	Загрузка 30 задач по 15 товару	Все загружено в БД	Стресс тест
69	Загрузка 60 задач по 50 товару	Все загружено в БД	Стресс тест
70	Загрузка 5 задач по 20 товару	Все загружено в БД	Стресс тест
71	Загрузка 30 задач по 50 товару	Все загружено в БД	Стресс тест
72	Загрузка 60 задач по 150 товару	Все загружено в БД	Стресс тест

Класс: `FunctionDocument`

функция – `LoadDocuments`

№	Условие	Ожидаемый результат	Тип теста
73	Получение 5 задач по 15 товару	Все задачи получены	Стресс тест
74	Получение 30 задач по 15 товару	Все задачи получены	Стресс тест
75	Получение 60 задач по	Все задачи получены	Стресс тест

	50 товару		
76	Получение 5 задач по 20 товару	Все задачи получены	Стресс тест
77	Получение 30 задач по 50 товару	Все задачи получены	Стресс тест
78	Получение 60 задач по 150 товару	Все задачи получены	Стресс тест

Реализации тестов

Описанные ниже тесты реализованы при помощи библиотеки JUnit с использованием Android SDK (поддержка платформозависимых средств).

Пример:

```
package ru.kenarov.qrtask;
```

```
import junit.framework.Test;
import junit.framework.TestCase;
```

```
/**
```

```
 * Created by Admin on 11.12.2016.
```

```
*/
```

```
public class PublicFunctionsTest extends TestCase {
```

```
    public void setUp() throws Exception {
        super.setUp();
    }
```

```
    public void testDecryptPassword() throws Exception {
```

```
        String encrypted_password = "1232354";
```

```
        String decrypted_password = "";
```

```
        char[] symbols = encrypted_password.toCharArray();
```

```
        int character;
```

```
        boolean flag = false;
```

```
        boolean first_symbol = true;
```

```
        for (int symbol : symbols) {
```

```
            if(flag) {
```

```
                if(!first_symbol) {
```

```
                    decrypted_password += Character.toString((char) symbol);
```

```
                }
```

```
                first_symbol = false;
```

```
            }
```

```
            flag = !flag;
```

```
        }
```

```
        String expValue = "225";
```

```
        assertEquals(expValue, decrypted_password);
```

```
    }
```

```
}
```

Журнал тестирования:

Блочные тесты:

№ теста	Дата	Результат	Номер ошибки в журнале ошибок
---------	------	-----------	-------------------------------

1	13.12.16	пройден	
2	13.12.16	пройден	
3	13.12.16	пройден	
4	13.12.16	ошибка	1
5	13.12.16	пройден	
6	13.12.16	пройден	
7	13.12.16	ошибка	2
8	13.12.16	пройден	
9	18.12.16	пройден	
10	18.12.16	пройден	
11	18.12.16	ошибка	3
12	18.12.16	пройден	
13	18.12.16	пройден	
14	18.12.16	пройден	
15	18.12.16	пройден	
16	18.12.16	пройден	
17	18.12.16	пройден	
18	18.12.16	пройден	
19	18.12.16	пройден	

20	18.12.16	ошибка	4
21	18.12.16	пройден	
22	18.12.16	пройден	
23	18.12.16	ошибка	5
24	18.12.16	пройден	
25	18.12.16	пройден	
26	18.12.16	пройден	
27	18.12.16	пройден	
28	18.12.16	пройден	
29	18.12.16	пройден	
30	18.12.16	пройден	
31	18.12.16	пройден	
32	18.12.16	пройден	
33	24.12.16	пройден	
34	24.12.16	пройден	
35	24.12.16	пройден	
36	24.12.16	пройден	
37	24.12.16	пройден	
38	24.12.16	пройден	

39	24.12.16	пройден	
40	24.12.16	пройден	
41	24.12.16	пройден	
42	24.12.16	пройден	
43	24.12.16	пройден	
44	24.12.16	пройден	
45	24.12.16	пройден	
46	24.12.16	пройден	
47	24.12.16	ошибка	6
48	24.12.16	пройден	
49	24.12.16	пройден	
50	24.12.16	пройден	
51	24.12.16	ошибка	
52	24.12.16	пройден	
53	24.12.16	пройден	
54	24.12.16	пройден	
55	24.12.16	пройден	

Интеграционные тесты:

№ теста	Дата	Результат	Номер ошибки в
---------	------	-----------	----------------

			журнале ошибок
56	24.12.16	пройден	
57	24.12.16	пройден	
58	24.12.16	пройден	
59	24.12.16	пройден	
60	24.12.16	пройден	
61	24.12.16	пройден	
62	24.12.16	пройден	
63	24.12.16	пройден	
64	24.12.16	пройден	
65	24.12.16	пройден	
66	24.12.16	пройден	

Нагрузочные тесты:

№ теста	Дата	Результат	Номер ошибки в журнале ошибок
67	24.12.16	пройден	
68	24.12.16	пройден	
69	24.12.16	пройден	
70	24.12.16	пройден	

71	24.12.16	пройден	
72	24.12.16	не пройден	7
73	24.12.16	пройден	
74	24.12.16	пройден	
75	24.12.16	пройден	
76	24.12.16	пройден	
77	24.12.16	пройден	
78	24.12.16	не пройден	8

Журнал найденных ошибок.

Номер отчета, дата составления отчета, номер теста, ожидаемый результат, фактический результат.

Ошибка 1

Дата составления 13.12.16

Номер теста: 4.

Описание

Отсутствие проверки входных данных.

Входные данные: РОРПРРвсдвос.

Ожидаемый результат: Пустая строка.

Фактический результат: ОПРсв.

Ошибка 2

Дата составления 13.12.16

Номер теста: 7.

Описание

Отсутствие проверки входных данных на символы, которые не должны попасть в базу.

Входные данные: Hrfuhreuhergurehguhruh fjrhfreherhgerghregjregergergb frejherhg284543b542nmb ;;;3435' "/242359@\$%#^&\$*(&\$#@.

Ожидаемый результат: Пустая строка.

Фактический результат:

Ошибка записи в базу данных

Method threw 'java.lang.NullPointerException' exception. Cannot evaluate android.content.ContentValues.toString().

Ошибка 3

Дата составления 13.12.16

Номер теста: 11.

Описание

Необходима проверка на домен.

Входные данные: <http://qr.234.ru/?A=138282&P=4&N=325253>.

Ожидаемый результат: пустая строка.

Фактический результат:

Структура вида (A – 138282, P – 4, N - 145656).

Ошибка 4.

Дата составления 13.12.16

Номер теста: 20.

Описание

Входные данные: оао

Ожидаемый результат: отсутствие записи в базе данных.

Фактический результат:

ошибка

at com.android.internal.

os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit.java:824)

at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:640).

Ошибка 5

Дата составления 13.12.16

Номер теста: 23.

Описание

Входные данные:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
```

```
  <soap:Header/>
```

```
  <soap:Body>
```

```
    <m:TaskListResponse xmlns:m="http://www.w3.org/2001/XMLSchema">
```

```
      <m:return xsi:type="m:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</m:return>
```

```
      <m:List xsi:type="m:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><![CDATA[<?xml version="1.0" encoding="windows-1251"?>
```

```
<ListTask Version="14">
```

```
  <Task StatusTask="Ожидает исполнения" View="Консолидация" NumberTask="17640">
```

```
    <ListProduct>
```

```
      <Number>1</Number>
```

```
      <Article>128130</Article>
```

```
      <NameProduct>Микрофон Audio Technica
```

```
ATM75cW/Микрофон</NameProduct>
```

```
      <Unit>шт</Unit>
```

```
      <SumProduct>1</SumProduct>
```

```
      <Zakaz>788042</Zakaz>
```

```
      <Fact>Нет</Fact>
```

```
<IdSticker>14586436436361</IdSticker>
</ListProduct>
<ListProduct>
  <Number>1</Number>
  <Article>120767</Article>
  <NameProduct>USB
кабелъвраокпркеопиоерокыпопууреоиокуоаокиуеикоимъвкиипрквиипорпоевопткеекрнк кморуъ
Prolink PB458-0150 (USB - micro USB 3.0 (AM-BM), 1,5м.)</NameProduct>
  <Unit>шт</Unit>
  <SumProduct>1</SumProduct>
  <Zakaz>789776</Zakaz>
  <Fact>Her</Fact>
  <IdSticker>146105</IdSticker>
</ListProduct>
</Task>
</ListTask>]]</m:List>
</m:TaskListResponse>
</soap:Body>
</soap:Envelope>
```

Ожидаемый результат:

Отсутствие записи в базе данных.

В строках

```
<Article>1435254520767</Article>
```

```
<NameProduct>USB
```

```
кабелъвраокпркеопиоерокыпопууреоиокуоаокиуеикоимъвкиипрквиипорпоевопткеекрнк кморуъ
Prolink PB458-0150 (USB - micro USB 3.0 (AM-BM), 1,5м.)</NameProduct>
```

Артикул и наименования за границами возможных данных.

Фактический результат:

Ошибка записи в базу данных.

Ошибка 6.

Дата составления 20.12.16

Номер теста: 47.

Описание

Входные данные:

id документа уукрпупоеу

Ожидаемый результат:

ничего не произошло.

Фактический результат:

ошибка попытки удаления документа с несуществующим идентификатором привело к падению приложения, отсутствие проверки входных данных.

Ошибка 7.

Дата составления 20.12.16

Номер теста: 51.

Описание

id документа уукрупное, id строки «оаорро»

Входные данные:

Ожидаемый результат:

ничего не произошло.

Фактический результат:

ошибка попытки удаления строки с несуществующим идентификатором привело к падению приложения, отсутствие проверки входных данных.

Ошибка 8.

Дата составления 24.12.16

Номер теста: 72

Описание

Входные данные:

Загрузка 60 задач по 150 товару.

Ожидаемый результат:

Загрузка всех задач.

Фактический результат:

Зависание процедуры разбора и вылет.

Ошибка 9.

Дата составления 24.12.16

Номер теста: 78

Описание

Входные данные:

Получение 60 задач по 150 товару

Ожидаемый результат:

Получение всех задач.

Фактический результат:

Не удалась загрузка задач, приложение зависло.

Методы покрытия

Расчёт покрытия тестами относительно исполняемого кода производится по формуле:

Covering = Testing_lenght/Code_lenght.

Testing_lenght - количество строк кода, покрытых тестами = 1356

Code_lenght - общее количество строк кода в программе = 2425

Коэффициент покрытия: 0,559

Процент покрытия: 55,9%