

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

*Факультет*  
вычислительной техники

*Кафедра*  
МОиПЭВМ

|                        |                                                                    |
|------------------------|--------------------------------------------------------------------|
| Направление подготовки | 09.04.02 «Информационные системы и технологии»                     |
| Магистерская программа | Проектирование, разработка и эксплуатация<br>информационных систем |

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему

**Модель и методика оценки качества  
мобильного программного обеспечения**

|                |                 |                            |
|----------------|-----------------|----------------------------|
| Студент        | _____           | Кузин Денис Олегович       |
|                | (подпись, дата) | (ФИО полностью)            |
| Руководитель   | _____           | Макарычев Петр Петрович    |
|                | (подпись, дата) | (ФИО полностью)            |
| Нормоконтролер | _____           | Такташкин Денис Витальевич |
|                | (подпись, дата) | (ФИО полностью)            |
| Рецензент      | _____           | Афонин Александр Юрьевич   |
|                | (подпись, дата) | (ФИО полностью)            |

Работа допущена к защите (протокол заседания кафедры от \_\_\_\_\_ № \_\_\_\_\_)

|                     |           |                     |
|---------------------|-----------|---------------------|
| Заведующий кафедрой | _____     | Макарычев П.П.      |
|                     | (подпись) | (фамилия, инициалы) |

Работа защищена с отметкой \_\_\_\_\_ (протокол заседания ГЭК от \_\_\_\_\_ № \_\_\_\_\_)

|               |           |                     |
|---------------|-----------|---------------------|
| Секретарь ГЭК | _____     | Попова Н.А.         |
|               | (подпись) | (фамилия, инициалы) |

Пенза, 2020

## Содержание

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Введение.....                                                            | 6  |
| 1 Анализ предметной области исследования.....                            | 9  |
| 1.1 Проблемы оценки качества программного обеспечения .....              | 9  |
| 1.1.1 Качество программного обеспечения как необходимость.....           | 9  |
| 1.1.2 Стандарт требований к качеству.....                                | 10 |
| 1.2 Методы оценки показателей качества МПО .....                         | 11 |
| 1.2.1 Измерительный метод.....                                           | 11 |
| 1.2.2 Регистрационный метод.....                                         | 11 |
| 1.2.3 Органолептический метод .....                                      | 12 |
| 1.2.4 Расчетный метод.....                                               | 12 |
| 1.2.5 Экспертный метод .....                                             | 12 |
| 1.2.6 Социологический метод.....                                         | 13 |
| 1.3 Метод анализа иерархий .....                                         | 13 |
| Выводы .....                                                             | 15 |
| 2 Описание методики оценки качества мобильного ПО .....                  | 17 |
| Выводы .....                                                             | 24 |
| 3 Проектирование мобильного программного обеспечения .....               | 25 |
| 3.1 Функциональные требования .....                                      | 25 |
| 3.2 Выбор инструментальных средств разработки .....                      | 26 |
| 3.3 Пользовательский интерфейс клиентского приложения.....               | 26 |
| Выводы .....                                                             | 39 |
| 4 Тестирование и оценка качества МПО на разных этапах<br>разработки..... | 40 |
| 4.1 Оценка функциональности МПО.....                                     | 40 |

|       |                                                                                                                     |    |
|-------|---------------------------------------------------------------------------------------------------------------------|----|
| 4.1.1 | Оценка функциональности МПО на раннем этапе<br>разработки.....                                                      | 40 |
| 4.1.2 | Оценка функциональности МПО после доработок и<br>устранения недостатков, выявленных после первой оценки .....       | 42 |
| 4.1.3 | Оценка функциональности финальной версии МПО .....                                                                  | 43 |
| 4.2   | Оценка производительности МПО .....                                                                                 | 46 |
| 4.2.1 | Оценка производительности МПО на раннем этапе<br>разработки.....                                                    | 46 |
| 4.2.2 | Оценка производительности МПО после доработок и<br>устранения недостатков, выявленных после первой оценки .....     | 47 |
| 4.2.3 | Оценка производительности финальной версии МПО .....                                                                | 48 |
| 4.3   | Оценка удобства использования МПО .....                                                                             | 49 |
| 4.3.1 | Оценка удобства использования МПО на раннем этапе<br>разработки.....                                                | 50 |
| 4.3.2 | Оценка удобства использования МПО после доработок и<br>устранения недостатков, выявленных после первой оценки ..... | 50 |
| 4.3.3 | Оценка удобства использования финальной версии МПО .....                                                            | 51 |
| 4.4   | Оценка надежности МПО .....                                                                                         | 52 |
| 4.4.1 | Оценка надежности МПО на раннем этапе разработки .....                                                              | 52 |
| 4.4.2 | Оценка надежности МПО после доработок и устранения<br>недостатков, выявленных после первой оценки .....             | 53 |
| 4.4.3 | Оценка надежности финальной версии МПО .....                                                                        | 54 |
| 4.5   | Оценка безопасности МПО .....                                                                                       | 54 |
| 4.5.1 | Оценка безопасности МПО на раннем этапе разработки.....                                                             | 54 |
| 4.5.2 | Оценка безопасности МПО после доработок и устранения<br>недостатков, выявленных после первой оценки .....           | 56 |
| 4.5.3 | Оценка безопасности финальной версии МПО.....                                                                       | 56 |

|       |                                                                                                                     |    |
|-------|---------------------------------------------------------------------------------------------------------------------|----|
| 4.6   | Оценка удобства сопровождения МПО .....                                                                             | 56 |
| 4.6.1 | Оценка удобства сопровождения МПО на раннем этапе<br>разработки.....                                                | 57 |
| 4.6.2 | Оценка удобства сопровождения МПО после доработок и<br>устранения недостатков, выявленных после первой оценки ..... | 58 |
| 4.6.3 | Оценка удобства сопровождения финальной версии МПО.....                                                             | 59 |
| 4.7   | Оценка переносимости МПО .....                                                                                      | 60 |
| 4.7.1 | Оценка переносимости МПО на раннем этапе разработки .....                                                           | 60 |
| 4.7.2 | Оценка переносимости МПО после доработок и устранения<br>недостатков, выявленных после первой оценки .....          | 61 |
| 4.7.3 | Оценка переносимости финальной версии МПО.....                                                                      | 61 |
| 4.8   | Оценка информационных характеристик МПО .....                                                                       | 62 |
| 4.9   | Ранжирование показателей качества и получение их весов с<br>помощью метода анализа иерархий .....                   | 63 |
| 4.10  | Итоговая оценка приложения.....                                                                                     | 66 |
|       | Выводы .....                                                                                                        | 67 |
|       | Заключение .....                                                                                                    | 68 |
|       | Список использованных источников .....                                                                              | 69 |
|       | Приложение А. Глоссарий .....                                                                                       | 74 |
|       | Приложение Б. Код программы.....                                                                                    | 76 |

### **Список сокращений**

ПО – программное обеспечение;

МПО – мобильное программное обеспечение;

ИТ – информационные технологии;

ЛС – лицевой счет;

ПУ – прибор учета;

УК – управляющая компания.

## Введение

**Актуальность темы исследования.** Современный человек делает все для того, чтобы достигнуть максимального комфорта и удобства в своей жизни. Для этого сегодня специалистами в области информационных технологий разрабатываются мобильные приложения, которые позволяют решать огромное количество задач в разных областях жизнедеятельности человека.

Разработка мобильных приложений отличается от разработки обычного ПО наличием ряда уникальных требований. Необходимо, чтобы они корректно функционировали на разных платформах, которые отличаются используемыми операционными системами, размерами экрана, вычислительными ресурсами и продолжительностью непрерывной работы от батареи. Мобильные приложения должны поддерживать множество каналов ввода (клавиатура, голос, жесты и т.д.), мультимедийные технологии и обладать другими особенностями, повышающими удобство их использования. Для удержания низких цен на оборудование необходимо широко использовать средства моделирования и виртуализации. И, поскольку большинство мобильных сервисов поддерживают достаточно широкий спектр беспроводных сетей (2G, 3G, 4G, Wi-Fi, WiMax), мобильные приложения должны корректно функционировать в неоднородной сетевой среде.

Мировое сообщество серьезно озабочено возможностью появления различного рода дефектов в программном коде. Подтверждением тому может быть факт создания международных стандартов на разработку программных систем (ISO/IEC, CMM/CMMI, др.). Более того, за последние несколько десятилетий было создано большое количество различных методологий и подходов к разработке различных типов программных систем. Каждая такая методология стремится к получению работоспособного продукта после завершения его разработки. Некоторые из методологий делают акцент на последовательность процессов разработки, некоторые – делают процесс

максимально простым и гибким. Тем не менее, каждая из таких методологий стремится к созданию качественного продукта.

Однако как известно, мы не можем контролировать то, что не можем измерить. Это же утверждение можно отнести и к качеству приложений. Трудно принимать какие-либо проектные решения при отсутствии количественных измерений характеристик. Поэтому проблема измерения качества приложений на сегодняшний день является актуальной.

Исходя из приведенных ранее доводов, встает задача контроля качества приложений и удержания его на необходимом уровне. Однако для того, чтобы контролировать качественные показатели, необходимо иметь численные характеристики этих показателей. Другими словами, для контроля качества приложений необходимо уметь измерять это качество.

**Целью диссертационной работы** является разработка методики оценки качества МПО с учетом требований и особенностей эксплуатации.

**Задачи диссертационной работы:**

- проведение критического анализа предметной области по тематике магистерской диссертации;
- анализ методов получения значений критериев качества;
- разработка МПО на платформе «1С: Предприятие»;
- практическое применение сформулированной методики.

**Объектом исследования** диссертационной работы являются методики оценки качества МПО.

**Предметом исследования** являются критерии качества МПО и методы получения их значений.

**Научная новизна диссертационной работы:**

- разработана методика оценки качества МПО, которая отличается от известных тем, что позволяет учитывать особенности исследуемого приложения;
- разработано МПО для практического применения сформулированного метода и оценки его эффективности.

### **Практическая значимость**

Разработанная методика была применена для оценки качества мобильного программного обеспечения, разработанного на платформе «1С: Предприятие», которое в дальнейшем планируется использовать в сервисе «DOMUCHET.ONLINE».

### **Результаты, выносимые на защиту:**

- проведён анализ существующих методик оценки качества МПО;
- проведен анализ существующих методов получения значений критериев качества МПО;
- предложена методика оценки качества МПО;
- разработано МПО на платформе «1С: Предприятие»;
- произведена оценка эффективности предложенной методики.

**Публикации.** По материалам диссертации опубликована 1 статья.

**Структура и объем диссертации.** Диссертация состоит из введения, четырех глав, заключения, списка использованных источников, 3 приложений. Основная часть изложена на 71 страницах, содержит 14 рисунков, 10 таблиц. Диссертационная работа изложена на 107 страницах.

## **1 Анализ предметной области исследования**

### **1.1 Проблемы оценки качества программного обеспечения**

Рынок мобильных приложений в настоящее время переживает небывалый рост. Пользователи всё чаще используют смартфон вместо компьютера. К январю 2020 года количество пользователей мобильных телефонов достигла 5,19 миллиарда человек [1].

На мобильные приложения теперь приходится 10 из каждых 11 минут пользования мобильным устройством, а на просмотр веб-страниц уходит только 9% нашего «мобильного времени». Мобильные приложения используются практически во всех сферах нашей жизни [2].

За 2019 год пользователи смартфонов загрузили более 200 миллиардов мобильных приложений, потратив совокупно 120 миллиардов долларов на приложения и покупки в них. В среднем на каждый подключенный смартфон приходится по 21 доллару. Суммарно в магазинах App Store и Google Play предлагалось более 5,5 млн приложений [2].

Учитывая такое огромное количество МПО, обычному потребителю может быть довольно сложно выбрать программный продукт, соответствующий его потребностям и должному уровню качества и безопасности. Помочь разобраться в таком обилии приложений и сделать правильный выбор могут независимые исследования мобильных приложений с последующим выставлением оценок и созданием рейтингов. В свою очередь, для проведения исследований необходимы стандарты и методики испытаний.

#### **1.1.1 Качество программного обеспечения как необходимость**

С тех пор как программное обеспечение стало неотъемлемой частью нашей повседневной жизни, спрос на него значительно увеличился. Сегодня высокое качество воспринимается как обязательный компонент программного обеспечения. Поэтому очень важно вовлечь группы контроля качества в процесс планирования и реализации проектов с самого начала. Тем не менее до сих пор существуют компании, которые считают, что задача обеспечения

качества ПО решается путем тестирования в конце жизненного цикла разработки.

Нужно иметь в виду, что рынок предлагает множество альтернатив, в том числе бесплатное ПО с открытым исходным кодом. Кроме того, постоянно растет осведомленность клиентов и конечных пользователей о качестве программного обеспечения, которое они покупают. Приложения или корпоративные системы, которые демонстрируют низкую производительность или некачественное взаимодействие с пользователями, будут вытеснены другими продуктами.

### **1.1.2 Стандарт требований к качеству**

Раньше не существовало российских и международных стандартов, которые давали бы исчерпывающий ответ на вопрос о том, какое мобильное приложение можно назвать качественным, и что необходимо учитывать, чтобы разработать качественный программный продукт для смартфонов. В связи с этим было решено разработать стандарт требований к качеству мобильных приложений. При разработке стандарта было изучено около 60 имеющихся ИТ-стандартов, методик и директив от ведущих компаний в области мобильного программного обеспечения, из них было использовано 10 ГОСТов, 6 международных стандартов, директивы и методики корпораций Apple, Google, AQuA и OWASP. В результате был разработан стандарт организации Роскачество [3].

Стандарт обсуждался рабочей группой проектно-технического комитета ПТК 702 «Российская система качества» и несколько раз претерпевал изменения и доработки. Стандарт требований к качеству мобильных приложений содержит критерии восьми категорий: функциональность, производительность, удобство использования, надежность, безопасность, сопровождаемость, переносимость и информационные характеристики (описание мобильного приложения в магазинах приложений).

## **1.2 Методы оценки показателей качества МПО**

В соответствии с ГОСТ 28195–89 «Оценка качества программных средств» методы определения показателей качества ПО различаются:

- по способам получения информации о ПО:
  - а) измерительный;
  - б) регистрационный;
  - с) органолептический;
  - д) расчетный;
- по источникам получения информации:
  - а) экспертный;
  - б) социологический.

### **1.2.1 Измерительный метод**

Основан на получении информации о характеристиках и свойствах программного обеспечения с помощью инструментальных средств. С использованием этого метода, например, определяется число строк исходного кода программы, число комментариев, время реакции, время выполнения ветви программы и другие показатели.

Достоинством этого метода можно считать то, что результаты выражаются в общепринятых единицах измерения, а результаты сопоставимы и воспроизводимы [4].

Недостатками являются высокие затраты, т.к. для проведения испытаний требуются оборудованные испытательные лаборатории, измерительное оборудование и высококвалифицированный персонал [4].

### **1.2.2 Регистрационный метод**

Основан на получении информации во время проведения испытаний и функционирования программного обеспечения, в том время как определённые события регистрируются и подсчитываются, например: число отказов, сбоев и время их появления, время начала и окончания работы программы.

Недостатками являются трудоёмкость этого метода и длительность проведения испытаний и наблюдений [5].

### **1.2.3 Органолептический метод**

Основан на получении информации, получаемой в результате анализа восприятия органов чувств (зрения, слуха). С помощью этого метода можно определить такой показатель, как удобство использования.

Достоинствами этого метода являются быстрота получения значений показателей, доступность и отсутствие необходимости в дорогостоящем оборудовании [6].

Недостатком является субъективизм оценки, полученной этим методом, выражение её результатов в безразмерных величинах, несопоставимость и недостаточная воспроизводимость результатов [6].

### **1.2.4 Расчетный метод**

Основан на получении информации с помощью статистических данных, которые накапливаются при проведении испытаний, эксплуатации и сопровождении программного обеспечения. С помощью этого метода определяются такие данные, как длительность вычислений, их точность и затрачиваемые ресурсы.

Достоинством этого метода является то, что его можно использовать при проектировании программного продукта, когда он ещё не может быть объектом экспериментальных исследований и испытаний [7].

### **1.2.5 Экспертный метод**

Используется, когда оценка параметров процессов или предметов невозможно непосредственно измерить, либо применить любые точные науки, поэтому оценка проводится на основании профессионального опыта специалиста, как одного, тогда экспертная оценка будет индивидуальной, так и нескольких, тогда экспертная оценка будет коллективной.

Эксперты должны быть компетентными в данных вопросах, а также должны обладать определенным опытом в областях, соответствующих решаемым задачам. При подборе экспертов желательно учитывать момент личной заинтересованности, который может стать очень существенным препятствием для получения объективного суждения. С этой целью, например, применяют методы Шара [8], когда один эксперт, наиболее уважаемый специалист, рекомендует ряд других и далее по цепочке, пока не будет подобран весь необходимый коллектив. При решении задачи формирования экспертной группы необходимо выявить и стабилизировать работоспособную сеть экспертов. Эксперт должен обладать соответствующим профессиональным опытом и развитой на его основе интуицией.

Достоинством этого метода является то, что его можно применять тогда, когда показатели качества не могут быть получены с использованием точных расчетов.

Недостатками этого метода являются субъективный фактор опрашиваемых экспертов, возможность неправильного понимания рассматриваемого вопроса и большие затраты.

### **1.2.6 Социологический метод**

Основан на получении данных от фактических или потенциальных потребителей программного продукта. Сбор информации может осуществляться с помощью устного опроса, распространения анкет, организации конференций и выставок.

## **1.3 Метод анализа иерархий**

Метод относится к классу критериальных методов. Получил широкое распространение и до сих пор активно используется в управленческой практике. Использование метода приводит лицо принимающее решение не к «правильному» решению, а к варианту, наилучшим образом согласующемуся с его пониманием сути проблемы и требованиями к ее решению [9].

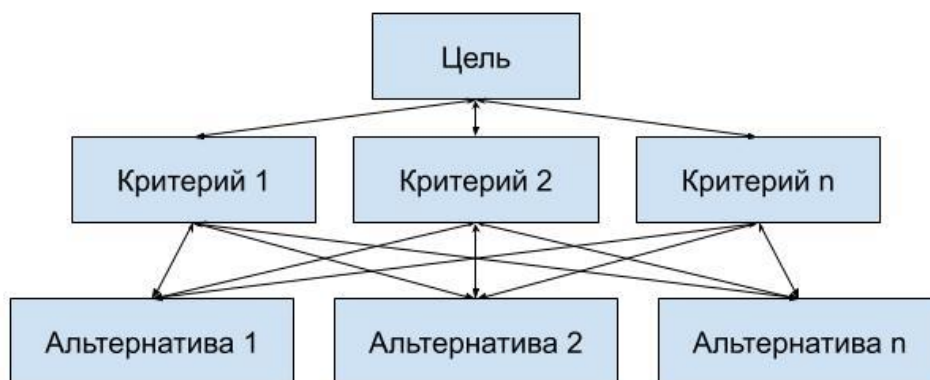


Рисунок 1 – Дерево критериев и альтернатив

Этапы метода.

1. Предварительное ранжирование критериев, в результате которого они располагаются в порядке убывания важности (значимости).

2. Парное сравнение критериев по важности по девятибалльной шкале с составлением соответствующей матрицы (таблицы) размера  $(n \times n)$ , где  $n$  – количество критериев). Система парных сведений приводит к результату, который может быть представлен в виде обратно симметричной матрицы. Элементом матрицы  $a(i,j)$  является интенсивность проявления элемента иерархии  $i$  относительно элемента иерархии  $j$ , оцениваемая по шкале интенсивности от 1 до 9, где оценки имеют следующий смысл:

- равная важность – 1;
- умеренное превосходство – 3;
- значительное превосходство – 5;
- сильное превосходство – 7;
- очень сильное превосходство – 9.

В промежуточных случаях ставятся четные оценки: 2, 4, 6, 8 (например, 4 – между умеренным и значительным превосходством).

Затем формируется матрица. В процессе заполнения матрицы если элемент  $i$  важнее элемента  $j$ , то клетка  $(i, j)$ , соответствующая строке  $i$  и столбцу  $j$ , заполняется целым числом, а клетка  $(j, i)$ , соответствующая строке  $j$  и столбцу  $i$ , заполняется обратным числом (дробью).

$a_{ij}$  = отношение критерия  $i$  к критерию  $j$ .

$$a_{ji} = 1/a_{ij}.$$

$$a_{ii} = 1.$$

3. Проводится попарное сравнение вариантов по каждому критерию аналогично тому, как это делалось для критериев, и заполняются соответствующие таблицы.

4. Находим сумму элементов каждого столбца.

$$S_j = a_{1j} + a_{2j} + \dots + a_{nj}.$$

Делим все элементы матрицы на сумму элементов соответствующего столбца.

$$A_{ij} = \frac{a_{ij}}{S_j}.$$

Эти действия называются нормировкой матрицы.

5. Находим среднее значение для каждой строки. Полученный столбец задает «веса» критериев с точки зрения поставленной цели. Этот столбец называют весовым столбцом критериев по цели.

6. Повторяем действия 4 – 5 для матриц попарного сравнения по критериям. В итоге получаем столбцы (векторы) весовых коэффициентов объектов сравнения с точки зрения соответствия отдельным критериям.

7. В результате были сформированы:

- вектор весов критериев;
- матрица весов альтернатив по каждому критерию (состоящая из полученных весовых столбцов).

Умножая полученную матрицу на столбец по правилу строка на столбец (матрично), получаем веса альтернатив с точки зрения достижения цели.

## Выводы

В данной главе выполнен анализ текущей ситуации, связанной с развитием сферы мобильных устройств и мобильных приложений. По итогам анализа выявлено, что количество пользователей, использующих мобильные

устройства, активно растёт, как и количество программных продуктов для них, публикуемых в App Store и Google Play.

Также было выявлено, что в настоящее время к проблематике указанного направления можно отнести недостаточный контроль за качеством разрабатываемого мобильного программного обеспечения.

С учетом поставленных целей и задач исследования проведен выбор методов для оценки показателей качества мобильного программного обеспечения и дальнейшего распределения их весов.

## 2 Описание методики оценки качества мобильного ПО

Методика использует для оценки качества ПО 8 показателей.

1. **Функциональность** – степень, в которой программный продукт обеспечивает выполнение функций в соответствии с заявленными и подразумеваемыми потребностями при указанных условиях. Функциональные возможности программного продукта оцениваются по степени их соответствия разработанным требованиям по десятибалльной шкале методом экспертной оценки.

При оценке применяется следующая шкала баллов [10]:

0 – функция отсутствует в имеющейся конфигурации;

2 – функция реализована частично, для ее реализации необходима серьезная доработка программного кода при настройке/внедрении;

4 – функция реализована частично, для ее реализации необходима незначительная доработка программного кода при настройке/внедрении;

6 – функция реализована удовлетворительно, требуется адаптация под нужды пользователей в процессе настройки/внедрения;

8 – функция реализована хорошо, однако в перспективе могут понадобиться ее доработки;

10 – функция реализована полностью, удовлетворяет требованиям (в том числе – на перспективу).

Оценки выстроены по шкале четности, при заполнении теста могут применяться нечетные оценки в случае, если ответ находится на грани двух смежных четных оценок.

Итоговая оценка за функциональность рассчитывается по формуле:

$$I = \frac{X_1 + X_2 + \dots + X_n}{n},$$

где I – итоговая оценка функциональности приложения;

n – количество оцениваемых функций приложения;

$X_n$  – оценка функции n.

2. Производительность – какое количество информации обрабатывается приложением в единицу времени. Для анализа быстродействия приложения будет использоваться методика Apdex (Application Performance Index) [11]. Apdex является числовой мерой удовлетворенности пользователей производительностью приложений. Для расчета Apdex собирается множество статистических данных о времени исполнения операций приложением. Далее производится компоновка данных, с учетом факторов влияния, и математическое преобразование, которое приводит все эти данные к значению в интервале от «0» до «10» («0» – все пользователи разочарованы, никто не может дождаться ответа от приложения, «10» – все пользователи удовлетворены, приложение работает быстро).

Основным фактором влияния на значение Apdex является целевое время отклика. Целевое время отклика задается до начала замера фактических времен исполнения, определяется специалистом, проводящим тестирование, и может быть скорректировано после первичных замеров.

Для расчета Apdex используются статистические данные с программных счетчиков, содержащие наименование операций, когда каждая из них инициирована, и длительность исполнения каждой операции приложением. Далее эти данные за период (например, сутки) компоуются по наименованию/операции, затем для каждой операции производится компоновка всех значений длительности исполнения этой операции, относительно целевого времени, на три зоны «отзывчивости» по отношению к пользователю.

Довольны – пользователи, при выполнении операции приложением, каких-либо задержек с ответом со стороны системы не ощутили. (Время исполнения операции приложением меньше или равно целевому времени  $T$ ).

Удовлетворены – пользователи заметили, что ответ от системы пришлось ждать, но пользователи дождались ответа от приложения. (Время исполнения операции приложением больше целевого времени  $T$ , но меньше или равно времени  $F$ . Чаще всего время  $F$  принимают равным  $4 \cdot T$ , но в

некоторых случаях коэффициент может быть изменен в большую или меньшую сторону).

Разочарованы – пользователи не дожидались ответа от приложения и прервали процесс исполнения операции приложением. (Время исполнения операции приложением больше времени F).

После этого производится расчет  $Ardex$  по формуле:

$$I = \frac{NS + \frac{NT}{2}}{N} * 10,$$

где  $I$  – итоговая оценка производительности приложения;

$N$  – количество операций;

$NS$  – количество операций с временем исполнения от 0 до  $T$ ;

$NT$  – количество операций с временем исполнения от  $T$  до  $F$ .

3. Удобство использования – показывает, насколько широкий круг пользователей может использовать приложение для достижения установленных целей с необходимой результативностью, удовлетворённостью и эффективностью. Удобство использования программного продукта оценивается по следующим параметрам [12].

Понятность – показатель, обратный к усилиям, затрачиваемым пользователями, чтобы воспринять набор понятий, на которых основано ПО, и их применимость для решения своих задач.

Удобство обучения – показатель, обратный к усилиям, затрачиваемым пользователями чтобы научиться работе с ПО.

Удобство работы – показатель, обратный к усилиям, предпринимаемым пользователями, чтобы решать свои задачи с помощью ПО.

Каждый из этих параметров оценивается по шкале от «0» до «3» методом экспертной оценки или социологическим методом, где

0 – ПО совсем не соответствует параметру;

1 – ПО частично соответствует параметру, требуется серьёзная доработка;

2 – ПО соответствует параметру, но требуется незначительная доработка;

3 – ПО полностью соответствует параметру.

Итоговая оценка за удобство использования рассчитывается по формуле:

$$I = X_1 + X_2 + \dots + X_n,$$

где  $I$  – итоговая оценка удобства использования приложения;

$n$  – количество параметров;

$X_n$  – оценка параметра  $n$ .

4. Надежность – степень выполнения мобильным приложением определенных функций в течение установленного периода времени. Для расчета надежности используется формула [13]:

$$I = \frac{n - k}{n} * 10,$$

где  $I$  – итоговая оценка надёжности;

$n$  – количество запусков приложения;

$k$  – количество запусков приложения с неудовлетворительным результатом.

5. Безопасность – степень защищенности данных пользователя. Безопасность программного продукта оценивается по следующим параметрам [12].

Небезопасное хранение данных – хранение критически важных данных в незащищенных или слабо защищенных локальных хранилищах.

Небезопасная аутентификация – отсутствие требований проверки идентификации пользователя, отсутствие проверки контроля сеанса.

Небезопасная передача данных – недостаточное подтверждение достоверности источников связи, неверные версии SSL и т.д.

Модификация данных – изменение исполняемых файлов, локальных ресурсов, перехват вызовов сторонних процессов и т.д.

Небезопасная авторизация – эта уязвимость описывает недостатки авторизации на стороне клиента, принудительный просмотр и т.д.

Каждый из этих параметров оценивается по шкале от «0» до «1» методом экспертной оценки, где

0 – ПО подвержено рассматриваемой угрозе;

1 – ПО не подвержено рассматриваемой угрозе.

Итоговая оценка за безопасность рассчитывается по формуле:

$$I = (X_1 + X_2 + \dots + X_n) * 2,$$

где I – итоговая оценка безопасности приложения;

n – количество параметров;

$X_n$  – оценка параметра n.

6. Удобство сопровождения – степень простоты внесения изменений для устранения ошибок и расширения функционала. Удобство сопровождения программного продукта оценивается по следующим параметрам [12].

Анализируемость – удобство проведения анализа ошибок, дефектов и недостатков, а также удобство анализа на предмет необходимых изменений и их возможных эффектов.

Удобство внесения изменений – показатель, обратный к трудозатратам на проведение необходимых изменений.

Стабильность – показатель, обратный к риску возникновения неожиданных эффектов при внесении необходимых изменений.

Удобство проверки – показатель, обратный к трудозатратам на проведение тестирования и других видов проверки того, что внесенные изменения привели к нужным эффектам.

Каждый из этих параметров оценивается по шкале от «0» до «3» методом экспертной оценки, где

0 – ПО совсем не соответствует параметру;

1 – ПО частично соответствует параметру, требуется серьёзная доработка;

2 – ПО соответствует параметру, но требуется незначительная доработка;

3 – ПО полностью соответствует параметру.

Итоговая оценка за удобство сопровождения рассчитывается по формуле:

$$I = (X_1 + X_2 + \dots + X_n) / 1,2,$$

где  $I$  – итоговая оценка удобства сопровождения приложения;

$n$  – количество параметров;

$X_n$  – оценка параметра  $n$ .

7. Переносимость – степень простоты переноса мобильного программного обеспечения из одной среды в другую. Переносимость программного продукта оценивается по следующим параметрам [12].

Адаптируемость – способность ПО приспосабливаться к различным окружениям без проведения для этого действий, помимо заранее предусмотренных.

Удобство установки – способность ПО быть установленным или развернутым в определенном окружении.

Способность к сосуществованию – способность ПО сосуществовать с другими программами в общем окружении, деля с ними ресурсы.

Удобство замены другого ПО данным – способность ПО использоваться вместо другого ПО для решения тех же самых задач в заданном окружении.

Каждый из этих параметров оценивается по шкале от «0» до «3» методом экспертной оценки, где

0 – ПО совсем не соответствует параметру;

1 – ПО частично соответствует параметру, требуется серьёзная доработка;

2 – ПО соответствует параметру, но требуется незначительная доработка;

3 – ПО полностью соответствует параметру.

Итоговая оценка за удобство сопровождения рассчитывается по формуле:

$$I = (X_1 + X_2 + \dots + X_n) / 1,2,$$

где  $I$  – итоговая оценка переносимости приложения;

$n$  – количество параметров;

$X_n$  – оценка параметра  $n$ .

8. Информационные характеристики – степень понятности и полноты описания функциональных возможностей приложения. Критерий оценивается по шкале от «0» до «3» методом экспертной оценки, где

0 – описание не соответствует функциональным возможностям приложения. Из описания не понятно, какие задачи позволяет решать ПО;

1 – описание частично соответствует функциональным возможностям приложения. Из описания сложно понять, какие задачи позволяет решать ПО;

2 – описание полностью соответствует функциональным возможностям приложения. Из описания легко понять, какие задачи позволяет решать ПО.

В зависимости от категории приложения, для которого будет проводиться оценка качества, перечень критериев может отличаться. Например, если сравнивать приложения для учета личных финансов и для спорта, то у них будут разные функциональные возможности. И если для первого приложения наличие пароля обязательно, то для второго нет. Поэтому веса критериев устанавливаются для каждой исследуемой категории приложений отдельно.

После расчета значений показатели ранжируются по степени важности и им присваиваются оценки по девятибалльной шкале при помощи экспертного метода. Полученные оценки используются для получения весов критериев при помощи метода анализа иерархий. Выставление итоговой оценки приложения осуществляется путём умножения каждого значения критерия на соответствующий ему вес. Все получившиеся значения суммируются. Формула для получения итоговой оценки приложения [14]:

$$I = \sum_{n=1}^n (x_n * y_n),$$

где  $I$  – итоговая оценка приложения;

$x_n$  – значение критерия;

$y_n$  – вес соответствующего критерия;

n – количество критериев.

## **Выводы**

В данной главе был изучен документ «ПНСТ 277–2018. Предварительный национальный стандарт Российской Федерации. Российская система качества. Сравнительные испытания мобильных приложений для смартфонов» (утв. и введен в действие Приказом Росстандарта от 26.06.2018 N 16–пнст). На основе полученной информации был определен и составлен список критериев, по которым будет происходить оценка качества мобильного программного обеспечения.

Также была разработана методика оценки качества МПО. Методика использует для расчета качества список критериев, который был составлен. Далее происходит ранжирование критериев с помощью экспертного метода и в установка их весов с помощью метода анализа иерархий. Это позволило сделать метод оценки универсальным, т.к. при оценке будут учитываться особенности исследуемого приложения.

### 3 Проектирование мобильного программного обеспечения

Поставлена задача реализовать МПО с целью практического применения на нём сформулированной методики оценки качества и оценки её эффективности.

#### 3.1 Функциональные требования

Диаграмма вариантов использования МПО представлена на рисунке 2.



Рисунок 2 – Диаграмма вариантов использования МПО

К разрабатываемому МПО выдвигаются следующие требования:

- получение и просмотр информации о ЛС;
- получение и просмотр информации о здании;
- получение и просмотр информации об управляющей компании;
- получение и просмотр истории начислений и оплат;
- оплата выставленных счетов;
- получение и просмотр информации о счетчиках, подключенных к ЛС;
- получение и просмотр прошлых показаний счетчиков;
- ввод текущих показаний счетчиков;

- подача заявок в диспетчерскую службу управляющей компании.

### **3.2 Выбор инструментальных средств разработки**

Для разработки мобильного программного обеспечения была выбрана среда разработки «1С: Предприятие». Данный продукт содержит все необходимые инструменты для выполнения поставленной задачи такие, как написание программного кода, отладка программного кода, визуальное описание интерфейса. «1С: Предприятие», являясь предметно-ориентированной средой разработки, уделяет особое внимание к подбору технологических возможностей, предоставляемых разработчику. Платформа ориентирована на то, чтобы актуальные для задач автоматизации бизнеса технологии предоставить разработчику в готовом виде. Фактически платформа позволяет разработчику прикладных решений задействовать необходимые и современные технологии своевременно, максимально просто и без радикальных изменений в своем приложении [15].

В качестве языка программирования использован встроенный язык программирования 1С. Данный язык является предварительно компилируемым предметно-ориентированным языком высокого уровня.

### **3.3 Пользовательский интерфейс клиентского приложения**

Пользовательский интерфейс – совокупность средств, с помощью которых пользователь взаимодействует с программой. Интерфейс одна из важнейших частей программы, так как простота и понятность интерфейса влияет на удобство и скорость работы пользователя с системой [16, 17].

Платформа «1С: Предприятие» предоставляет, для взаимодействия с пользователем, механизмы создания форм, которые содержат стандартные элементы управления:

- поля ввода;
- таблицы;
- вкладки;
- кнопки;

- надписи;
- командные панели.

Исходя из диаграммы вариантов использования и этапа анализа требований, с помощью предоставленных возможностей был спроектирован простой и интуитивно понятный графический пользовательский интерфейс. Интерфейс главной страницы показан на рисунке 3.

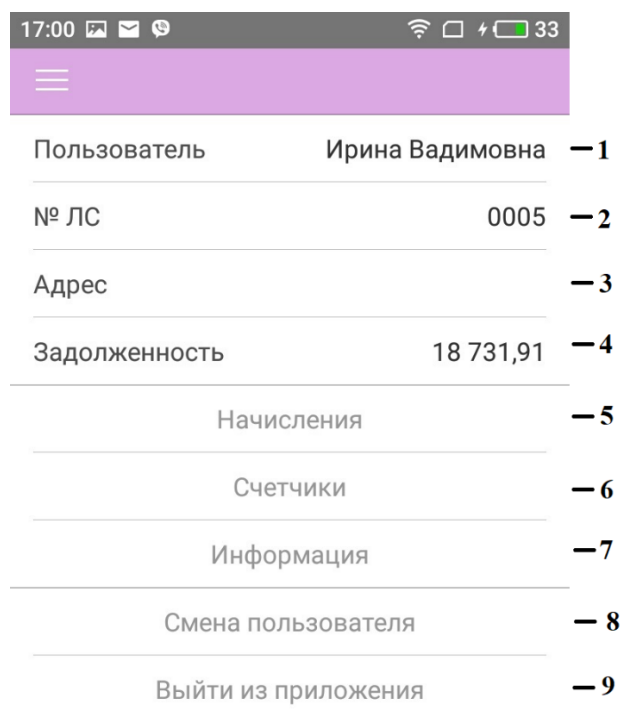


Рисунок 3 – Пользовательский интерфейс главной страницы МПО

Интерфейс главной страницы состоит из формы, на которой расположены текстовые поля и кнопки, отражающие один из вариантов работы программы:

- 1 – в текстовом поле «Пользователь» указаны Фамилия Имя Отчество текущего пользователя;
- 2 – в текстовом поле «№ ЛС» указан номер ЛС текущего пользователя;
- 3 – в текстовом поле «Адрес» указан адрес, по которому зарегистрирован текущий пользователь;
- 4 – в текстовом поле «Задолженность» указана задолженность текущего пользователя;

5 – по кнопке «Начисления» происходит переход на страницу со списком начислений;

6 – по кнопке «Счетчики» происходит переход на страницу со списком счетчиков;

7 – по кнопке «Информация» происходит переход на страницу с основной информацией о пользователе, управляющей организации, здании и помещении;

8 – по кнопке «Смена пользователя» открывается окно авторизации пользователя;

9 – по кнопке «Выйти из приложения» происходит выход из приложения.

Интерфейс страницы с начислениями показан на рисунке 4.

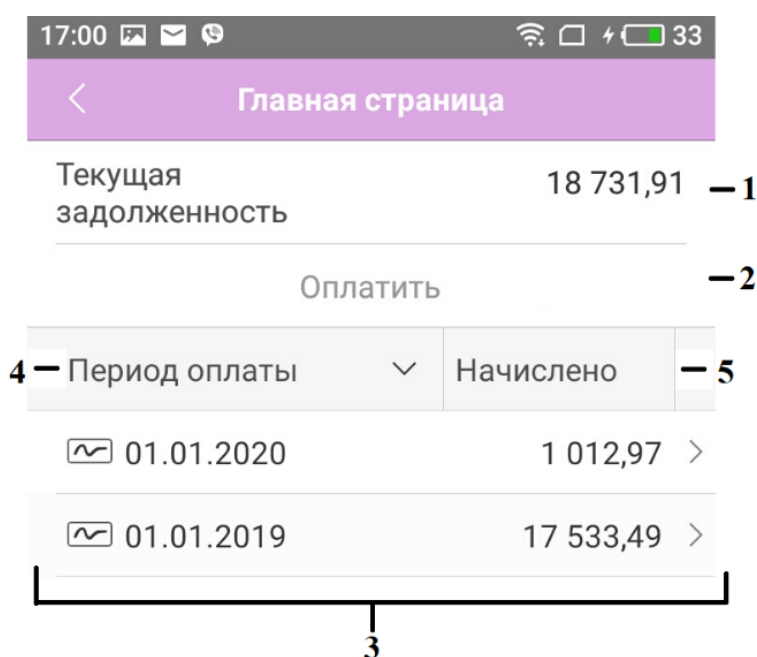
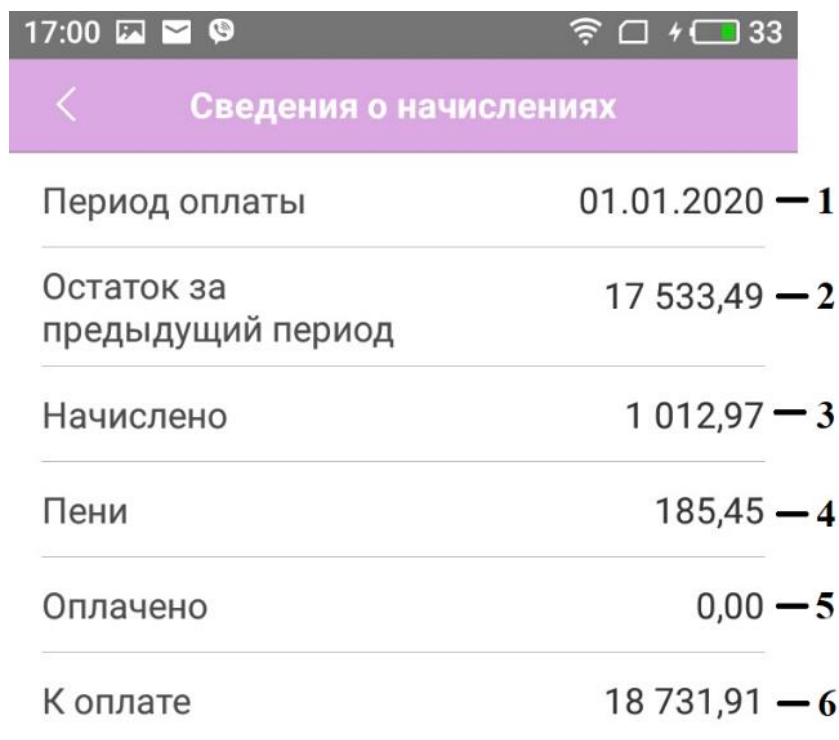


Рисунок 4 – Пользовательский интерфейс страницы начислений МПО

Интерфейс страницы начислений состоит из формы, на которой расположены текстовые поля, табличная часть и кнопки, отражающие один из вариантов работы программы:

1 – в текстовом поле «Текущая задолженность» указана общая задолженность пользователя на текущую дату;

- 2 – по кнопке «Оплатить» открывается страница внесения оплаты;
  - 3 – в табличной части расположен список всех начислений по текущему пользователю;
  - 4 – в колонке «Период оплаты» указана дата начисления;
  - 5 – в колонке «Начислено» указана начисленная сумма.
- Интерфейс страницы с расшифровкой начислений показан на рисунке 5.



|                              |                |
|------------------------------|----------------|
| Период оплаты                | 01.01.2020 — 1 |
| Остаток за предыдущий период | 17 533,49 — 2  |
| Начислено                    | 1 012,97 — 3   |
| Пени                         | 185,45 — 4     |
| Оплачено                     | 0,00 — 5       |
| К оплате                     | 18 731,91 — 6  |

Рисунок 5 – Пользовательский интерфейс страницы с расшифровкой платежей

Интерфейс страницы расшифровки начислений состоит из формы, на которой расположены текстовые поля, отражающие один из вариантов работы программы:

- 1 – в текстовом поле «Период оплаты» указана дата начисления;
- 2 – в текстовом поле «Остаток за предыдущий период» указана сумма, которая является долгом за предыдущие периоды;
- 3 – в текстовом поле «Начислено» указана сумма, которая была начислена в выбранном периоде;

4 – в текстовом поле «Пени» указана сумма пени, которая была начислена в выбранном периоде;

5 – в текстовом поле «Оплачено» указана сумма, которую оплатили в выбранном периоде;

6 – в текстовом поле «К оплате» указана сумма, которую осталось оплатить за выбранный период.

Интерфейс страницы со списком активных счетчиков показан на рисунке 6.

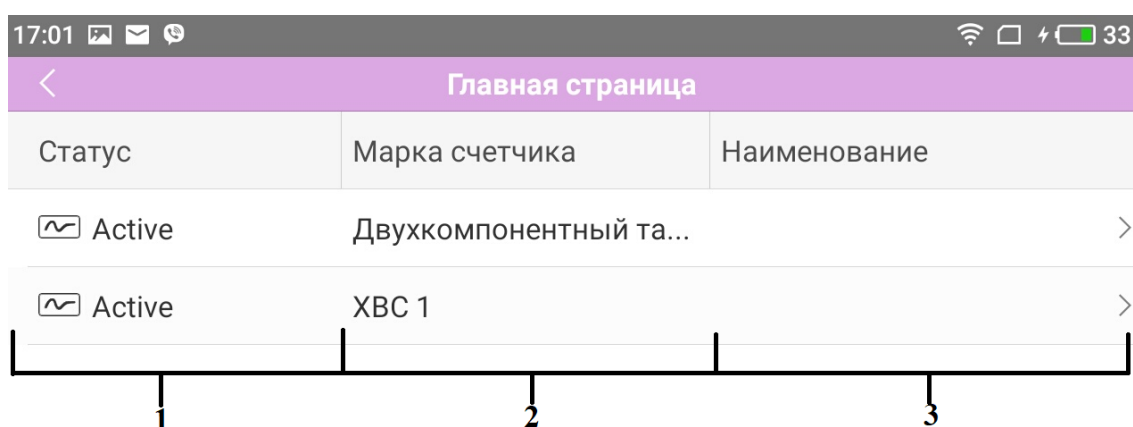


Рисунок 6 – Пользовательский интерфейс страницы со списком активных счетчиков

Интерфейс страницы со списком счетчиков состоит из формы, на которой расположена табличная часть, отражающая один из вариантов работы программы:

1 – в колонке «Статус» указан текущий статус счетчика;

2 – в колонке «Марка счетчика» указана марка счетчика;

3 – в колонке «Наименование» указано наименование счетчика.

Интерфейс страницы с подробной информацией о конкретном счетчике показан на рисунке 7.

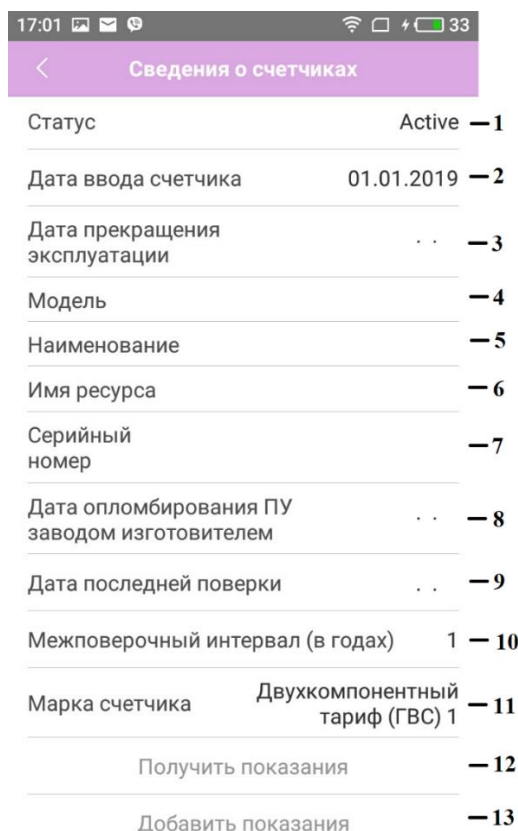


Рисунок 7 – Пользовательский интерфейс страницы с подробной информацией о конкретном счетчике

Интерфейс страницы с подробной информацией о конкретном счетчике состоит из формы, на которой расположены текстовые поля и кнопки, отражающие один из вариантов работы программы:

1 – в текстовом поле «Статус» указан текущий статус выбранного счетчика: активен или нет;

2 – в текстовом поле «Дата ввода счетчика» указана дата ввода выбранного счетчика в эксплуатацию;

3 – в текстовом поле «Дата прекращения эксплуатации» указана дата прекращения эксплуатации выбранного счетчика;

4 – в текстовом поле «Модель» указано наименование модели выбранного счетчика;

5 – в текстовом поле «Наименование» указано наименование выбранного счетчика;

6 – в текстовом поле «Имя ресурса» указано наименование ресурса, для которого установлен выбранный счетчик;

7 – в текстовом поле «Серийный номер» указан серийный номер выбранного счетчика;

8 – в текстовом поле «Дата опломбирования ПУ заводом изготовителем» указана дата опломбирования выбранного счетчика заводом изготовителем;

9 – в текстовом поле «Дата последней поверки» указана дата последней поверки выбранного счетчика;

10 – в текстовом поле «Межповерочный интервал (в годах)» указан межповерочный интервал выбранного счетчика, который измеряется в годах;

11 – в текстовом поле «Марка счетчика» указано наименование марки выбранного счетчика;

12 – по кнопке «Получить показания» открывается страница со списком показаний выбранного счетчика;

13 – по кнопке «Добавить показаний» открывается форма, в которой можно указать и записать текущие показания выбранного счетчика.

Интерфейс страницы со списком показаний конкретного счетчика показан на рисунке 8.

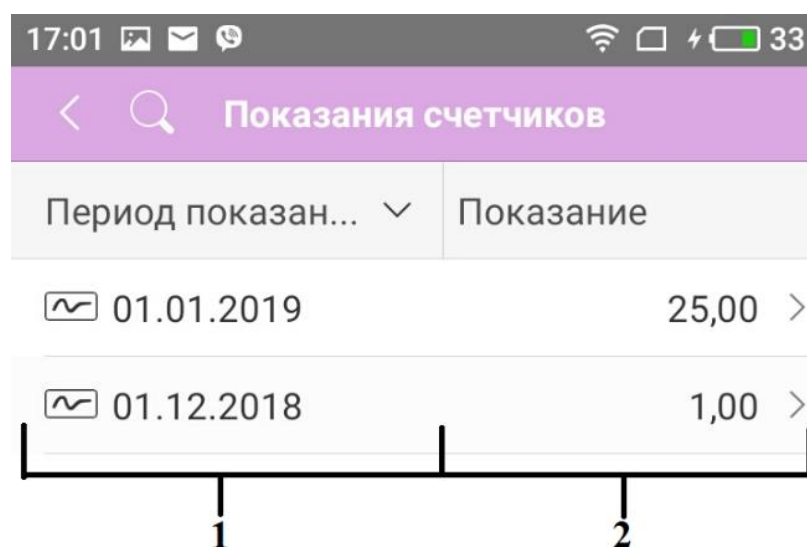


Рисунок 8 – Пользовательский интерфейс страницы со списком показаний конкретного счетчика

Интерфейс страницы со списком показаний конкретного счетчика состоит из формы, на которой расположена табличная часть, отражающая один из вариантов работы программы:

1 – в колонке «Период показаний» указана дата внесения показаний выбранного счетчика;

2 – в колонке «Показание» указано значение показания выбранного счетчика.

Интерфейс страницы с расшифровкой показания конкретного счетчика показан на рисунке 9.

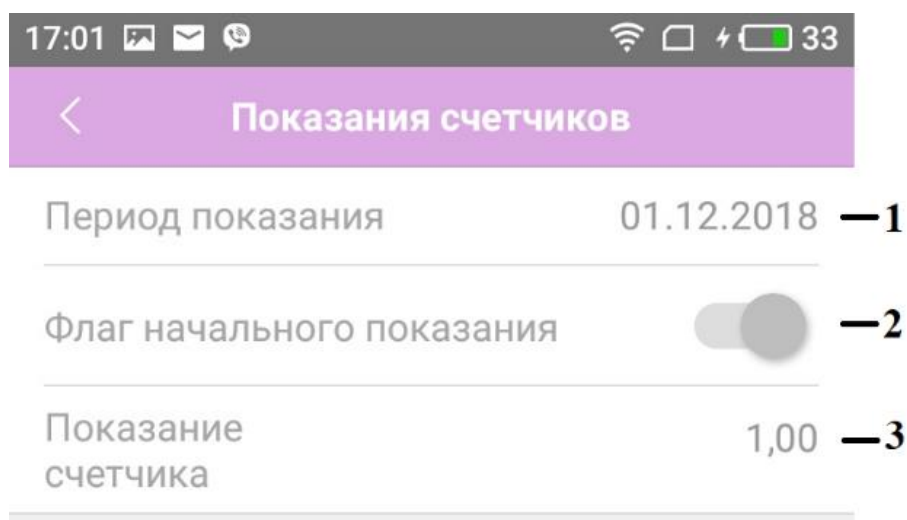


Рисунок 9 – Пользовательский интерфейс страницы с расшифровкой показания конкретного счетчика

Интерфейс страницы с расшифровкой показания конкретного счетчика состоит из формы, на которой расположены текстовые поля и переключатель значения, отражающие один из вариантов работы программы:

1 – в текстовом поле «Период показаний» указана дата ввода показания выбранного счетчика;

2 – поле переключателя значения «Флаг начального показания» указывает на то, было ли показание начальным для выбранного счетчика;

3 – в текстовом поле «Показание счетчика» указано значение показания выбранного счетчика.

Интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении показан на рисунке 10.

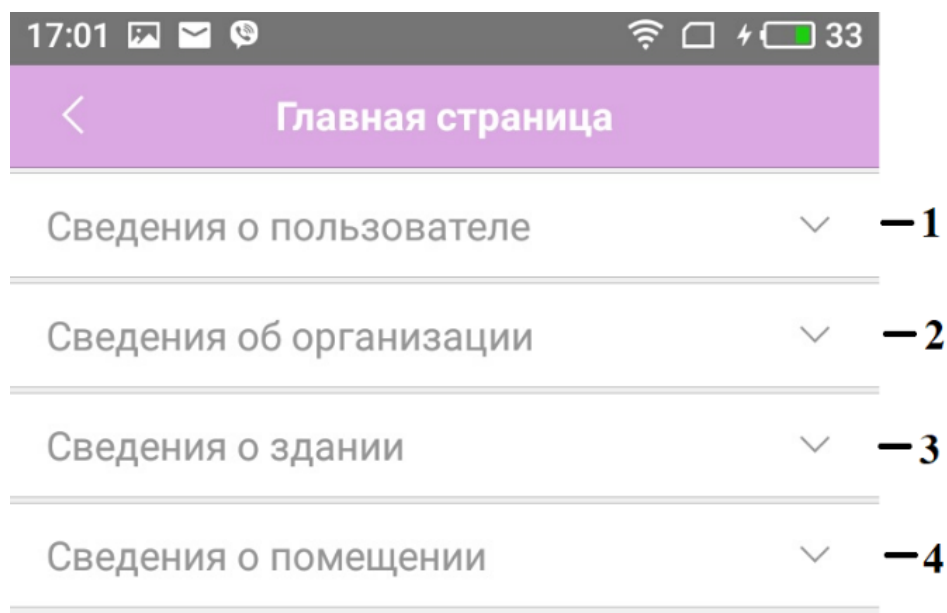


Рисунок 10 – Пользовательский интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении

Интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении состоит из формы, на которой расположены текстовые поля и раскрывающиеся вкладки, отражающие один из вариантов работы программы:

1 – на вкладке «Сведения о пользователе» находится подробная информация о текущем пользователе;

2 – на вкладке «Сведения об организации» находится подробная информация об организации, которая обслуживает дом, в котором зарегистрирован текущий пользователь;

3 – на вкладке «Сведения о здании» находится подробная информация о доме, в котором зарегистрирован текущий пользователь;

4 – на вкладке «Сведения о помещении» находится подробная информация о квартире, в которой зарегистрирован текущий пользователь.

Интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения о пользователе» показан на рисунке 11.

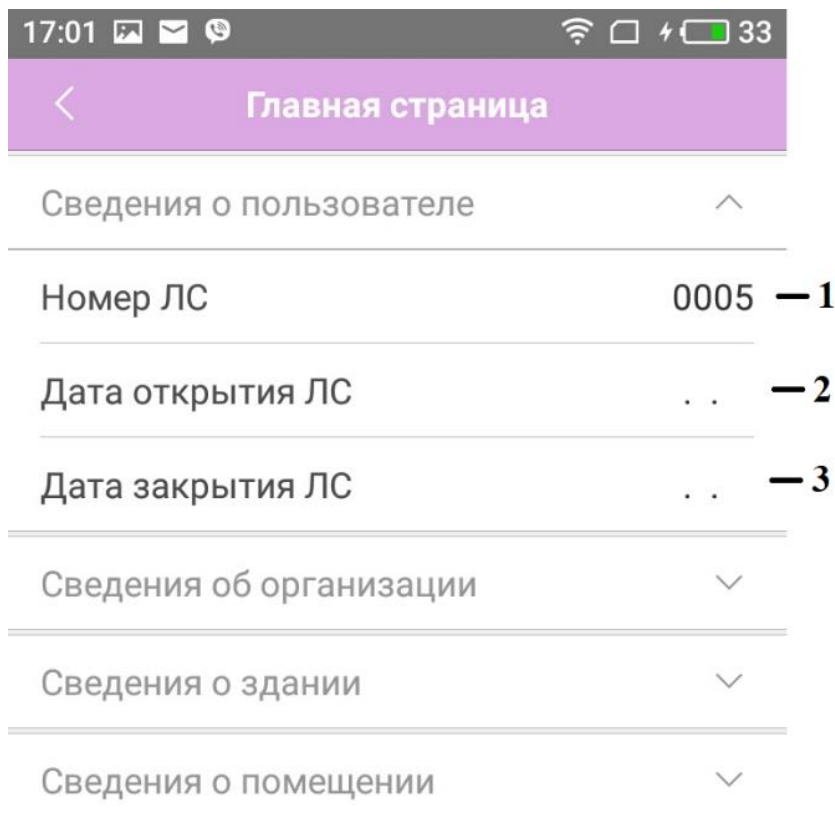


Рисунок 11 – Пользовательский интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения о пользователе»

1 – в текстовом поле «Номер ЛС» указан номер ЛС текущего пользователя;

2 – в текстовом поле «Дата открытия ЛС» указана дата открытия ЛС текущего пользователя;

3 – в текстовом поле «Дата закрытия ЛС» указана дата закрытия ЛС текущего пользователя.

Интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения об организации» показан на рисунке 12.

17:01 [Icons] 33

< Главная страница

Сведения о пользователе ▾

---

Сведения об организации ▴

---

Наименование ООО "Домучет" — 1

---

Адрес — 2

---

Номер телефона — 3

---

Email — 4

---

Сайт организации — 5

---

ИНН — 6

---

КПП — 7

---

Сведения о здании ▾

---

Сведения о помещении ▾

---

Рисунок 12 – Пользовательский интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения об организации»

1 – в текстовом поле «Наименование» указано наименование управляющей компании, которая обслуживает дом, в котором зарегистрирован текущий пользователь;

2 – в текстовом поле «Адрес» указан адрес управляющей компании, которая обслуживает дом, в котором зарегистрирован текущий пользователь;

3 – в текстовом поле «Номер телефона» указан контактный номер телефона управляющей компании, которая обслуживает дом, в котором зарегистрирован текущий пользователь;

4 – в текстовом поле «Email» указан контактный адрес электронной почты управляющей компании, которая обслуживает дом, в котором зарегистрирован текущий пользователь;

5 – в текстовом поле «Сайт организации» указан сайт управляющей компании, которая обслуживает дом, в котором зарегистрирован текущий пользователь;

6 – в текстовом поле «ИНН» указан идентификационный номер налогоплательщика управляющей компании, которая обслуживает дом, в котором зарегистрирован текущий пользователь;

7 – в текстовом поле «КПП» указан код причины постановки на учет управляющей компании, которая обслуживает дом, в котором зарегистрирован текущий пользователь.

Интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения о здании» показан на рисунке 13.

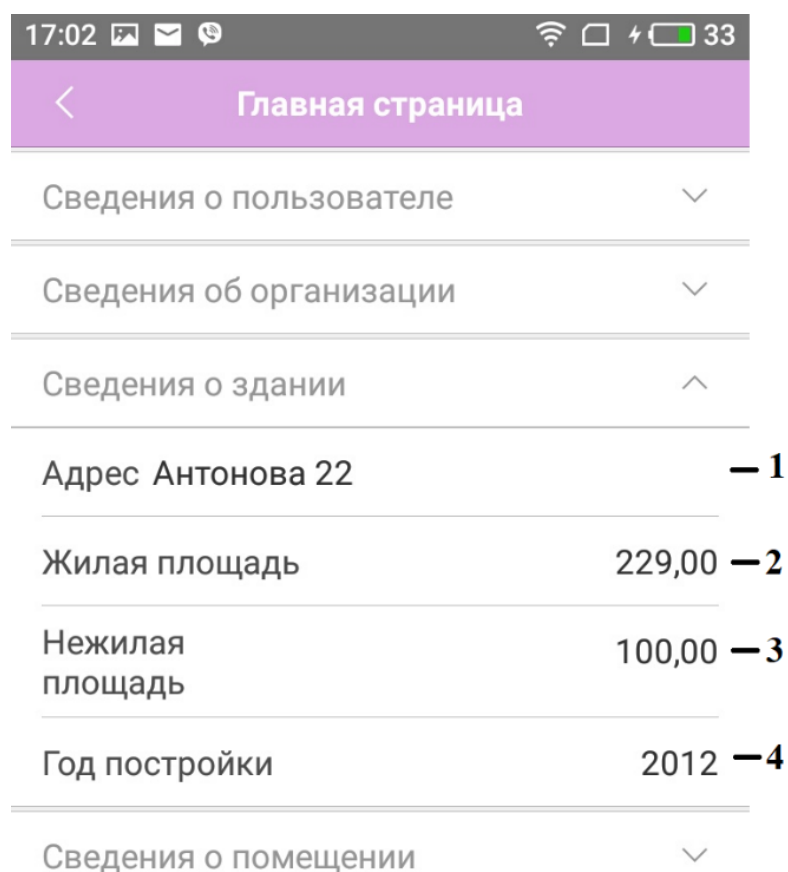


Рисунок 13 – Пользовательский интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения о здании»

1 – в текстовом поле «Адрес» указан адрес здания, в котором зарегистрирован текущий пользователь;

2 – в текстовом поле «Жилая площадь» указана жилая площадь здания, в котором зарегистрирован текущий пользователь;

3 – в текстовом поле «Нежилая площадь» указана нежилая площадь здания, в котором зарегистрирован текущий пользователь;

4 – в текстовом поле «Год постройки» указан год постройки здания, в котором зарегистрирован текущий пользователь.

Интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения о помещении» показан на рисунке 14.

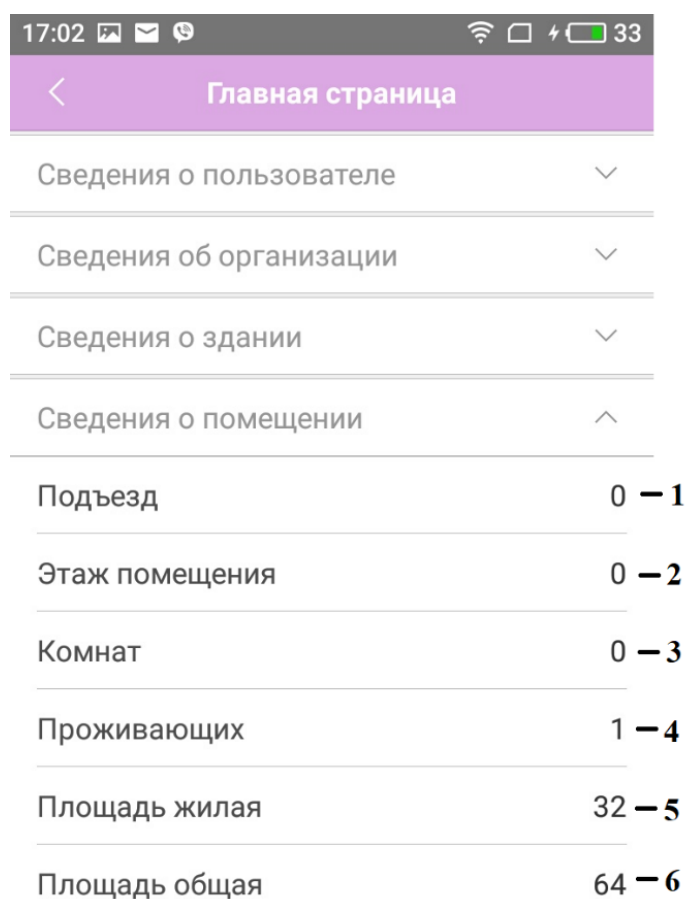


Рисунок 14 – Пользовательский интерфейс страницы с подробной информацией о пользователе, управляющей организации, здании и помещении, на которой раскрыта вкладка «Сведения о помещении»

1 – в текстовом поле «Подъезд» указан номер подъезда, в котором находится квартира, в которой зарегистрирован текущий пользователь;

2 – в текстовом поле «Этаж помещения» указан этаж, на котором находится квартира, в которой зарегистрирован текущий пользователь;

3 – в текстовом поле «Комнат» указано количество комнат в квартире, в которой зарегистрирован текущий пользователь;

4 – в текстовом поле «Проживающих» указано количество людей, которое проживает в квартире, в которой зарегистрирован текущий пользователь;

5 – в текстовом поле «Площадь жилая» указана жилая площадь квартиры, в которой зарегистрирован текущий пользователь;

6 – в текстовом поле «Площадь общая» указана общая площадь квартиры, в которой зарегистрирован текущий пользователь.

## **Выводы**

В соответствии с функциональными требованиями была выбрана среда разработки «1С: Предприятие». Платформа ориентирована на то, чтобы актуальные для задач автоматизации бизнеса технологии предоставить разработчику в готовом виде. Это позволяет разработчикам прикладных решений внедрять необходимые технологии своевременно, быстро и без радикальных изменений в разрабатываемом приложении.

Благодаря возможностям, предоставляемым платформой «1С: предприятие», в рамках проектирования мобильного программного обеспечения была спроектирована конфигурация мобильного приложения, а также спроектирован простой и интуитивно понятный графический пользовательский интерфейс. Это позволило повысить один из главных показателей качества для разрабатываемого приложения – удобство использования.

## **4 Тестирование и оценка качества МПО на разных этапах разработки**

Для оценки и повышения качества разрабатываемого МПО сформулированной методикой были проведены три оценки на разных этапах разработки:

- 1 – оценка качества МПО на раннем этапе разработки, выявление показателей с самыми низкими значениями;
- 2 – оценка качества МПО после доработок и устранения недостатков, выявленных после первой оценки;
- 3 – оценка финальной версии МПО после доработок и устранения недостатков, выявленных после второй оценки.

### **4.1 Оценка функциональности МПО**

К разрабатываемому МПО выдвигаются следующие требования функциональности:

- получение и просмотр информации о ЛС;
- получение и просмотр информации о здании;
- получение и просмотр информации об управляющей компании;
- получение и просмотр истории начислений и оплат;
- оплата выставленных счетов;
- получение и просмотр информации о счетчиках, подключенных к ЛС;
- получение и просмотр прошлых показаний счетчиков;
- ввод текущих показаний счетчиков;
- подача заявок в диспетчерскую службу управляющей компании.

#### **4.1.1 Оценка функциональности МПО на раннем этапе разработки**

Результаты оценки функциональности МПО после разработки первой стабильной версии:

- получение и просмотр информации о ЛС – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация о ЛС и для её вывода потребуется доработка);
- получение и просмотр информации о здании – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация о доме и для её вывода потребуется доработка);
- получение и просмотр информации об управляющей компании – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация об управляющей организации и для её вывода потребуется доработка);
- получение и просмотр истории начислений и оплат – оценка 2 (причины снижения оценки: реализовано получение необходимых данных, но не создана форма для вывода информации на экран пользователя, требуется серьезная доработка программного кода);
- оплата выставленных счетов – оценка 0 (причины снижения оценки: функция отсутствует в оцениваемой версии МПО);
- получение и просмотр информации о счетчиках, подключенных к ЛС – оценка 6 (причины снижения оценки: функция реализована, но требуется адаптация формы вывода информации под нужды пользователей);
- получение и просмотр прошлых показаний счетчиков – оценка 4 (причины снижения оценки: функция реализована частично, требуется доработка программного кода и создание формы вывода для отображения расшифровки каждого из показаний);
- ввод текущих показаний счетчиков – оценка 6 (причины снижения оценки: функция реализована удовлетворительно, но требуется адаптация формы ввода показаний под нужды пользователей);
- подача заявок в диспетчерскую службу управляющей компании – оценка 0 (причины снижения оценки: функция отсутствует в оцениваемой версии МПО).

Итоговая оценка функциональности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{8 + 8 + 8 + 2 + 0 + 6 + 4 + 6 + 0}{9} = 4,6.$$

Итоговая оценка за функциональность оцениваемой версии приложения – 4,6.

#### **4.1.2 Оценка функциональности МПО после доработок и устранения недостатков, выявленных после первой оценки**

Результаты оценки функциональности МПО после доработок и устранения недостатков, выявленных после первой оценки:

- получение и просмотр информации о ЛС – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация о ЛС и для её вывода потребуется доработка);
- получение и просмотр информации о здании – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация о доме и для её вывода потребуется доработка);
- получение и просмотр информации об управляющей компании – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация об управляющей организации и для её вывода потребуется доработка);
- получение и просмотр истории начислений и оплат – оценка 6 (причины снижения оценки: функция реализована, была создана форма для вывода информации, но требуется её адаптация под нужды пользователей);
- оплата выставленных счетов – оценка 2 (причины снижения оценки: функция была реализована частично, данные для оплаты приходят с сервера, но требуется серьёзная доработка программного кода для осуществления оплаты);
- получение и просмотр информации о счетчиках, подключенных к ЛС – оценка 8, была определена наиболее важная информация для пользователей, которую необходимо выводить, была доработана форма

вывода информации (причины снижения оценки: в перспективе может добавиться дополнительная информация о счетчиках и для её вывода потребуется доработка);

- получение и просмотр прошлых показаний счетчиков – оценка 10, была создана форма для вывода расшифровки платежей, была определена наиболее важная информация для вывода её пользователям, функция реализована полностью;

- ввод текущих показаний счетчиков – оценка 10, была доработана форма ввода показаний под нужды пользователей, функция реализована полностью;

- подача заявок в диспетчерскую службу управляющей компании – оценка 2, была добавлена форма для оформления заявок в диспетчерскую службу, была добавлена возможность отправки заявок в УК (причины снижения оценки: требуется серьёзная доработка программного кода для реализации возможности отслеживания созданных заявок и получения обратной связи по ним).

Итоговая оценка функциональности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{8 + 8 + 8 + 6 + 2 + 8 + 10 + 10 + 2}{9} = 6,8.$$

Итоговая оценка за функциональность оцениваемой версии приложения – 6,8.

#### **4.1.3 Оценка функциональности финальной версии МПО**

Результаты оценки функциональности финальной версии МПО:

- получение и просмотр информации о ЛС – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация о ЛС и для её вывода потребуется доработка);

- получение и просмотр информации о здании – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация о доме и для её вывода потребуется доработка);

– получение и просмотр информации об управляющей компании – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация об управляющей организации и для её вывода потребуется доработка);

– получение и просмотр истории начислений и оплат – оценка 10, была адаптирована форма для вывода информации, функция реализована полностью;

– оплата выставленных счетов – оценка 2 (причины снижения оценки: функция была реализована частично, данные для оплаты приходят с сервера, но требуется серьёзная доработка программного кода для осуществления оплаты);

– получение и просмотр информации о счетчиках, подключенных к ЛС – оценка 8 (причины снижения оценки: в перспективе может добавиться дополнительная информация о счетчиках и для её вывода потребуется доработка);

– получение и просмотр прошлых показаний счетчиков – оценка 10, функция реализована полностью;

– ввод текущих показаний счетчиков – оценка 10, функция реализована полностью;

– подача заявок в диспетчерскую службу управляющей компании – оценка 4 (причины снижения оценки: требуется незначительная доработка программного кода для получения обратной связи по оставленным заявкам).

Итоговая оценка функциональности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{8 + 8 + 8 + 10 + 2 + 8 + 10 + 10 + 4}{9} = 7,5.$$

Итоговая оценка за функциональность оцениваемой версии приложения – 7,5.

Результаты оценки всех версий МПО приведены в таблице 1.

Таблица 1 – Результаты оценок функциональности всех версий МПО

| Показатель                                                     | Версия МПО 1 | Версия МПО 2 | Финальная версия МПО |
|----------------------------------------------------------------|--------------|--------------|----------------------|
| Получение и просмотр информации о ЛС                           | 8            | 8            | 8                    |
| Получение и просмотр информации о здании                       | 8            | 8            | 8                    |
| Получение и просмотр информации об управляющей компании        | 8            | 8            | 8                    |
| Получение и просмотр истории начислений и оплат                | 2            | 6            | 10                   |
| Оплата выставленных счетов                                     | 0            | 2            | 2                    |
| Получение и просмотр информации о счетчиках, подключенных к ЛС | 6            | 8            | 8                    |
| Получение и просмотр прошлых показаний счетчиков               | 4            | 10           | 10                   |
| Ввод текущих показаний счетчиков                               | 6            | 10           | 10                   |
| Подача заявок в диспетчерскую службу управляющей компании      | 0            | 2            | 4                    |
| Итоговая оценка                                                | 4,6          | 6,8          | 7,5                  |

В ходе разработки мобильного программного обеспечения оценка за функциональность возросла с 4,6 до 7,5.

## 4.2 Оценка производительности МПО

Для анализа быстродействия приложения будет использовалась методика Ardex. Для расчета Ardex используются статистические данные с программных счетчиков, содержащие наименование операций, когда каждая из них инициирована, и длительность исполнения каждой операции приложением. Был определён список операций, в которых происходит получение, чтение и сохранение данных, которые больше всех влияют на общую производительность МПО:

- процедура получения и чтения общей информации;
- процедура получения и чтения информации с показаниями счетчиков (к ЛС, на котором происходило тестирование приложения, было подключено 3 счетчика);
- процедура получения и чтения информации о начислениях.

Перед оценкой производительности первой версии МПО значение целевого времени  $T$  было равно 1 секунде, значение  $F$  – 4 секундам. Для измерения времени выполнения использовался встроенный в платформу «1С: Предприятие» счетчик производительности. Было сделано 10 запусков приложения. Во время тестирования программы были выполнены основные операции, список которых был представлен выше, по три раза.

### 4.2.1 Оценка производительности МПО на раннем этапе разработки

Результаты оценки производительности МПО после разработки первой стабильной версии:

- процедура получения и чтения общей информации – среднее время выполнения 0,56 сек., количество вызовов процедуры со временем выполнения от 0 до  $T$  – 90, количество вызовов процедуры со временем от  $T$  до  $F$  – 0;
- процедура получения показаний счетчиков – среднее время выполнения 1,66 сек., количество вызовов процедуры со временем

выполнения от 0 до  $T - 0$ , количество вызовов процедуры со временем от  $T$  до  $F - 90$ ;

– процедура получения начислений – среднее время выполнения 1,81 сек., количество вызовов процедуры со временем выполнения от 0 до  $T - 0$ , количество вызовов процедуры со временем от  $T$  до  $F - 30$ .

Итоговая оценка производительности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{90 + \frac{120}{2}}{210} * 10 = 7,1.$$

Итоговая оценка производительности оцениваемой версии приложения – 7,1.

#### **4.2.2 Оценка производительности МПО после доработок и устранения недостатков, выявленных после первой оценки**

После первой оценки производительности было принято решение при первом получении данных сохранять их. Для этого были созданы регистры сведений: «СведенияОПользователе», «СведенияОЗдании», «СведенияОбОрганизации», «СведенияОПомещении», «СведенияОСчетчиках», «СведенияОНачислениях» и «ПоказанияСчетчиков». Так же было изменено время  $T$  и  $F$ :  $T = 0,5$  сек.,  $F = 2$  сек.

Результаты оценки функциональности МПО после доработок и устранения недостатков, выявленных после первой оценки:

– процедура получения и сохранения общей информации – среднее время выполнения 0,61 сек., количество вызовов процедуры со временем выполнения от 0 до  $T - 0$ , количество вызовов процедуры со временем от  $T$  до  $F - 10$ ;

– процедура чтения общей информации – среднее время выполнения 0,14 сек., количество вызовов процедуры со временем выполнения от 0 до  $T - 90$ , количество вызовов процедуры со временем от  $T$  до  $F - 0$ ;

– процедура получения и сохранения показаний счетчиков – среднее время выполнения 1,71 сек., количество вызовов процедуры со временем выполнения от 0 до T – 0, количество вызовов процедуры со временем от T до F – 10;

– процедура чтения показаний счетчиков – среднее время выполнения 0,26 сек., количество вызовов процедуры со временем выполнения от 0 до T – 90, количество вызовов процедуры со временем от T до F – 0;

– процедура получения и сохранения начислений – среднее время выполнения 1,90 сек., количество вызовов процедуры со временем выполнения от 0 до T – 0, количество вызовов процедуры со временем от T до F – 10;

– процедура чтения начислений – среднее время выполнения 0,35 сек., количество вызовов процедуры со временем выполнения от 0 до T – 30, количество вызовов процедуры со временем от T до F – 0.

Итоговая оценка производительности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{210 + \frac{30}{2}}{240} * 10 = 9,3.$$

Итоговая оценка производительности оцениваемой версии приложения – 9,3.

Благодаря использованию регистров сведений удалось увеличить скорость работы приложения и его оценку за производительность.

#### **4.2.3 Оценка производительности финальной версии МПО**

Дальнейшая работа с получением и сохранением данных не принесла заметного прироста производительности, поэтому было принято решение не проводить оценку производительности финальной версии приложения, и в качестве финальной оценки взять результаты второго тестирования.

Результаты оценки всех версий МПО приведены в таблице 2.

Таблица 2 – Результаты оценок производительности всех версий МПО

| Операция                                             | Количество операций |         |              |         |                      |         |
|------------------------------------------------------|---------------------|---------|--------------|---------|----------------------|---------|
|                                                      | Версия МПО 1        |         | Версия МПО 2 |         | Финальная версия МПО |         |
|                                                      | T = 1 с             | F = 4 с | T = 0,5 с    | F = 2 с | T = 0,5 с            | F = 2 с |
| Процедура получения и сохранения общей информации    | 90                  | 0       | 0            | 10      | 0                    | 10      |
| Процедура чтения общей информации                    |                     |         | 90           | 0       | 90                   | 0       |
| Процедура получения и сохранения показаний счетчиков | 0                   | 90      | 0            | 10      | 0                    | 10      |
| Процедура чтения показаний счетчиков                 |                     |         | 90           | 0       | 90                   | 0       |
| Процедура получения и сохранения начислений          | 0                   | 30      | 0            | 10      | 0                    | 10      |
| Процедура чтения начислений                          |                     |         | 30           | 0       | 30                   | 0       |
| Итоговая оценка                                      | 7,1                 |         | 9,3          |         | 9,3                  |         |

В ходе разработки мобильного программного обеспечения оценка за производительность возросла с 7,1 до 9,3.

### 4.3 Оценка удобства использования МПО

Оценка удобства использования проводилась социологическим методом с использованием анкет. Анкетирование было проведено среди десяти потенциальных пользователей МПО. В анкетах необходимо было выставить оценку от 0 до 3 для следующих параметров: понятность, удобство обучения, удобство работы. Также, если оценка по какому ни будь параметру была меньше 3, необходимо было написать причину снижения оценки.

### **4.3.1 Оценка удобства использования МПО на раннем этапе разработки**

Результаты оценки удобства использования после разработки первой стабильной версии:

- понятность – средняя оценка 1 (причины снижения оценки: непонятная система регистрации, непонятная система восстановления пароля для входа в систему);
- удобство обучения – средняя оценка 2,7 (причины снижения оценки: неудобное для запоминания расположение кнопок для просмотра и ввода показаний счетчиков);
- удобство работы – средняя оценка 2 (причины снижения оценки: неудобное расположение кнопок для просмотра и внесения показаний счетчиков, неудобное окно для просмотра подробной информации о ЛС, здании, организации, помещении, неудобная процедуры регистрации и смены пароля).

Итоговая оценка удобства использования оцениваемой версии приложения рассчитывается по формуле:

$$I = 1 + 2.7 + 2 = 5,7.$$

Итоговая оценка удобства использования оцениваемой версии – 5,7.

### **4.3.2 Оценка удобства использования МПО после доработок и устранения недостатков, выявленных после первой оценки**

После первой оценки был серьёзно доработан интерфейс авторизации пользователя. Были переписаны заголовки форм и полей, чтобы пользователям было проще ориентироваться в интерфейсе. Были добавлены отдельные кнопки для регистрации в сервисе и восстановления пароля. Каждый этап регистрации был оформлен в отдельном окне, чтобы пользователь на каждом этапе видел только нужные в данный момент поля. Была добавлена маска в поле для ввода мобильного телефона.

Результаты оценки удобства использования после доработок и устранения недостатков, выявленных после первой оценки:

- понятность – средняя оценка 3;
- удобство обучения – средняя оценка 2,7 (причины снижения оценки: неудобное для запоминания расположение кнопок для просмотра и ввода показаний счетчиков);
- удобство работы – средняя оценка 2 (причины снижения оценки: неудобное расположение кнопок для просмотра и внесения показаний счетчиков, неудобное окно для просмотра подробной информации о ЛС, здании, организации).

Итоговая оценка удобства использования оцениваемой версии приложения рассчитывается по формуле:

$$I = 3 + 2.7 + 2 = 7,7.$$

Итоговая оценка удобства использования оцениваемой версии – 7,7.

#### **4.3.3 Оценка удобства использования финальной версии МПО**

В финальной версии МПО были доработана форма для вывода общей информации по ЛС, зданию, помещению и управляющей компании. Информация по каждому объекту была сгруппирована в отдельные блоки, которые по умолчанию свернуты. При необходимости пользователь сможет раскрыть блок только с нужной информацией. Данные по текущей задолженности были выведены на главный экран. Кнопки для просмотра показаний и внесения счетчиков были добавлены на форму простора данных по выбранному счетчику.

Результаты оценки удобства использования финальной версии МПО:

- понятность – средняя оценка 3;
- удобство обучения – средняя оценка 3;
- удобство работы – средняя оценка 2,8 (причины снижения оценки: необходимо выполнить много действий для просмотра и внесения показаний счетчиков).

Итоговая оценка удобства использования оцениваемой версии приложения рассчитывается по формуле:

$$I = 3 + 3 + 2,8 = 8,8.$$

Результаты оценки всех версий МПО приведены в таблице 3.

Таблица 3 – Результаты оценок удобства использования всех версий МПО

| Параметры         | Версия МПО 1 | Версия МПО 1 | Финальная версия МПО |
|-------------------|--------------|--------------|----------------------|
| понятность        | 1            | 3            | 3                    |
| удобство обучения | 2,7          | 2,7          | 3                    |
| удобство работы   | 2            | 2            | 2,8                  |
| Итоговая оценка   | 5,7          | 7,7          | 8,8                  |

В ходе разработки мобильного программного обеспечения оценка за удобство использования возросла с 5,7 до 8,8.

#### 4.4 Оценка надежности МПО

Для оценки надежности МПО оно было запущено 30 раз. При каждом запуске проверялись все основные функции приложения. Каждый раз вводились разные данные, в разных форматах. Были попытки выполнить те или иные действия, которые требуют ввода информации, без заполнения соответствующих полей, либо с заполнением некорректными данными.

##### 4.4.1 Оценка надежности МПО на раннем этапе разработки

Результаты оценки надежности после разработки первой стабильной версии: количество запусков приложения – 30, количество запусков с неудовлетворительным результатом – 13.

Причины неудовлетворительных запусков: отсутствие проверки на формат мобильного телефона при авторизации пользователя, отсутствие проверки на окончание времени действия кода для регистрации, отсутствие проверки на время, в течении которого нельзя повторно посылать запрос на получение СМС с кодом авторизации, отсутствие проверки на тип пользователя, отсутствие проверки пароля пользователя, который он указывает при регистрации, на минимальную сложность, отсутствие отбора по

номеру ЛС в окне списка начислений и счетчиков, отсутствие процедуры обновления списка показаний счетчиков, после внесения новых.

Итоговая оценка надежности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{30 - 13}{30} * 10 = 5,6.$$

Итоговая оценка надежности оцениваемой версии приложения – 5,6.

#### **4.4.2 Оценка надежности МПО после доработок и устранения недостатков, выявленных после первой оценки**

Результаты оценки надежности после доработок и устранения недостатков, выявленных после первой оценки: количество запусков приложения – 30, количество запусков с неудовлетворительным результатом – 4. Была доработана процедура авторизации, регистрации и восстановления пароля. Была добавлена маска в поле ввода номера телефона. Были добавлены проверки на время действия кода для регистрации и на время, через которое можно повторно запросить СМС с кодом. Была добавлена проверка на сложность пароля, который пользователь вводит при регистрации. Была добавлена проверка на тип пользователя. Был добавлен отбор по id ЛС в списках начислений и счетчиков. Было добавлено автоматическое обновление показаний счетчиков, после добавления нового.

Причины неудовлетворительных запусков: ошибки при записи данных в регистры сведений, некорректный переход между разными окнами приложения.

Итоговая оценка надежности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{30 - 4}{30} * 10 = 8,6.$$

Итоговая оценка надежности оцениваемой версии приложения – 8,6.

#### 4.4.3 Оценка надежности финальной версии МПО

Результаты оценки надежности финальной версии МПО: количество запусков приложения – 30, количество запусков с неудовлетворительным результатом – 0. В финальной версии приложения были устранены все найденные ошибки, которые приводили к неудовлетворительному запуску приложения.

Итоговая оценка надежности оцениваемой версии приложения рассчитывается по формуле:

$$I = \frac{30}{30} * 10 = 10.$$

Итоговая оценка надежности оцениваемой версии приложения – 10.

Результаты оценки всех версий МПО приведены в таблице 4.

Таблица 4 – Результаты оценок надежности всех версий МПО

| Виды запусков приложения      | Версия МПО 1 | Версия МПО 2 | Финальная версия МПО |
|-------------------------------|--------------|--------------|----------------------|
| Всего запусков                | 30           | 30           | 30                   |
| Неудовлетворительных запусков | 13           | 4            | 0                    |
| Итоговая оценка               | 5,6          | 8,6          | 10                   |

В ходе разработки мобильного программного обеспечения оценка за надежность возросла с 5,6 до 10.

#### 4.5 Оценка безопасности МПО

Безопасность программного продукта оценивается по следующим параметрам: небезопасное хранение данных, небезопасная аутентификация, небезопасная передача данных, модификация данных, небезопасная авторизация.

##### 4.5.1 Оценка безопасности МПО на раннем этапе разработки

Результаты оценки безопасности после разработки первой стабильной версии:

– небезопасное хранение данных – МПО не подвержено данной угрозе (все критически важные данные хранятся в памяти только во время сеанса пользователя, пользователь не может получить к ним доступ из приложения или через локальные хранилища);

– небезопасная аутентификация – МПО не подвержено данной угрозе (после авторизации пользователя происходит проверка на состояние авторизации и на тип пользователя, полученные cookies сохраняются в параметры сеанса и хранятся там пока пользователь пользуется приложением, перед выполнением запросов к серверу происходит проверка актуальности сеанса);

– небезопасная передача данных – МПО не подвержено данной угрозе (при выполнении запросов к серверу используется защищенное SSL соединение, после авторизации все последующие запросы используют cookies для идентификации пользователя вместо логина и пароля, перед выполнением запросов к серверу, с ним устанавливается новое соединение и происходит проверка на корректность подключения);

– модификация данных – МПО не подвержено данной угрозе (приложение не изменяет исполняемые файлы, локальные ресурсы, не обращается к сторонним программам и не корректирует их данные);

– небезопасная авторизация – МПО не подвержено данной угрозе (на стороне клиента пароли вводятся в скрытом виде, они не хранятся в открытом виде на устройстве, для создаваемого пользователем пароля предусмотрены минимальные требования к безопасности).

Итоговая оценка безопасности оцениваемой версии приложения рассчитывается по формуле:

$$I = (1 + 1 + 1 + 1 + 1) * 2 = 10.$$

Итоговая оценка безопасности оцениваемой версии приложения – 10.

#### **4.5.2 Оценка безопасности МПО после доработок и устранения недостатков, выявленных после первой оценки**

Внесенные доработки никак не повлияли на безопасность МПО поэтому оценка осталась такой же, как и у МПО на раннем этапе разработки.

#### **4.5.3 Оценка безопасности финальной версии МПО**

Внесенные доработки никак не повлияли на безопасность МПО поэтому оценка осталась такой же, как и у МПО на раннем этапе разработки.

Результаты оценки всех версий МПО приведены в таблице 5.

Таблица 5 – Результаты оценок безопасности всех версий МПО

| Параметры                    | Версия МПО 1 | Версия МПО 2 | Финальная версия МПО |
|------------------------------|--------------|--------------|----------------------|
| небезопасное хранение данных | 1            | 1            | 1                    |
| небезопасная аутентификация  | 1            | 1            | 1                    |
| небезопасная передача данных | 1            | 1            | 1                    |
| модификация данных           | 1            | 1            | 1                    |
| небезопасная авторизация     | 1            | 1            | 1                    |
| Итоговая оценка              | 10           | 10           | 10                   |

В ходе разработки мобильного программного обеспечения оценка за безопасность не изменилась.

#### **4.6 Оценка удобства сопровождения МПО**

Удобство сопровождения программного продукта оценивается по следующим параметрам: анализируемость, удобство внесения изменений, стабильность, удобство проверки.

#### 4.6.1 Оценка удобства сопровождения МПО на раннем этапе разработки

Результаты оценки удобства сопровождения после разработки первой стабильной версии:

- анализируемость – оценка 1 (причины снижения оценки: процедуры и функции не были сгруппированы по смысловому признаку, у процедур и функций не было комментариев с описанием того, что они делают, и с описанием входных параметров);

- удобство внесения изменений – оценка 0 (причины снижения оценки: для каждого запроса к серверу использовалась отдельная функция в том модуле, откуда происходил вызов – это вызывало трудности при внесении изменений в запросы к серверу, т.к. приходилось исправлять функции, выполняющие этот запрос во всех модулях, где они использовались, для вывода и внесения данных использовались общие формы, что вызывало трудности при их адаптации к различным типам данных);

- стабильность – оценка 2 (причины снижения оценки: отсутствовали проверки на корректность получаемых данных от сервера и на коды ответа – это негативно сказывалось на стабильности, т.к. небольшие изменения в типе или составе возвращаемых сервером данных могли привести к ошибкам в работе программы);

- удобство проверки – оценка 1 (причины снижения оценки: для каждого запроса к серверу использовалась отдельная функция в том модуле, откуда происходил вызов – это вызывало трудности при тестировании, т.к. после внесения изменений в запросы к серверу приходилось проверять каждый модуль, в котором выполняются запросы).

Итоговая оценка удобства сопровождения оцениваемой версии приложения рассчитывается по формуле:

$$I = (1 + 0 + 2 + 1)/1,2 = 3,3.$$

Итоговая оценка удобства сопровождения оцениваемой версии приложения – 3,3.

#### **4.6.2 Оценка удобства сопровождения МПО после доработок и устранения недостатков, выявленных после первой оценки**

Результаты оценки удобства сопровождения после доработок и устранения недостатков, выявленных после первой оценки:

- анализируемость – оценка 1 (причины снижения оценки: процедуры и функции не были сгруппированы по смысловому признаку, у процедур и функций не было комментариев с описанием того, что они делают, и с описанием входных параметров);

- удобство внесения изменений – оценка 3 (были созданы общие модули, в них были добавлены все возможные функции с запросами к серверу, а также функции и процедуры, которые использовались в нескольких модулях, теперь, вместо выполнения одного и того-же программного кода в разных модулях, достаточно вызвать необходимую функцию или процедуру через указание имени общего модуля – это значительно упрощает внесение изменений, т.к. достаточно внести доработки только в одном общем модуле, большинство общих форм было заменено формами объектов, к которым относятся обрабатываемые данные – это упростило процесс создания и редактирования форм);

- стабильность – оценка 3 (в каждую функцию с запросом к серверу были добавлены проверки на тип и состав получаемых данных – это позволит вносить изменения в состав и тип возвращаемых сервером данных без риска нарушить стабильную работу МПО);

- удобство проверки – оценка 3 (т.к. теперь все процедуры и функции, которые использовались в нескольких модулях или делали запрос к серверу, находятся в общих модулях, то тестирование приложения значительно упростилось, теперь, если возникла ошибка в какой-либо общей процедуре или функции, не нужно проверять их работу во всех остальных модулях форм, достаточно исправить ошибку в общем модуле).

Итоговая оценка удобства сопровождения оцениваемой версии приложения рассчитывается по формуле:

$$I = (1 + 3 + 3 + 3)/1,2 = 8,3.$$

Итоговая оценка удобства сопровождения оцениваемой версии приложения – 8,3.

#### 4.6.3 Оценка удобства сопровождения финальной версии МПО

Результаты оценки удобства сопровождения финальной версии МПО:

- анализируемость – оценка 3 (для каждой процедуры и функции были добавлены комментарии с описанием их функционала, типом и содержанием их входных параметров, также они были сгруппированы в именованные области по смысловому признаку для повышения анализируемости кода);

- удобство внесения изменений – оценка 3;

- стабильность – оценка 3;

- удобство проверки – оценка 3.

Итоговая оценка удобства сопровождения оцениваемой версии приложения рассчитывается по формуле:

$$I = (3 + 3 + 3 + 3)/1,2 = 10.$$

Итоговая оценка удобства сопровождения оцениваемой версии приложения – 10.

Результаты оценки всех версий МПО приведены в таблице 6.

Таблица 6 – Результаты оценок удобства сопровождения всех версий МПО

| Параметры                   | Версия МПО 1 | Версия МПО 2 | Финальная версия МПО |
|-----------------------------|--------------|--------------|----------------------|
| анализируемость             | 1            | 1            | 3                    |
| удобство внесения изменений | 0            | 3            | 3                    |
| стабильность                | 2            | 3            | 3                    |
| удобство проверки           | 1            | 3            | 3                    |
| Итоговая оценка             | 3,3          | 8,3          | 10                   |

В ходе разработки мобильного программного обеспечения оценка за удобство сопровождения возросла с 3,3 до 10.

## **4.7 Оценка переносимости МПО**

Переносимость программного продукта оценивается по следующим параметрам: адаптируемость, удобство установки, способность к сосуществованию, удобство замены другого ПО данным.

### **4.7.1 Оценка переносимости МПО на раннем этапе разработки**

Благодаря использованию платформы «1С: Предприятие» для разработки МПО за адаптируемость, удобство установки и способность к сосуществованию были выставлены максимальные оценки, т.к. все необходимые действия для адаптации программного обеспечения под разные среды платформа выполняет автоматически, без необходимости внесения сторонних доработок.

Результаты оценки переносимости после разработки первой стабильной версии:

- адаптируемость – оценка 3 (приложение, написанное на платформе «1С: Предприятие», способно адаптироваться под любой размер экрана и под любую его ориентацию, способно работать на любом устройстве, которое подходит под минимальные требования);
- удобство установки – оценка 3 (приложение не требует дополнительных разрешений при установке и во время работы, для его установки не требует производить дополнительных настроек, после установки оно сразу готово к работе);
- способность к сосуществованию – оценка 3 (приложение способно функционировать, не нарушая работу устройства и других приложений, оно не использует и не изменяет файлы другого ПО);
- удобство замены другого ПО данным – оценка 1 (причины снижения оценки: в приложении ещё не реализована система подачи заявок в диспетчерскую службу и оплаты задолженности, требуются серьёзные доработки).

Итоговая оценка переносимости оцениваемой версии приложения рассчитывается по формуле:

$$I = (3 + 3 + 3 + 1)/1,2 = 8,3.$$

Итоговая оценка переносимости оцениваемой версии приложения – 8,3.

#### **4.7.2 Оценка переносимости МПО после доработок и устранения недостатков, выявленных после первой оценки**

Результаты оценки переносимости после доработок и устранения недостатков, выявленных после первой оценки:

- адаптируемость – оценка 3;
- удобство установки – оценка 3;
- способность к сосуществованию – оценка 3;
- удобство замены другого ПО данным – оценка 1 (причины снижения оценки: в приложении не до конца реализована система подачи заявок в диспетчерскую службу и оплаты задолженности, требуются серьёзные доработки).

Итоговая оценка переносимости оцениваемой версии приложения рассчитывается по формуле:

$$I = (3 + 3 + 3 + 1)/1,2 = 8,3.$$

Итоговая оценка переносимости оцениваемой версии приложения – 8,3.

#### **4.7.3 Оценка переносимости финальной версии МПО**

Результаты оценки переносимости финальной версии МПО:

- адаптируемость – оценка 3;
- удобство установки – оценка 3;
- способность к сосуществованию – оценка 3;
- удобство замены другого ПО данным – оценка 1 (причины снижения оценки: в приложении не до конца реализована система подачи заявок в диспетчерскую службу и оплаты задолженности, требуются серьёзные доработки).

Итоговая оценка переносимости оцениваемой версии приложения рассчитывается по формуле:

$$I = (3 + 3 + 3 + 1)/1,2 = 8,3.$$

Итоговая оценка переносимости оцениваемой версии приложения – 8,3.

Результаты оценки всех версий МПО приведены в таблице 7.

Таблица 7 – Результаты оценок переносимости всех версий МПО

| Параметры                         | Версия МПО 1 | Версия МПО 2 | Финальная версия МПО |
|-----------------------------------|--------------|--------------|----------------------|
| адаптируемость                    | 3            | 3            | 3                    |
| удобство установки                | 3            | 3            | 3                    |
| способность к сосуществованию     | 3            | 3            | 3                    |
| удобство замены другого ПО данным | 1            | 1            | 1                    |
| Итоговая оценка                   | 8,3          | 8,3          | 8,3                  |

В ходе разработки мобильного программного обеспечения оценка за переносимость не изменилась.

#### 4.8 Оценка информационных характеристик МПО

Для МПО было написано описание его функциональных возможностей, которое размещено на сайте <https://domuchet.online/>. Текст описания: «Приложение для мобильных телефонов, которое позволяет жильцу:

- просматривать начисления за коммунальные услуги / капитальный ремонт;
- просматривать квитанции за прошлые периоды;
- оплатить задолженность из приложения;
- отправить показания квартирных счетчиков;
- отправить обращение в диспетчерскую службу».

Оценка этого описания – 1 (причины снижения оценки: в оцениваемых версиях МПО не реализована до конца система оплаты задолженности и подача заявок в диспетчерскую службу).

#### **4.9 Ранжирование показателей качества и получение их весов с помощью метода анализа иерархий**

Для МПО показатели качества были ранжированы в порядке убывания важности следующим образом:

- функциональность – самый важный параметр, основная цель этого приложения – облегчить выполнение ежемесячных рутинных задач, таких как оплата задолженностей, ввод показаний счетчиков, просмотр истории начислений без необходимости хранить квитанции, подача заявок в диспетчерскую службу без звонков и личного посещения управляющей компании, так же, это МПО позволит ускорить работу управляющей компании, так как данные от жильцов будут приходить быстрее и оперативнее, из этого следует, что чем больше функциональных возможностей будет реализовано в приложении, тем легче будет взаимодействие жильцов с управляющей компанией и тем быстрее УК будет решать возникающие проблемы;
- удобство использования – приложение рассчитано на пользователей разных возрастов и с разными навыками владения мобильным устройством, именно поэтому удобство использования – один из важнейших показателей качества, ведь МПО будет использовать широкий круг пользователей;
- надежность – приложение рассчитано на пользователей разных возрастов и с разными навыками владения мобильным устройством, именно поэтому в приложении должно быть предусмотрено максимально возможное количество проверок, чтобы пользователь, по не знаю, не смог вызвать сбой в работе МПО, указав некорректные данные;
- переносимость – предполагается, что МПО будет использовано на большом количестве разнообразных устройств, поэтому важно, чтобы оно безошибочно устанавливалось и работало на всех устройствах, которые подходят под минимальные требования;

– информационные характеристики – из описания сотрудникам УК должно быть понятно, какие преимущества можно получить, подключив в сервисе МПО, а жителям должно быть понятно, какие задачи они смогут выполнять быстрее и эффективнее, запросив доступ к мобильному приложению;

– производительность – основные расчеты выполняются на сервере, также все данные пользователей хранятся на сервере и загружаются в приложение по необходимости, поэтому производительность МПО не является важным показателем качества;

– безопасность – МПО работает с данными пользователей, которые не могут быть использованы злоумышленниками для получения выгоды или нанесения вреда, поэтому безопасность не является важным показателем качества данного приложения;

– удобство сопровождения – весь основной функционал для МПО уже был придуман и реализован, дальнейшие доработки приложения будут направлены в основном на исправление ошибок, а не на добавление нового функционала, поэтому удобство сопровождения не является важным показателем качества.

После ранжирования показателей качества по степени важности им были присвоены веса при помощи метода анализа иерархий. Для этого была построена матрица с отношением критериев друг к другу.

Таблица 8 – Таблица с отношением критериев

|                  | функциональность | удобство использования | надежность | переносимость | информационные характеристики | производительность | безопасность | удобство сопровождения |
|------------------|------------------|------------------------|------------|---------------|-------------------------------|--------------------|--------------|------------------------|
| функциональность | 1                | 1,125                  | 1,285      | 1,5           | 1,8                           | 2,25               | 3            | 4,5                    |

Продолжение таблицы 8

|                               |       |       |       |       |     |      |        |     |
|-------------------------------|-------|-------|-------|-------|-----|------|--------|-----|
| удобство использования        | 0,888 | 1     | 1,142 | 1,333 | 1,6 | 2    | 2,666  | 4   |
| надежность                    | 0,777 | 0,875 | 1     | 1,166 | 1,4 | 1,75 | 2,333  | 7,5 |
| переносимость                 | 0,666 | 0,75  | 0,857 | 1     | 1,2 | 1,5  | 2      | 3   |
| информационные характеристики | 0,555 | 0,625 | 0,714 | 0,833 | 1   | 1,25 | 1,666  | 2,5 |
| производительность            | 0,444 | 0,5   | 0,571 | 0,666 | 0,8 | 1    | 1,333  | 2   |
| безопасность                  | 0,333 | 0,375 | 0,428 | 0,5   | 0,6 | 0,75 | 1      | 1,5 |
| удобство сопровождения        | 0,222 | 0,25  | 0,285 | 0,333 | 0,4 | 0,5  | 0,666  | 1   |
| Сумма отношений               | 4,885 | 5,5   | 6,282 | 7,331 | 8,8 | 11   | 14,664 | 26  |

Далее необходимо поделить все элементы матрицы на сумму элементов соответствующего столбца и найти среднее значение для каждой строки.

Таблица 9 – Таблица с весами критериев

|                               | функциональность | удобство использования | надежность | переносимость | информационные характеристики | производительность | безопасность | удобство сопровождения | Среднее значение |
|-------------------------------|------------------|------------------------|------------|---------------|-------------------------------|--------------------|--------------|------------------------|------------------|
| функциональность              | 0,205            | 0,205                  | 0,205      | 0,205         | 0,205                         | 0,205              | 0,205        | 0,205                  | 0,205            |
| удобство использования        | 0,182            | 0,182                  | 0,182      | 0,182         | 0,182                         | 0,182              | 0,182        | 0,182                  | 0,182            |
| надежность                    | 0,159            | 0,159                  | 0,159      | 0,159         | 0,159                         | 0,159              | 0,159        | 0,159                  | 0,159            |
| переносимость                 | 0,136            | 0,136                  | 0,136      | 0,136         | 0,136                         | 0,136              | 0,136        | 0,136                  | 0,136            |
| информационные характеристики | 0,114            | 0,114                  | 0,114      | 0,114         | 0,114                         | 0,114              | 0,114        | 0,114                  | 0,114            |
| производительность            | 0,091            | 0,091                  | 0,091      | 0,091         | 0,091                         | 0,091              | 0,091        | 0,091                  | 0,091            |
| безопасность                  | 0,068            | 0,068                  | 0,068      | 0,068         | 0,068                         | 0,068              | 0,068        | 0,068                  | 0,068            |
| удобство сопровождения        | 0,045            | 0,045                  | 0,045      | 0,045         | 0,045                         | 0,045              | 0,045        | 0,045                  | 0,045            |

Получили веса для каждого показателя качества:

- функциональность – 0,205;
- удобство использования – 0,182;
- надежность – 0,159;
- переносимость – 0,136;

- информационные характеристики – 0,114;
- производительность – 0,091;
- безопасность – 0,068;
- удобство сопровождения – 0,045.

#### 4.10 Итоговая оценка приложения

Были протестированы 3 версии приложения на разных этапах разработки. Для каждого показателя был определён его вес. Результаты оценки каждой версии МПО по каждому критерию представлены в таблице 10.

Таблица 10 – Финальные оценки каждого критерия каждой версии МПО

|                                  | Версия МПО<br>1 | Версия МПО<br>2 | Финальная<br>версия МПО | Веса<br>показателей |
|----------------------------------|-----------------|-----------------|-------------------------|---------------------|
| функциональность                 | 4,6             | 6,8             | 7,5                     | 0,205               |
| удобство<br>использования        | 5,7             | 7,7             | 8,8                     | 0,182               |
| надежность                       | 5,6             | 8,6             | 10                      | 0,159               |
| переносимость                    | 8,3             | 8,3             | 8,3                     | 0,136               |
| информационные<br>характеристики | 1               | 1               | 1                       | 0,114               |
| производительность               | 7,1             | 9,3             | 9,3                     | 0,091               |
| безопасность                     | 10              | 10              | 10                      | 0,068               |
| удобство<br>сопровождения        | 3,3             | 8,3             | 10                      | 0,045               |

Для каждой версии МПО была произведена общая оценка по всем критериям. Результаты финальных оценок качества мобильного программного обеспечения приведены в таблице 11.

Таблица 11 – Итоговые оценки каждой версии МПО

|                              | Версия МПО 1 | Версия МПО 2 | Финальная версия<br>МПО |
|------------------------------|--------------|--------------|-------------------------|
| Финальная оценка<br>качества | 5,6          | 7,3          | 7,9                     |

В ходе разработки мобильного программного обеспечения итоговая оценка качества увеличилась с 5,6 до 7,9.

## **Выводы**

Было протестировано мобильное программное обеспечение с учетом его особенностей и функциональных возможностей на разных этапах разработки. При каждом тестировании были определены факторы, которые способствовали снижению оценки. Это позволило по ходу разработки находить и устранять критические ошибки, недочеты, повышать производительность, безопасность, удобство использования.

После тестирования на каждом этапе выполнялся расчет общей оценки качества рассматриваемой версии. Благодаря этому можно было отслеживать общее повышение качества разрабатываемого продукта. Из этого можно сделать вывод, что тестирование и оценка качества мобильного программного обеспечения на разных этапах разработки положительно сказывается на итоговом качестве разрабатываемого продукта.

## Заключение

В результате выполнения диссертационной работы была сформулирована методика оценки качества мобильного программного обеспечения, также был проведён расширенный и углубленный критический анализ предметной области, разработано и реализовано мобильное приложение для сервиса «DOMUCHET.ONLINE».

Основные результаты диссертационной работы состоят в следующем.

Проведены исследования и анализ методов оценки показателей качества мобильного программного обеспечения, был рассмотрен и изучен метод анализа иерархий.

Разработана методика оценки качества мобильного программного обеспечения, которая учитывает особенности оцениваемого приложения и позволяет на ранних этапах разработки выявлять и устранять проблемные места для повышения качества готового продукта.

Разработаны программные средства на платформе «1С: Предприятие» для практического применения сформулированного метода и оценки его эффективности.

Разработанное МПО планируется в дальнейшем использовать в сервисе «DOMUCHET.ONLINE».

### Список использованных источников

1. Digital 2020: 3.8 billion people use social media [Электронный ресурс]. URL: <https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media> (дата обращения: 20.02.2020)
2. Вся статистика интернета на 2020 год — цифры и тренды в мире и в России [Электронный ресурс]. URL: <https://zen.yandex.ru/media/id/5d5d45710ce57b00aeb7070c/vsia-statistika-interneta-na-2020-god--cifry-i-trendy-v-mire-i-v-rossii-5e8a4d2b9858d7030b845ee4> (дата обращения: 21.03.2020)
3. Российская система качества. Мобильные приложения для смартфонов. Специальные требования к качеству [Электронный ресурс]. URL: [http://d-russia.ru/wp-content/uploads/2017/04/Roskachestvo\\_mob\\_app\\_3-04-2017.pdf](http://d-russia.ru/wp-content/uploads/2017/04/Roskachestvo_mob_app_3-04-2017.pdf) (дата обращения: 21.03.2020)
4. Измерительные методы [Электронный ресурс]. URL: [https://studbooks.net/63827/tovarovedenie/izmeritelnye\\_metody](https://studbooks.net/63827/tovarovedenie/izmeritelnye_metody) (дата обращения: 22.03.2020)
5. Регистрационный метод [Электронный ресурс]. URL: <http://900igr.net/prezentatsii/ekonomika/Kvalimetrija/014-Registratsionnyj-metod.html> (дата обращения: 22.03.2020)
6. Органолептические методы [Электронный ресурс]. URL: [https://studopedia.ru/7\\_163400\\_organolepticheskie-metodi.html](https://studopedia.ru/7_163400_organolepticheskie-metodi.html) (дата обращения: 22.03.2020)
7. Методы определения показателей качества [Электронный ресурс]. URL: [https://studbooks.net/1556128/marketing/metody\\_opredeleniya\\_pokazateley\\_kachestva](https://studbooks.net/1556128/marketing/metody_opredeleniya_pokazateley_kachestva) (дата обращения: 22.03.2020)
8. Метод экспертных оценок [Электронный ресурс]. URL: <https://center-yf.ru/data/Marketologu/Metod-ekspertnyh-ocenok.php> (дата обращения: 04.04.2020)

9. Метод анализа иерархий [Электронный ресурс]. URL: <https://www.hse.ru/mirror/pubs/share/204810376> (дата обращения: 04.04.2020)
10. Проектирование информационных систем [Электронный ресурс]. URL: <https://www.intuit.ru/studies/courses/2195/55/lecture/15051> (дата обращения: 11.04.2020)
11. Методика APDEX – стандарт оценки производительности корпоративных приложений [Электронный ресурс]. URL: <http://www.gilev.ru/apdex-teoriya/> (дата обращения: 11.04.2020)
12. Технологии программирования. Компонентный подход [Электронный ресурс]. URL: <http://panda.ispras.ru/~RedVerst/RedVerst/Lectures%20and%20training%20courses/Software%20Development%20Technologies/Lecture05.pdf> (дата обращения: 25.04.2020)
13. Оценка надежности программного обеспечения [Электронный ресурс]. URL: <https://www.osp.ru/os/1994/04/178540> (дата обращения: 25.04.2020)
14. Исследование качества мобильных приложений [Электронный ресурс]. URL: [http://journal.tc22.ru/wp-content/uploads/2018/02/issledovanie\\_kachestva\\_mobilnih\\_prilozheniy.pdf](http://journal.tc22.ru/wp-content/uploads/2018/02/issledovanie_kachestva_mobilnih_prilozheniy.pdf) (дата обращения: 25.04.2020)
15. 1С-Публишинг «Профессиональная разработка в системе 1С:Предприятие 8» – М.: 1С-Публишинг, 2012. – 689 с. (дата обращения: 10.05.20)
16. Хрусталева Е. Ю., Знакомство с разработкой мобильных приложений на платформе «1С:Предприятие 8». / Хрусталева Е. Ю., Издательство 1С-Публишинг, 2012, 276 с. (дата обращения: 10.05.20)
17. Рыбалка В.В., Mobile 1С. Пример быстрой разработки мобильного приложения на платформе 1С:Предприятие 8.3. Версия 1/Рыбалка В.В., Из-во 1С-Публишинг, 2014, 215 с. (дата обращения: 10.05.20)

18. Кузин Д.О., Афонин А.Ю., Оценка качества мобильного программного обеспечения // Международный журнал: Цифровая наука, – Саратов, 2020. – с. 128 – 144. (дата обращения: 06.05.2020)
19. Меликов А.В., Макарычев П.П., Модели предметной области систем электронного анкетирования // Психолого-педагогический журнал Гаудеамус, – Тамбов, 2010. – с. 59 – 60. (дата обращения: 06.05.2020)
20. Шибанов С.В., Мезенков А.А., Макарычев П.П., Применение метаданных в адаптивных информационных системах клиент-серверной архитектуры // Труды международного симпозиума надежность и качество, – Пенза, 2010. – с. 120 – 125. (дата обращения: 06.05.2020)
21. Селищев Н. Администрирование системы 1С: Предприятие 8.2,\ Селищев Н., Издательство Питер, 2012, 325 с. (дата обращения: 11.05.20)
22. Филиппов Е.В. , Настольная книга 1С: Эксперта по технологическим вопросам/ Филиппов Е.В. , Издательство 1С-Публишинг, 2015, 312 с. (дата обращения: 11.05.20)
23. Асатрян, А., Методическое пособие по эксплуатации крупных информационных систем на платформе 1С: Предприятие 8: учебник/ А. Асатрян , А. Голиков, А. Морозов, Издательство 1С-Публишинг, 2017, 220 с. (дата обращения: 16.05.20)
24. Ажеронок, В.А. «1С:Предприятие 8» / В.А. Ажеронок, А.В. Островерх, М.Г. Радченко, Е.Ю. Хрусталева. – М., Издательство «1С – Публишинг», 2018. – 902 с. (дата обращения: 16.05.20)
25. 1С Предприятие: 8.2. Руководство администратора – М.: ООО «1С – Публишинг», 2009. – 216 с. (дата обращения: 16.05.20)
26. Архитектура системы программ «1С: Предприятие». Фирма «1С». Москва, 2003 г. – 312 с. (дата обращения: 17.05.20)
27. Основные механизмы платформы «1С: Предприятие 8.2». Арутюнов С.Р. – М.: ЗАО «1С», 2010 – 426 с. (дата обращения: 17.05.20)

28. Хрусталева Е.Ю., Знакомство с разработкой мобильных приложений на платформе «1С:Предприятие 8». / Хрусталева Е. Ю., Издательство 1С-Паблишинг, 2012, 276 с. (дата обращения: 25.05.20)

29. Жарко Е.Ф., Сравнение моделей качества программного обеспечения: аналитический подход // XII Всероссийское совещание по проблемам управления ВСПУ, – Москва, 2014. – с. 4585 – 4594. (дата обращения: 06.06.2020)

30. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов [Электронный ресурс]. URL: <http://docs.cntd.ru/document/1200121069> (дата обращения: 06.06.2020)

31. Ларин С.Н., Лазарева Л.Ю., Ларина Т.С., Модели, методы, показатели, характеристики и метрики, применяемые в экспертных системах оценки качества разработки и создания инновационных программных проектов // Региональная экономика: теория и практика, – Москва, 2017. – с. 1187 – 1198. (дата обращения: 07.06.2020)

32. Горбаченко И.М., Оценка качества программного обеспечения для создания систем тестирования // Научный журнал Фундаментальные исследования, – Москва, 2013. – с. 823 – 827. (дата обращения: 07.06.2020)

33. Качество программного обеспечения [Электронный ресурс]. URL: <https://analytics.infozone.pro/software-quality/> (дата обращения: 07.06.2020)

34. Тестирование на основе моделей [Электронный ресурс]. URL: <http://panda.ispras.ru/~kuliamin/lectures-mbt/Lecture01.pdf> (дата обращения: 07.06.2020)

35. Моисеев С.И., Черная Ю.В., Паршина Е.В., Модель оценки качества программного обеспечения, основанная на методе Раша оценки латентных переменных // Вестник ВГУ, серия: системный анализ и информационные технологии, - Воронеж, 2016. – с. 102 – 109. (дата обращения: 10.06.2020)

36. Качество ПО и методы его контроля [Электронный ресурс]. URL: <https://www.intuit.ru/studies/courses/64/64/lecture/1874> (дата обращения: 10.06.2020)

37. Белик А.Г., Качество и надежность программных систем // Минобрнауки России, ОмГТУ, – Омск, 2018. – 80 с. (дата обращения: 13.06.2020)

38. Стандартизация и сертификация программного обеспечения [Электронный ресурс]. URL: <https://www.sibsau.ru/sveden/edufiles/144332/> (дата обращения: 13.06.2020)

39. Управление качеством программного обеспечения [Электронный ресурс]. URL: <https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/izuchite-upravlenie-kachestvom-programmnogo-obespecheniia/upravlenie-kachestvom-programmnogo-obespecheniia-kratkoe-rukovodstvo> (дата обращения: 14.06.2020)

40. Качество программного обеспечения [Электронный ресурс]. URL: [https://studref.com/320312/informatika/kachestvo\\_programmnogo\\_obespecheniya](https://studref.com/320312/informatika/kachestvo_programmnogo_obespecheniya) (дата обращения: 14.06.2020)

41. Современные модели качества программного обеспечения [Электронный ресурс]. URL: <http://www.interface.ru/fset.asp?Url=/misc/qs.htm> (дата обращения: 14.06.2020)

42. Модели качества программного обеспечения и методы его оценивания [Электронный ресурс]. URL: <https://ppt-online.org/364847> (дата обращения: 15.06.2020)

43. Критерии качества программного обеспечения [Электронный ресурс]. URL: <https://www.bestreferat.ru/referat-315051.html> (дата обращения: 17.06.2020)

44. Характеристики качества программного обеспечения и методы их оценки [Электронный ресурс]. URL: <http://www.ibtrans.ru/upload/iblock/79b/79b3293930c9d4d0c38c815d97c954fa.pdf> (дата обращения: 17.06.2020)

**Приложение А**  
**ГЛОССАРИЙ**  
**(справочное)**

1С: Предприятие – программный продукт компании 1С, включающий платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц.

Конфигурация – это набор алгоритмов и форм, написанных на платформе 1С, для учета хозяйственной деятельности компании.

Конфигуратор – это программа для редактирования конфигурации 1С. Конфигуратор позволяет выполнять функции разработки, отладки, обновления, администрирования базы данных.

Android – операционная система для смартфонов, планшетов, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, и других устройств. В настоящее время большинство смартфонов мира функционируют именно на этой ОС. Первая версия Android вышла в 2008 году, вместе с первым полноценным пакетом для разработки приложений SDK 1.0.

Вес – это числовой показатель, выражающий относительную важность признака.

**Приложение Б**  
**КОД ПРОГРАММЫ**  
**(рекомендуемое)**

## Листинг Б.1 – Код общего модуля «APIПроцедурыИФункции»

```

#Область API
#Область Авторизация
//Вход пользователя
Функция  ВходПользователя(Логин      =      Неопределено,      Пароль      =
Неопределено) Экспорт

    Если Логин = Неопределено Тогда
        Логин = ПолучитьЛогинПользователя();
    КонецЕсли;
    Если Пароль = Неопределено Тогда
        Пароль = ПолучитьПарольПользователя();
    КонецЕсли;
    //Создание соединения
    Попытка
        Соединение = Новый
        HTTPСоединение("develop.domuchet.online", 443,,,, Новый
        ЗащищенноеСоединениеOpenSSL);
    Искключение
        Сообщить(ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Вход пользователя ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();
    Запрос = Новый
    HTTPЗапрос("/api/auth?command=login&login=" + СокрЛП(Логин) +
    "&password=" + СокрЛП(Пароль));
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
    ФайлДляОтвета);
    Если ОтветНаЗапрос.КодСостояния = 200 Тогда
        ТекстОтвета = Новый ЧтениеJSON;
        ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
        Ответ = ПрочитатьJSON(ТекстОтвета);
        ТекстОтвета.Закрыть();
        Если Ответ.status.success Тогда
            //При успешной авторизации записываем в параметр
сеанса "CookieПользователя" и в заголовок Cookie для выполнения
дальнейших запросов
            ВзаимодействиеСПараметрамиСеанса.УстановитьCookieПользовател
я(ОтветНаЗапрос.Заголовки["Set-Cookie"]);
        Иначе
            Сообщить(Ответ.status.message);
            Соединение = Неопределено;
            Возврат Ложь;
        КонецЕсли;
    Иначе
        Сообщить("Возникла ошибка при входе пользователя. Код
состояния: " + ОтветНаЗапрос.КодСостояния);
        Соединение = Неопределено;
        Возврат Ложь;
    КонецЕсли;
    //Удаляем временный файл, чтобы не засорять память
    ВременныйФайл = Новый Файл(ФайлДляОтвета);

```

```

Если ВременныйФайл.Существует() Тогда
    Попытка
        УдалитьФайлы(ФайлДляОтвета);
    Исключение
        Сообщить(ОписаниеОшибки());
    КонецПопытки;
КонецЕсли;
//Вход пользователя --
//Удаление соединения
Соединение = Неопределено;
Возврат Истина;
КонецФункции // АвторизацияПользователя()

//Подтверждение номера телефона
Функция ВходПользователяПоНомеруТелефона(КодАвторизации) Экспорт
    CookieПользователя =
ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
    Если Не ЗначениеЗаполнено(CookieПользователя) Тогда
        Сообщить("Ошибка при получении Cookie");
        Возврат Ложь;
    КонецЕсли;
    //Создание соединения
    Попытка
        Соединение = Новый
        HTTPСоединение("develop.domuchet.online", 443,,,,,Новый
        ЗащищенноеСоединениеOpenSSL);
    Исключение
        Сообщить(ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Устанавливаем заголовки
    ЗаголовокHTTP = Новый Соответствие();
    ЗаголовокHTTP.Вставить("Cookie", CookieПользователя);
    //Вход пользователя ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();

    Запрос = Новый
    HTTPЗапрос("/api/auth?command=phone_confirm&code="
    КодАвторизации, ЗаголовокHTTP);
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
    ФайлДляОтвета);
    Если ОтветНаЗапрос.КодСостояния = 200 Тогда
        ТекстОтвета = Новый ЧтениеJSON;
        ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
        Ответ = ПрочитатьJSON(ТекстОтвета);
        ТекстОтвета.Закрыть();
        Если Ответ.status.success Тогда
            //При успешной авторизации записываем в параметр
сеанса "CookieПользователя" Cookie для выполнения дальнейших
запросов
            ВзаимодействиеСПараметрамиСеанса.УстановитьCookieПользовател
я(ОтветНаЗапрос.Заголовки["Set-Cookie"]);
        Иначе
            Сообщить(Ответ.status.message);

```

```

        Соединение = Неопределено;
        Возврат Ложь;
    КонецЕсли;
Иначе
    Сообщить ("Возникла ошибка при входе пользователя по
номеру телефона. Код состояния: " + ОтветНаЗапрос.КодСостояния);
    Соединение = Неопределено;
    Возврат Ложь;
КонецЕсли;
//Удаляем временный файл, чтобы не засорять память
ВременныйФайл = Новый Файл(ФайлДляОтвета);
Если ВременныйФайл.Существует() Тогда
    Попытка
        УдалитьФайлы(ФайлДляОтвета);
    Исключение
        Сообщить (ОписаниеОшибки());
    КонецПопытки;
КонецЕсли;
//Вход пользователя --
//Удаление соединения
Соединение = Неопределено;
Возврат Ответ;
КонецФункции

//Запрос на подтверждение номера телефона
Функция ПолучитьКодАвторизации(НомерТелефона) Экспорт
    //Создание соединения
    Попытка
        Соединение = Новый
        HTTPСоединение("develop.domuchet.online", 443,,,,Новый
        ЗащищенноеСоединениеOpenSSL);
    Исключение
        Сообщить (ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Вход пользователя ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();
    Запрос = Новый
    HTTPЗапрос("/api/auth?command=phone&phone="
    +
    СтрЗаменить(СтрЗаменить(СтрЗаменить(СтрЗаменить(СокрЛП(НомерТеле
фона), "+", ""), "(", ""), "-"), ""));
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
ФайлДляОтвета);
    Если ОтветНаЗапрос.КодСостояния = 200 Тогда
        ТекстОтвета = Новый ЧтениеJSON;
        ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
        Ответ = ПрочитатьJSON(ТекстОтвета);
        ТекстОтвета.Закреть();
        Если Ответ.status.success Тогда
            //При успешной авторизации записываем в параметр
сеанса "CookieПользователя" Cookie для выполнения дальнейших
запросов
            ВзаимодействиеСПараметрамиСеанса.УстановитьCookieПользовател
я(ОтветНаЗапрос.Заголовки["Set-Cookie"]);

```

```

        Иначе
            Сообщить (Ответ.status.message);
            Соединение = Неопределено;
            Возврат Ложь;
        КонецЕсли;
    Иначе
        Сообщить ("Возникла ошибка при получении кода
авторизации. Код состояния: " + ОтветНаЗапрос.КодСостояния);
        Соединение = Неопределено;
        Возврат Ложь;
    КонецЕсли;
    //Удаляем временный файл, чтобы не засорять память
    ВременныйФайл = Новый Файл(ФайлДляОтвета);
    Если ВременныйФайл.Существует() Тогда
        Попытка
            УдалитьФайлы(ФайлДляОтвета);
        Исключение
            Сообщить (ОписаниеОшибки());
            Соединение = Неопределено;
            Возврат Ложь;
        КонецПопытки;
    КонецЕсли;
    //Вход пользователя --
    //Удаление соединения
    Соединение = Неопределено;
    Возврат Ответ;
КонецФункции

//Выход пользователя
Функция ВыходПользователя() Экспорт
    //Получаем Cookie
    CookieПользователя =
ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
    Если Не ЗначениеЗаполнено(CookieПользователя) Тогда
        Сообщить ("Ошибка при получении Cookie");
        Возврат Ложь;
    КонецЕсли;
    //Создание соединения
    Попытка
        Соединение =
        HTTPСоединение("develop.domuchet.online",
        443,,,,,Новый
        ЗащищенноеСоединениеOpenSSL);
    Исключение
        Сообщить (ОписаниеОшибки());
    КонецПопытки;
    //Устанавливаем заголовки
    ЗаголовокHTTP = Новый Соответствие();
    ЗаголовокHTTP.Вставить("Cookie", CookieПользователя);
    //Выход пользователя ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();
    Запрос = Новый HTTPЗапрос("/api/auth?command=logout",
    ЗаголовокHTTP);
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
    ФайлДляОтвета);

```

```

Если ОтветНаЗапрос.КодСостояния = 200 Тогда
    ТекстОтвета = Новый ЧтениеJSON;
    ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
    Ответ = ПрочитатьJSON(ТекстОтвета);
    ТекстОтвета.Закрыть();
Иначе
    Сообщить("Возникла ошибка при выходе пользователя. Код
состояния: " + ОтветНаЗапрос.КодСостояния);
    Соединение = Неопределено;
    Возврат Ложь;
КонецЕсли;
//Удаляем временный файл, чтобы не засорять память
ВременныйФайл = Новый Файл(ФайлДляОтвета);
Если ВременныйФайл.Существует() Тогда
    Попытка
        УдалитьФайлы(ФайлДляОтвета);
    Исключение
        Сообщить(ОписаниеОшибки());
    КонецПопытки;
КонецЕсли;
//Выход пользователя --
//Удаляем соединение
Соединение = Неопределено;
Возврат Истина;
КонецФункции // ВыходПользователя()

// Проверка авторизации
Функция ПроверкаАвторизации() Экспорт
    //Получаем Cookie
    CookieПользователя =
ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
    Если Не ЗначениеЗаполнено(CookieПользователя) Тогда
        Сообщить("Ошибка при получении Cookie");
        Возврат Ложь;
    КонецЕсли;
    //Создание соединения
    Попытка
        Соединение = Новый
HTTPСоединение("develop.domuchet.online", 443,,,,,Новый
ЗащищенноеСоединениеOpenSSL);
    Исключение
        Сообщить(ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Устанавливаем заголовки
    ЗаголовокHTTP = Новый Соответствие();
    ЗаголовокHTTP.Вставить("Cookie", CookieПользователя);
    //Проверка авторизации ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();
    Запрос = Новый
HTTPЗапрос("/api/auth?command=check", ЗаголовокHTTP);
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
ФайлДляОтвета);
    Если ОтветНаЗапрос.КодСостояния = 200 Тогда

```

```

ТекстОтвета = Новый ЧтениеJSON;
ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
Ответ = ПрочитатьJSON(ТекстОтвета);
ТекстОтвета.Закрыть();
Если Не Ответ.status.success ИЛИ Ответ.id = Неопределено
Тогда
    Сообщить("Ошибка при проверке авторизации
пользователя. Попробуйте авторизоваться ещё раз. При повторном
возникновении ошибки обратитесь в службу поддержки");
    Соединение = Неопределено;
    Возврат Ложь;
КонецЕсли;
Иначе
    Сообщить("Возникла ошибка при проверке авторизации. Код
состояния: " + ОтветНаЗапрос.КодСостояния);
    Соединение = Неопределено;
    Возврат Ложь;
КонецЕсли;
//Удаляем временный файл, чтобы не засорять память
ВременныйФайл = Новый Файл(ФайлДляОтвета);
Если ВременныйФайл.Существует() Тогда
    Попытка
        УдалитьФайлы(ФайлДляОтвета);
    Исключение
        Сообщить(ОписаниеОшибки());
    КонецПопытки;
КонецЕсли;
//Проверка авторизации --
Соединение = Неопределено;
Возврат Истина;
КонецФункции

Функция СменаПароля(Пароль) Экспорт
    //Получаем Cookie
    CookieПользователя =
ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
    Если Не ЗначениеЗаполнено(CookieПользователя) Тогда
        Сообщить("Ошибка при получении Cookie");
        Возврат Ложь;
    КонецЕсли;
    //Создание соединения
    Попытка
        Соединение =
        HTTPСоединение("develop.domuchet.online", 443,,,,,Новый
        ЗащищенноеСоединениеOpenSSL);
    Исключение
        Сообщить(ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Устанавливаем заголовки
    ЗаголовокHTTP = Новый Соответствие();
    ЗаголовокHTTP.Вставить("Cookie", CookieПользователя);
    //Смена пароля ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();

```

```

        Запрос = Новый
        HTTPЗапрос("/api/auth?command=change_password&password="
        Пароль, ЗаголовокHTTP);
        ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
        ФайлДляОтвета);
        Если ОтветНаЗапрос.КодСостояния = 200 Тогда
            ТекстОтвета = Новый ЧтениеJSON;
            ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
            Ответ = ПрочитатьJSON(ТекстОтвета);
            ТекстОтвета.Заккрыть();
            Если Не Ответ.status.success Тогда
                Сообщить("Ошибка при смене пароля. Попробуйте
                сменить пароль ещё раз. При повторном возникновении ошибки
                обратитесь в службу поддержки");
                Соединение = Неопределено;
                Возврат Ложь;
            Иначе
                ВзаимодействиеСПараметрамиСеанса.УстановитьCookieПользовател
                я(ОтветНаЗапрос.Заголовки["Set-Cookie"]);
                КонецЕсли;
            Иначе
                Сообщить("Возникла ошибка при смене пароля. Код
                состояния: " + ОтветНаЗапрос.КодСостояния);
                Соединение = Неопределено;
                Возврат Ложь;
            КонецЕсли;
            //Удаляем временный файл, чтобы не засорять память
            ВременныйФайл = Новый Файл(ФайлДляОтвета);
            Если ВременныйФайл.Существует() Тогда
                Попытка
                    УдалитьФайлы(ФайлДляОтвета);
                Искключение
                    Сообщить(ОписаниеОшибки());
                КонецПопытки;
            КонецЕсли;
            //Смена пароля --
            Соединение = Неопределено;
            Возврат Истина;
        КонецФункции

#КонецОбласти

#Область МобильноеПриложениеЖильца
//Обзорная информация для жильца
Функция ПолучитьОбщуюИнформацию() Экспорт
    //Создание соединения
    Попытка
        Соединение =
        HTTPСоединение("develop.domuchet.online",
        ЗащищенноеСоединениеOpenSSL);
    Искключение
        Сообщить(ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;

```

```

        //Получаем Cookie
        CookieПользователя =
        ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
        Если Не ЗначениеЗаполнено(CookieПользователя) Тогда
            Сообщить("Ошибка при получении Cookie");
            Возврат Ложь;
        КонецЕсли;
        //Устанавливаем заголовки
        ЗаголовокHTTP = Новый Соответствие();
        ЗаголовокHTTP.Вставить("Cookie", CookieПользователя);
        //Получение общей информации о пользователе ++
        ФайлДляОтвета = ПолучитьИмяВременногоФайла();
        Запрос = Новый
        HTTPЗапрос("/api/roover/information", ЗаголовокHTTP);
        ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
        ФайлДляОтвета);
        Если ОтветНаЗапрос.КодСостояния = 200 Тогда
            ТекстОтвета = Новый ЧтениеJSON;
            ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
            Ответ = ПрочитатьJSON(ТекстОтвета);
            ТекстОтвета.Закрыть();
            Если Ответ.status.success Тогда
                //При успешном получении данных идём дальше
            Иначе
                Сообщить(Ответ.status.message);
                Соединение = Неопределено;
                Возврат Ложь;
            КонецЕсли;
        Иначе
            Сообщить("Возникла ошибка при получении общей
информации. Код состояния: " + ОтветНаЗапрос.КодСостояния);
            Соединение = Неопределено;
            Возврат Ложь;
        КонецЕсли;
        //Удаляем временный файл, чтобы не засорять память
        ВременныйФайл = Новый Файл(ФайлДляОтвета);
        Если ВременныйФайл.Существует() Тогда
            Попытка
                УдалитьФайлы(ФайлДляОтвета);
            Исключение
                Сообщить(ОписаниеОшибки());
            КонецПопытки;
        КонецЕсли;
        //Получение общей информации о пользователе --
        //Удаление соединения
        Соединение = Неопределено;
        Возврат Ответ;
    КонецФункции

//Получение показаний счетчика
Функция ПолучитьПоказанияСчетчика(IDСчетчика) Экспорт
    //Создание соединения
    Попытка

```

```

        Соединение = Новый
        HTTPСоединение ("develop.domuchet.online", 443,,,,,Новый
        ЗащищенноеСоединениеOpenSSL);
        Искключение
            Сообщить (ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Получаем Cookie
    CookieПользователя =
    ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
    Если Не ЗначениеЗаполнено(CookieПользователя) Тогда
        Сообщить("Ошибка при получении Cookie");
        Возврат Ложь;
    КонецЕсли;
    //Устанавливаем заголовки
    ЗаголовокHTTP = Новый Соответствие();
    ЗаголовокHTTP.Вставить("Cookie", CookieПользователя);
    //Получение показаний счетчика ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();
    Запрос = Новый
    HTTPЗапрос("/api/roomer/counters?command=get&counter_id="
    +
    Формат(IDСчетчика, "ЧГ=0"), ЗаголовокHTTP);
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
    ФайлДляОтвета);
    Если ОтветНаЗапрос.КодСостояния = 200 Тогда
        ТекстОтвета = Новый ЧтениеJSON;
        ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
        Ответ = ПрочитатьJSON(ТекстОтвета);
        ТекстОтвета.Закрыть();
        Если Ответ.status.success Тогда
            //При успешном получении данных идём дальше
        Иначе
            Сообщить (Ответ.status.message);
            Соединение = Неопределено;
            Возврат Ложь;
        КонецЕсли;
    Иначе
        Сообщить ("Возникла ошибка при получении показаний
        счетчика. Код состояния: " + ОтветНаЗапрос.КодСостояния);
        Соединение = Неопределено;
        Возврат Ложь;
    КонецЕсли;
    //Удаляем временный файл, чтобы не засорять память
    ВременныйФайл = Новый Файл(ФайлДляОтвета);
    Если ВременныйФайл.Существует() Тогда
        Попытка
            УдалитьФайлы(ФайлДляОтвета);
        Искключение
            Сообщить (ОписаниеОшибки());
        КонецПопытки;
    КонецЕсли;
    //Получение показаний счетчика --
    //Удаление соединения
    Соединение = Неопределено;

```

```

    Возврат Ответ;
КонецФункции

//Установка показаний счетчика
Функция УстановитьПоказанияСчетчика (IDСчетчика, Показание,
Переполнение = 0) Экспорт
    //Создание соединения
    Попытка
        Соединение = Новый
        HTTPСоединение ("develop.domuchet.online", 443,,,,,Новый
ЗащищенноеСоединениеOpenSSL);
    Исклучение
        Сообщить (ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Получаем Cookie
    CookieПользователя =
    ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
    Если Не ЗначениеЗаполнено (CookieПользователя) Тогда
        Сообщить ("Ошибка при получении Cookie");
        Возврат Ложь;
    КонецЕсли;
    //Устанавливаем заголовки
    ЗаголовокHTTP = Новый Соответствие();
    ЗаголовокHTTP.Вставить ("Cookie", CookieПользователя);
    //Установка показаний счетчика ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();
    Запрос = Новый
    HTTPЗапрос ("/api/roomer/counters?command=set?counter_id=" +
IDСчетчика + "&value=" + Показание + "&overflow=" + Переполнение,
ЗаголовокHTTP);
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод ("POST", Запрос,
ФайлДляОтвета);
    Если ОтветНаЗапрос.КодСостояния = 200 Тогда
        ТекстОтвета = Новый ЧтениеJSON;
        ТекстОтвета.ОткрытьФайл (ФайлДляОтвета);
        Ответ = ПрочитатьJSON (ТекстОтвета);
        ТекстОтвета.Закрыть();
        Если Ответ.status.success Тогда
            //При успешном получении данных идём дальше
        Иначе
            Сообщить (Ответ.status.message);
            Соединение = Неопределено;
            Возврат Ложь;
        КонецЕсли;
    Иначе
        Сообщить ("Возникла ошибка при установке показаний
счетчика. Код состояния: " + ОтветНаЗапрос.КодСостояния);
        Соединение = Неопределено;
        Возврат Ложь;
    КонецЕсли;
    //Удаляем временный файл, чтобы не засорять память
    ВременныйФайл = Новый Файл (ФайлДляОтвета);
    Если ВременныйФайл.Существует() Тогда

```

```

        Попытка
            УдалитьФайлы(ФайлДляОтвета);
        Исключение
            Сообщить(ОписаниеОшибки());
        КонецПопытки;
    КонецЕсли;
    //Установка показаний счетчика --
    //Удаление соединения
    Соединение = Неопределено;
    Возврат Истина;
КонецФункции

//Получение потребления счетчика за период
Функция ПолучитьПотреблениеСчетчика(ИДСчетчика, Период =
Неопределено) Экспорт
    //Создание соединения
    Попытка
        Соединение = Новый
        HTTPСоединение("develop.domuchet.online", 443,,,, Новый
        ЗащищенноеСоединениеOpenSSL);
    Исключение
        Сообщить(ОписаниеОшибки());
        Возврат Ложь;
    КонецПопытки;
    //Получаем Cookie
    CookieПользователя =
    ВзаимодействиеСПараметрамиСеанса.ПолучитьCookieПользователя();
    Если Не ЗначениеЗаполнено(CookieПользователя) Тогда
        Сообщить("Ошибка при получении Cookie");
        Возврат Ложь;
    КонецЕсли;
    //Устанавливаем заголовки
    ЗаголовокHTTP = Новый Соответствие();
    ЗаголовокHTTP.Вставить("Cookie", CookieПользователя);
    //Получение показаний счетчика ++
    ФайлДляОтвета = ПолучитьИмяВременногоФайла();
    Запрос = Новый
    HTTPЗапрос("/api/roomer/counters?command=consumption&counter_id=
" + Формат(ИДСчетчика, "ЧГ=0") + "(Период = Неопределено, "",
Формат(Период, "ДФ=yyyy-MM-dd")), ЗаголовокHTTP);
    ОтветНаЗапрос = Соединение.ВызватьHTTPМетод("POST", Запрос,
    ФайлДляОтвета);
    Если ОтветНаЗапрос.КодСостояния = 200 Тогда
        ТекстОтвета = Новый ЧтениеJSON;
        ТекстОтвета.ОткрытьФайл(ФайлДляОтвета);
        Ответ = ПрочитатьJSON(ТекстОтвета);
        ТекстОтвета.Закреть();
        Если Ответ.status.success Тогда
            //При успешном получении данных идём дальше
        Иначе
            Сообщить(Ответ.status.message);
            Соединение = Неопределено;
            Возврат Ложь;
        КонецЕсли;

```

```

        Иначе
            Сообщить ("Возникла ошибка при получении потребления
счетчика. Код состояния: " + ОтветНаЗапрос.КодСостояния);
            Соединение = Неопределено;
            Возврат Ложь;
        КонецЕсли;
        //Удаляем временный файл, чтобы не засорять память
        ВременныйФайл = Новый Файл(ФайлДляОтвета);
        Если ВременныйФайл.Существует() Тогда
            Попытка
                УдалитьФайлы(ФайлДляОтвета);
            Исключение
                Сообщить (ОписаниеОшибки());
            КонецПопытки;
        КонецЕсли;
        //Получение показаний счетчика --

        //Удаление соединения
        Соединение = Неопределено;
        Возврат Ответ;
    КонецФункции
#КонецОбласти
#КонецОбласти

#Область РаботаСДаннымиПользователя
Функция ПолучитьЛогинПользователя() Экспорт
    Возврат Константы.ЛогинПользователя.Получить();
КонецФункции

Функция ПолучитьПарольПользователя() Экспорт
    Возврат Константы.ПарольПользователя.Получить();
КонецФункции

Процедура СохранитьЛогинПользователя(Логин) Экспорт
    Константы.ЛогинПользователя.Установить(Логин);
КонецПроцедуры

Процедура СохранитьПарольПользователя(Пароль) Экспорт
    Константы.ПарольПользователя.Установить(Пароль);
КонецПроцедуры
#КонецОбласти

```

## Листинг Б.2 – Код общего модуля «ВзаимодействиеСПараметрамиСеанса»

```

#Область РаботаСCookie
Процедура УстановитьCookieПользователя(CookieПользователя)
Экспорт
    ПараметрыСеанса.CookieПользователя = CookieПользователя;
КонецПроцедуры // УстановитьCookieПользователя()

Функция ПолучитьCookieПользователя() Экспорт
    Возврат ПараметрыСеанса.CookieПользователя;
КонецФункции // ПолучитьCookieПользователя()

```

```
#КонецОбласти
```

```
#Область РаботаСПараметрамиСеанса
Процедура ОчисткаПараметровСеанса () Экспорт
    ПараметрыСеанса.Очистить ();
КонецПроцедуры
#КонецОбласти
```

### Листинг Б.3 – Код общей формы «АвторизацияПользователя»

```
&НаСервере
Перем ЛогинПользователя;
&НаСервере
Перем ПарольПользователя;

#Область СтандартныеОбработки
&НаКлиенте
Процедура ПриОткрытии(Отказ)
    ВосстановитьНастройкиАвторизации ();
    ЗапомнитьУчетныеДанные = Истина;
КонецПроцедуры
#КонецОбласти

#Область КомандыФормы
&НаКлиенте
Процедура Войти(Команда)
    Если Не ПроверитьЗаполнение () Тогда
        Возврат;
    КонецЕсли;
    Если АРІПроцедурыИФункции.ВходПользователя(Логин, Пароль)
Тогда
        Если АРІПроцедурыИФункции.ПроверкаАвторизации() Тогда
            СохранитьНастройкиАвторизации(Логин, Пароль);

            ЗаккрытиеРазрешено = Истина;
            Заккрыть ();
            ОткрытьФорму ("Обработка.ГлавнаяСтраница.Форма.Форма");
            КонецЕсли;
        КонецЕсли;
КонецПроцедуры
#КонецОбласти

&НаСервере
Процедура ВосстановитьНастройкиАвторизации ()
    ЛогинПользователя = Константы.ЛогинПользователя.Получить ();
    ПарольПользователя =
Константы.ПарольПользователя.Получить ();
    Если ЗначениеЗаполнено(ЛогинПользователя) Тогда
        Логин = ЛогинПользователя;
    КонецЕсли;
    Если ЗначениеЗаполнено(ПарольПользователя) Тогда
        Пароль = ПарольПользователя;
    КонецЕсли;
```

```

КонецПроцедуры //Получаем логин и пароль пользователя

&НаСервере
Процедура СохранитьНастройкиАвторизации(Логин, Пароль)
    Если ЗапомнитьУчетныеДанные Тогда
        Если Логин <> ЛогинПользователя ИЛИ Пароль <>
ПарольПользователя Тогда
            АРІПроцедурыИФункции.СохранитьЛогинПользователя (СокрЛП (Логин
)) ;
            АРІПроцедурыИФункции.СохранитьПарольПользователя (СокрЛП (Паро
ль)) ;
        КонецЕсли;
    Иначе
        АРІПроцедурыИФункции.СохранитьЛогинПользователя ("") ;
        АРІПроцедурыИФункции.СохранитьПарольПользователя ("") ;
    КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура Регистрация(Команда)
    ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователяПоТелефону")
;
    Заккрыть () ;
КонецПроцедуры

&НаКлиенте
Процедура ЗабылПароль(Команда)
    ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователяПоТелефону")
;
    Заккрыть () ;
КонецПроцедуры

```

#### Листинг Б.4 – Код общей формы «АвторизацияПользователяПоТелефону»

```

&НаКлиенте
Перем ВернутьсяНазад;

#Область СтандартныеОбработки
&НаКлиенте
Процедура ПриЗакрытии(ЗавершениеРаботы)
    Если Не ЗавершениеРаботы И ВернутьсяНазад Тогда
        ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователя") ;
    КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии(Отказ)
    ВернутьсяНазад = Истина;
КонецПроцедуры
#КонецОбласти

#Область КомандыФормы
&НаКлиенте

```

```

Процедура ПолучитьКод(Команда)
    ДатаОтправкиКода = ПолучитьДатуОтправкиКода();
    Если ТекущаяДата() < ДатаОтправкиКода Тогда
        Сообщить("Новый код Вы сможете получить через " +
        Формат((ДатаОтправкиКода - ТекущаяДата()), "ЧДЦ=0") + " секунд");
        Возврат;
    КонецЕсли;
    Если Не ЗначениеЗаполнено(НомерТелефона) Тогда
        Сообщить("Введите номер телефона");
        Возврат;
    КонецЕсли;
    Ответ =
    АРІПроцедурыИФункции.ПолучитьКодАвторизации(НомерТелефона);
    Если Ответ = Ложь Тогда
        Возврат;
    КонецЕсли;
    УстановитьДатуОкончанияДействияКода(Ответ.timeout);
    УстановитьДатуОтправкиКода(Ответ.interval);
    ОткрытьФорму("ОбщаяФорма.АвторизацияПользователяВводКода");
    ВернутьсяНазад = Ложь;
    Заккрыть();
КонецПроцедуры

```

&НаКлиенте

Процедура Назад(Команда)

```

    ОткрытьФорму("ОбщаяФорма.ФормаВыбораАвторизации");
    ВернутьсяНазад = Истина;
    Заккрыть();

```

КонецПроцедуры

#КонецОбласти

&НаСервере

Процедура УстановитьДатуОтправкиКода(interval)

```

    Константы.ДатаОтправкиКодаАвторизации.Установить(ТекущаяДата
    () + interval);
КонецПроцедуры

```

&НаСервере

Функция ПолучитьДатуОтправкиКода()

```

    Возврат Константы.ДатаОтправкиКодаАвторизации.Получить();

```

КонецФункции

&НаСервере

Процедура УстановитьДатуОкончанияДействияКода(timeout)

```

    Константы.ДатаОкончанияДействияКода.Установить(ТекущаяДата()
    + timeout);
КонецПроцедуры

```

## Листинг Б.5 – Код общей формы «ФормаВыбораАвторизации»

&НаКлиенте

Перем ЗаккрытиеРазрешено;

&НаКлиенте

Перем СменаФормы;

#Область КомандыФормы

&НаКлиенте

Процедура Справка(Команда)

Сообщить("Если у вас есть логин (email/номер телефона) и пароль, то используйте авторизацию по логину и паролю. " +

"Если у вас нет логина и пароля или пароль был утерян/забыт, то используйте регистрацию по номеру телефона");

КонецПроцедуры

&НаКлиенте

Процедура АвторизацияПоЛогинуИПаролю(Команда)

СменаФормы = Истина;

ОткрытьФорму("ОбщаяФорма.АвторизацияПользователя");

Закрыть();

КонецПроцедуры

&НаКлиенте

Процедура РегистрацияПоНомеруТелефона(Команда)

СменаФормы = Истина;

ОткрытьФорму("ОбщаяФорма.АвторизацияПользователяПоТелефону")

;

Закрыть();

КонецПроцедуры

&НаКлиенте

Процедура ВыйтиИзПриложения(Команда)

ЗавершитьРаботуСистемы();

КонецПроцедуры

#КонецОбласти

#Область СтандартныеОбработки

&НаКлиенте

Процедура ПередЗакрытием(Отказ, ЗавершениеРаботы, ТекстПредупреждения, СтандартнаяОбработка)

Если Не ЗавершениеРаботы Тогда

Если ЗакрытиеРазрешено = Неопределено И СменаФормы = Ложь

Тогда

Отказ = Истина;

ПоказатьВопрос

(Новый

ОписаниеОповещения("ПередЗакрытиемЗавершение", ЭтотОбъект), "Выйти из приложения?", РежимДиалогаВопрос.ДаНет);

КонецЕсли;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

СменаФормы = Ложь;

КонецПроцедуры

#КонецОбласти

&НаКлиенте

```

Процедура          ПередЗакрытиемЗавершение (РезультатВопроса,
ДополнительныеПараметры) Экспорт
    Если РезультатВопроса = КодВозвратаДиалога.Да Тогда
        ЗакрытиеРазрешено = Истина;
        ЗавершитьРаботуСистемы();
    КонецЕсли;
КонецПроцедуры

```

## Листинг Б.6 – Код общей формы «ФормаСменыПароля»

```

&НаКлиенте
Перем ВернутьсяНазад;

#Область СтандартныеОбработки
&НаКлиенте
Процедура          ПередЗакрытием(Отказ,          ЗавершениеРаботы,
ТекстПредупреждения, СтандартнаяОбработка)
    Если Не ЗавершениеРаботы И ВернутьсяНазад Тогда
        ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователя");
    КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии(Отказ)
    ВернутьсяНазад = Истина;
КонецПроцедуры
#КонецОбласти

#Область КомандыФормы
&НаКлиенте
Процедура Подтвердить(Команда)
    Если      Не      ЗначениеЗаполнено(Пароль)      Или      Не
ЗначениеЗаполнено(ПодтверждениеПароля) Тогда
        Сообщить ("Заполните поля ""Пароль"" и ""Подтверждение
пароля""");
        Возврат;
    КонецЕсли;
    Если СтрДлина(Пароль) < 8 Или СтрДлина(ПодтверждениеПароля)
< 8 Тогда
        Сообщить ("Длина пароля должна быть не меньше 8
символов");
        Возврат;
    КонецЕсли;
    Если Пароль <> ПодтверждениеПароля Тогда
        Сообщить ("Введённые пароли не совпадают");
        Возврат;
    КонецЕсли;
    Если АРІПроцедурыИФункции.СменаПароля(Пароль) Тогда
        ВернутьсяНазад = Ложь;
        Закрыть();
        ОткрытьФорму ("Обработка.ГлавнаяСтраница.Форма.Форма");
    КонецЕсли;
КонецПроцедуры

```

```
#КонецОбласти
```

## Листинг Б.7 – Код общей формы «АвторизацияПользователяВводКода»

```
&НаКлиенте
Перем ВернутьсяНазад;

#Область СтандартныеОбработки
&НаКлиенте
Процедура ПриЗакрытии(ЗавершениеРаботы)
    Если Не ЗавершениеРаботы И ВернутьсяНазад Тогда
        ОткрытьФорму("ОбщаяФорма.АвторизацияПользователяПоТелефону")
;
    КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии(Отказ)
    ВернутьсяНазад = Истина;
КонецПроцедуры
#КонецОбласти

#Область КомандыФормы
&НаКлиенте
Процедура Авторизоваться(Команда)

    ДатаОкончанияДействияКода =
ПолучитьДатуОкончанияДействияКода();
    Если ТекущаяДата() > ДатаОкончанияДействияКода Тогда
        Сообщить("Время действия кода истекло. Запросите новый
код, вернувшись на предыдущую страницу");
        Возврат;
    КонецЕсли;
    Ответ =
APIПроцедурыИФункции.ВходПользователяПоНомеруТелефона(СокрЛП(Код
Авторизации));
    Если Ответ = Ложь Тогда
        Возврат;
    ИначеЕсли Ответ.role = "support" Тогда
        Сообщить("Вы ввели номер телефона сотрудника
тех.поддержки. Приложение предназначено только для жильцов");
        Возврат;
    КонецЕсли;
    Если APIПроцедурыИФункции.ПроверкаАвторизации() Тогда
        Логин = Неопределено;
        Пароль = Неопределено;
        Если Ответ.Свойство("phone", Логин) И
Ответ.Свойство("quick_password", Пароль) Тогда
            APIПроцедурыИФункции.СохранитьЛогинПользователя(Логин);
            APIПроцедурыИФункции.СохранитьПарольПользователя(Пароль);
        Иначе
            APIПроцедурыИФункции.СохранитьЛогинПользователя("");
```

```

APIПроцедурыИФункции.СохранитьПарольПользователя("");

КонецЕсли;
ВернутьсяНазад = Ложь;
Заккрыть();
ОткрытьФорму("ОбщаяФорма.ФормаСменыПароля");
Иначе
    Возврат;
КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура Назад(Команда)
    ОткрытьФорму("ОбщаяФорма.АвторизацияПользователяПоТелефону")
;
    ВернутьсяНазад = Истина;
    Заккрыть();
КонецПроцедуры
#КонецОбласти

&НаСервере
Функция ПолучитьДатуОкончанияДействияКода()
    Возврат Константы.ДатаОкончанияДействияКода.Получить();
КонецФункции

```

## Листинг Б.8 – Код формы «Форма» обработки «ГлавнаяСтраница»

```

&НаКлиенте
Перем ЗаккрытиеРазрешено;
&НаКлиенте
Перем СменаФормы;

#Область СтандартныеОбработки
&НаКлиенте
Процедура ПриОткрытии(Отказ)
    СменаФормы = Ложь;
    //Получаем данные о пользователе
    Ответ = APIПроцедурыИФункции.ПолучитьОбщуюИнформацию();
    Если Ответ = Ложь Тогда
        Сообщить("Ошибка при получении данных. Попробуйте авторизоваться снова");
        Отказ = Истина;
        ОткрытьФорму("ОбщаяФорма.АвторизацияПользователя");
        Возврат;
    КонецЕсли;
    //Проверка на роль пользователя
    СведенияОПользователе = Неопределено;
    Роль = Неопределено;
    Если Не Ответ.Свойство("user", СведенияОПользователе) Тогда
        Сообщить("Ошибка при получении данных. Попробуйте авторизоваться снова");
        Отказ = Истина;
        СменаФормы = Истина;
    КонецЕсли;
КонецПроцедуры

```

```

        ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователя");
        Возврат;
    Иначе
        Если Не СведенияОПользователе.Свойство("role", Роль)
Тогда
        Сообщить ("Ошибка при получении данных. Попробуйте
авторизоваться снова");
        Отказ = Истина;
        СменаФормы = Истина;

        ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователя");
        Возврат;
    Иначе
        Если Роль <> "roomer" Тогда
            Сообщить ("Приложение предназначено только для
жильцов");

            Отказ = Истина;
            СменаФормы = Истина;

        ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователя");
        Возврат;
        КонецЕсли;
    КонецЕсли;
КонецЕсли;
Если Ответ.accounts.Количество() > 0 Тогда
    //Основные сведения
    СтруктураАккаунта = Неопределено;
    СтруктураЗдания = Неопределено;
    СтруктураОрганизации = Неопределено;
    СтруктураПомещения = Неопределено;
    МассивСчетчиков = Неопределено;
    Аккаунт = Ответ.accounts[0];
    Аккаунт.Свойство("account", СтруктураАккаунта);
    Аккаунт.Свойство("building", СтруктураЗдания);
    Аккаунт.Свойство("organization", СтруктураОрганизации);
    Аккаунт.Свойство("room", СтруктураПомещения);
    Аккаунт.Свойство("counters", МассивСчетчиков);
    СведенияОПользователе.Свойство("id",
Объект.IdПользователя);
    Если СтруктураАккаунта = Неопределено Или
СтруктураЗдания = Неопределено Или СтруктураОрганизации =
Неопределено Или
        СтруктураПомещения = Неопределено Или
МассивСчетчиков = Неопределено Тогда

        Сообщить ("Ошибка получения данных. Обратитесь к
разработчикам");
        Отказ = Истина;
        СменаФормы = Истина;

        ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователя");
        Возврат;
    КонецЕсли;
    //Заполняем сведения о пользователе

```

```

        СведенияОПользователе.Свойство("name",
Объект.Пользователь);
        СтруктураАккаунта.Свойство("number",    Объект.НомерЛС);
        ЗаполнитьДанныеОПользователе(СтруктураАккаунта,
Объект.IdПользователя);
        //Заполняем сведения о здании и помещении
        Адрес = Неопределено;
        НомерКвартиры = Неопределено;
        СтруктураЗдания.Свойство("address",    Адрес);
        СтруктураПомещения.Свойство("number",    НомерКвартиры);
        Объект.Адрес = Адрес + ", кв. " + НомерКвартиры;
        ЗаполнитьДанныеОЗдании(СтруктураЗдания,
Объект.IdПользователя);
        ЗаполнитьДанныеОПомещении(СтруктураПомещения,
Объект.IdПользователя);
        //Заполняем сведения об оплатах
        Если Аккаунт.settlements.Количество() > 0 Тогда
            МаксимальныйПериод = Дата("00010101");
            Для Каждого СтрокаНачислений Из Аккаунт.settlements
Цикл
                ПериодНачисления =
Дата(СтрЗаменить(СтрокаНачислений.period, "-", ""));
                Если ПериодНачисления > МаксимальныйПериод
Тогда
                    Объект.ТекущаяЗадолженность =
СтрокаНачислений.GIS_TotalPayableByPDWithDebtAndAdvance;
                    МаксимальныйПериод = ПериодНачисления;
                    КонецЕсли;

                    Если Объект.ТекущаяЗадолженность < 0 Тогда
                        Элементы.ТекущаяЗадолженность.Заголовок =
"Переплата";

                        КонецЕсли;
                    КонецЦикла;
                ЗаполнитьДанныеОНачислениях(Аккаунт.settlements,
Объект.IdПользователя);
                КонецЕсли;
            //Заполняем данные об организации
            ЗаполнитьДанныеОБорганизации(СтруктураОрганизации,
Объект.IdПользователя);
            //Заполняем данные о счетчиках
            ЗаполнитьДанныеОСчетчиках(МассивСчетчиков,
Объект.IdПользователя);
            Иначе
                Сообщить("Не найдены сведения по ЛС для указанного логина
и пароля. Попробуйте авторизоваться снова");
                Отказ = Истина;
                ОткрытьФорму("ОбщаяФорма.АвторизацияПользователя");
            КонецЕсли;
        КонецПроцедуры

&НаКлиенте
Процедура    ПередЗакрытием(Отказ,    ЗавершениеРаботы,
ТекстПредупреждения, СтандартнаяОбработка)

```

```

        Если Не ЗавершениеРаботы Тогда
            Если ЗакрытиеРазрешено = Неопределено И СменаФормы = Ложь
Тогда
                Отказ = Истина;
                ПоказатьВопрос (Новый
ОписаниеОповещения ("ПередЗакрытиемЗавершение", ЭтотОбъект),
"Выйти из приложения?", РежимДиалогаВопрос.ДаНет);
                КонецЕсли;
            КонецЕсли;
        КонецПроцедуры
#КонецОбласти

#Область ЗаполнениеРегистровСведений
&НаСервере
Процедура ЗаполнитьДанныеОПользователе (СтруктураАккаунта,
IdПользователя)
    НаборЗаписей =
РегистрыСведений.СведенияОПользователе.СоздатьНаборЗаписей();
    ЗаписьНабора = НаборЗаписей.Добавить();
    ЗаписьНабора.IdПользователя = IdПользователя;
    СтруктураАккаунта.Свойство("begin_date",
ЗаписьНабора.ДатаОткрытия);
    СтруктураАккаунта.Свойство("end_date",
ЗаписьНабора.ДатаЗакрытия);
    СтруктураАккаунта.Свойство("number",
ЗаписьНабора.НомерЛС);
    НаборЗаписей.Записать (Истина);
КонецПроцедуры

&НаСервере
Процедура ЗаполнитьДанныеОЗдании (СтруктураЗдания, IdПользователя)
    НаборЗаписей =
РегистрыСведений.СведенияОЗдании.СоздатьНаборЗаписей();
    ЗаписьНабора = НаборЗаписей.Добавить();
    ЗаписьНабора.IdПользователя = IdПользователя;
    СтруктураЗдания.Свойство("address",
ЗаписьНабора.Адрес);
    СтруктураЗдания.Свойство("area_living",
ЗаписьНабора.ЖилаяПлощадь);
    СтруктураЗдания.Свойство("area_nonliving",
ЗаписьНабора.НежилаяПлощадь);
    СтруктураЗдания.Свойство("year_of_construction",
ЗаписьНабора.ГодПостройки);
    НаборЗаписей.Записать (Истина);
КонецПроцедуры

&НаСервере
Процедура ЗаполнитьДанныеОбОрганизации (СтруктураОрганизации,
IdПользователя)
    НаборЗаписей =
РегистрыСведений.СведенияОбОрганизации.СоздатьНаборЗаписей();
    ЗаписьНабора = НаборЗаписей.Добавить();
    ЗаписьНабора.IdПользователя = IdПользователя;

```

```

        СтруктураОрганизации.Свойство("address",
ЗаписьНабора.Адрес);
        СтруктураОрганизации.Свойство("inn",      ЗаписьНабора.ИНН);
        СтруктураОрганизации.Свойство("kpp",      ЗаписьНабора.КПП);
        СтруктураОрганизации.Свойство("mail",
ЗаписьНабора.Email);
        СтруктураОрганизации.Свойство("name",
ЗаписьНабора.Наименование);
        СтруктураОрганизации.Свойство("ogrn",
ЗаписьНабора.ОГРН);
        СтруктураОрганизации.Свойство("okpo",
ЗаписьНабора.ОКПО);
        СтруктураОрганизации.Свойство("phone",
ЗаписьНабора.НомерТелефона);
        СтруктураОрганизации.Свойство("site",
ЗаписьНабора.Сайт);
        НаборЗаписей.Записать(Истина);
КонецПроцедуры

```

```

&НаСервере
Процедура      ЗаполнитьДанныеОПомещении(СтруктураПомещения,
IdПользователя)
    НаборЗаписей
РегистрыСведений.СведенияОПомещении.СоздатьНаборЗаписей();
    ЗаписьНабора = НаборЗаписей.Добавить();
    ЗаписьНабора.IdПользователя = IdПользователя;
    СтруктураПомещения.Свойство("area",
ЗаписьНабора.ПлощадьОбщая);
    СтруктураПомещения.Свойство("GIS_GrossArea",
ЗаписьНабора.ПлощадьЖилая);
    СтруктураПомещения.Свойство("floor",
ЗаписьНабора.Этаж);
    СтруктураПомещения.Свойство("entrance",
ЗаписьНабора.Подъезд);
    СтруктураПомещения.Свойство("isliving",
ЗаписьНабора.Проживающих);
    СтруктураПомещения.Свойство("rooms",
ЗаписьНабора.Комнат);
    НаборЗаписей.Записать(Истина);
КонецПроцедуры

```

```

&НаСервере
Процедура      ЗаполнитьДанныеОНачислениях(СведенияОНачислениях,
IdПользователя)
    НаборЗаписей
РегистрыСведений.СведенияОНачислениях.СоздатьНаборЗаписей();
    Для Каждого СтрокаНачисления Из СведенияОНачислениях Цикл
        ЗаписьНабора
НаборЗаписей.Добавить();
        ЗаписьНабора.IdПользователя = IdПользователя;
        ЗаписьНабора.ОстатокПредыдущийПериод
СтрокаНачисления.GIS_DebtPreviousPeriods;
        ЗаписьНабора.Начислено
СтрокаНачисления.GIS_TotalPayableByChargeInfo;

```

```

        ЗаписьНабора.Пени =
СтрокаНачисления.GIS_TotalByPenaltiesAndCourtCosts;
        ЗаписьНабора.Оплачено =
СтрокаНачисления.GIS_PaidCash;
        ЗаписьНабора.КОплате =
СтрокаНачисления.GIS_TotalPayableByPDWithDebtAndAdvance;
        ЗаписьНабора.ПериодОплаты =
Дата (СтрЗаменить (СтрокаНачисления.period, "-", ""));
        КонецЦикла;
        НаборЗаписей.Записать (Истина) ;
КонецПроцедуры

&НаСервере
Процедура                ЗаполнитьДанныеОСчетчиках (МассивСчетчиков,
IdПользователя)
        НаборЗаписей =
РегистрыСведений.СведенияОСчетчиках.СоздатьНаборЗаписей();
        Для Каждого СтрокаСчетчика Из МассивСчетчиков Цикл
                ЗаписьНабора =
НаборЗаписей.Добавить ();
                ЗаписьНабора.IdПользователя =
IdПользователя;
                ЗаписьНабора.IdСчетчика =
СтрокаСчетчика.id;
                ЗаписьНабора.ДатаВводаСчетчика =
Дата (СтрЗаменить (? (СтрокаСчетчика.begin_date = Неопределено,
"00010101", СтрокаСчетчика.begin_date), "-", ""));
                ЗаписьНабора.ДатаОпломбированияПУ =
Дата (СтрЗаменить (? (СтрокаСчетчика.GIS_FactorySealDate =
Неопределено, "00010101", СтрокаСчетчика.GIS_FactorySealDate), "-
", ""));
                ЗаписьНабора.ДатаПоследнейПоверки =
Дата (СтрЗаменить (? (СтрокаСчетчика.GIS_FirstVerificationDate =
Неопределено, "00010101",
СтрокаСчетчика.GIS_FirstVerificationDate), "-", ""));
                ЗаписьНабора.ДатаПрекращенияЭксплуатации =
Дата (СтрЗаменить (? (СтрокаСчетчика.end_date = Неопределено,
"00010101", СтрокаСчетчика.end_date), "-", ""));
                ЗаписьНабора.ИмяРесурса =
СтрокаСчетчика.resource_name;
                ЗаписьНабора.МаркаСчетчика =
СтрокаСчетчика.name;
                ЗаписьНабора.МежповерочныйИнтервал =
СтрокаСчетчика.GIS_VerificationInterval;
                ЗаписьНабора.Модель =
СтрокаСчетчика.model;
                ЗаписьНабора.Наименование =
СтрокаСчетчика.title;
                ЗаписьНабора.Период =
ТекущаяДата ();
                ЗаписьНабора.СерийныйНомер =
СтрокаСчетчика.serial;
                ЗаписьНабора.Статус =
СтрокаСчетчика.GIS_StatusRootDoc;

```

```

        КонецЦикла;
        НаборЗаписей.Записать (Истина) ;
КонецПроцедуры
#КонецОбласти

#Область КомандыФормы
&НаКлиенте
Процедура СменаПользователя (Команда)
    СменаФормы = Истина;
    APIПроцедурыИФункции.ВыходПользователя ();
    Закрыть ();
    ОткрытьФорму ("ОбщаяФорма.АвторизацияПользователя");
КонецПроцедуры

&НаКлиенте
Процедура Информация (Команда)
    ОткрытьФорму ("Обработка.ГлавнаяСтраница.Форма.ФормаИнформаци
и", Новый Структура ("IdПользователя", Объект.IdПользователя));
КонецПроцедуры

&НаКлиенте
Процедура Начисления (Команда)
    ОткрытьФорму ("Обработка.ГлавнаяСтраница.Форма.ФормаНачислени
я", Новый Структура ("IdПользователя", ТекущаяЗадолженность",
Объект.IdПользователя, Объект.ТекущаяЗадолженность));
КонецПроцедуры

&НаКлиенте
Процедура Счетчики (Команда)
    ОткрытьФорму ("Обработка.ГлавнаяСтраница.Форма.ФормаСчетчики"
, Новый Структура ("IdПользователя", Объект.IdПользователя));

КонецПроцедуры

&НаКлиенте
Процедура ВыйтиИзПриложения (Команда)
    ЗавершитьРаботуСистемы ();
КонецПроцедуры
#КонецОбласти

&НаКлиенте
Процедура ПередЗакрытиемЗавершение (РезультатВопроса,
ДополнительныеПараметры) Экспорт
    Если РезультатВопроса = КодВозвратаДиалога.Да Тогда
        ЗакрытиеРазрешено = Истина;
        ЗавершитьРаботуСистемы ();
    КонецЕсли;
КонецПроцедуры

```

Листинг Б.9 – Код формы «ФормаИнформации» обработки  
«ГлавнаяСтраница»

#Область СтандартныеОбработки

```

&НаСервере
Процедура ПриОткрытииНаСервере()
    //Заполнение сведений о пользователе
    ЗаполнитьСведенияОПользователе();
    //Заполнение сведений об организации
    ЗаполнитьСведенияОбОрганизации();
    //Заполнение сведений о здании
    ЗаполнитьСведенияОЗдании();
    //Заполнение сведений о помещении
    ЗаполнитьСведенияОПомещении();
КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии(Отказ)
    ПриОткрытииНаСервере();
КонецПроцедуры

&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    Параметры.Свойство("IdПользователя", Объект.IdПользователя);
КонецПроцедуры
#КонецОбласти

#Область ЗаполнениеСведений
&НаСервере
Процедура ЗаполнитьСведенияОПользователе()
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
         |      СведенияОПользователе.ДатаОткрытия           КАК
ДатаОткрытия,
         |      СведенияОПользователе.ДатаЗакрытия           КАК
ДатаЗакрытия,
         |      СведенияОПользователе.НомерЛС КАК НомерЛС
         | ИЗ
         |      РегистрСведений.СведенияОПользователе           КАК
СведенияОПользователе
         | ГДЕ
         |      СведенияОПользователе.IdПользователя           =
&IdПользователя";
    Запрос.УстановитьПараметр("IdПользователя",
Объект.IdПользователя);
    РезультатЗапроса = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
    Если ВыборкаДетальныеЗаписи.Следующий() Тогда
        ДатаОткрытияЛС = ВыборкаДетальныеЗаписи.ДатаОткрытия;
        ДатаЗакрытияЛС = ВыборкаДетальныеЗаписи.ДатаЗакрытия;
        НомерЛС        = ВыборкаДетальныеЗаписи.НомерЛС;
    КонецЕсли;
КонецПроцедуры

&НаСервере
Процедура ЗаполнитьСведенияОбОрганизации()
    Запрос = Новый Запрос;

```

```

Запрос.Текст =
    "ВЫБРАТЬ
      | СведенияОБОрганизации.Адрес КАК Адрес,
      | СведенияОБОрганизации.ИНН КАК ИНН,
      | СведенияОБОрганизации.КПП КАК КПП,
      | СведенияОБОрганизации.Email КАК Email,
      | СведенияОБОрганизации.Наименование КАК
Наименование,
      | СведенияОБОрганизации.ОГРН КАК ОГРН,
      | СведенияОБОрганизации.ОКПО КАК ОКПО,
      | СведенияОБОрганизации.НомерТелефона КАК
НомерТелефона,
      | СведенияОБОрганизации.Сайт КАК Сайт
      | ИЗ
      | РегистрСведений.СведенияОБОрганизации КАК
СведенияОБОрганизации
      | ГДЕ
      | СведенияОБОрганизации.IdПользователя =
&IdПользователя";
    Запрос.УстановитьПараметр ("IdПользователя",
Объект.IdПользователя);
    РезультатЗапроса = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
    Если ВыборкаДетальныеЗаписи.Следующий() Тогда
        АдресОрганизации = ВыборкаДетальныеЗаписи.Адрес;
        ИннОрганизации =
ВыборкаДетальныеЗаписи.ИНН;
        КппОрганизации =
ВыборкаДетальныеЗаписи.КПП;
        EmailОрганизации = ВыборкаДетальныеЗаписи.Email;
        НаименованиеОрганизации =
ВыборкаДетальныеЗаписи.Наименование;
        НомерТелефонаОрганизации =
ВыборкаДетальныеЗаписи.НомерТелефона;
        СайтОрганизации = ВыборкаДетальныеЗаписи.Сайт;
    КонецЕсли;
КонецПроцедуры

&НаСервере
Процедура ЗаполнитьСведенияОЗдании()
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
          | СведенияОЗдании.Адрес КАК Адрес,
          | СведенияОЗдании.ЖилаяПлощадь КАК ЖилаяПлощадь,
          | СведенияОЗдании.НежилаяПлощадь КАК НежилаяПлощадь,
          | СведенияОЗдании.ГодПостройки КАК ГодПостройки
          | ИЗ
          | РегистрСведений.СведенияОЗдании КАК СведенияОЗдании
          | ГДЕ
          | СведенияОЗдании.IdПользователя = &IdПользователя";
    Запрос.УстановитьПараметр ("IdПользователя",
Объект.IdПользователя);
    РезультатЗапроса = Запрос.Выполнить();

```

```

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать ();
Если ВыборкаДетальныеЗаписи.Следующий() Тогда
    Адрес = ВыборкаДетальныеЗаписи.Адрес;
    ЖилаяПлощадь = ВыборкаДетальныеЗаписи.ЖилаяПлощадь;
    НежилаяПлощадь = ВыборкаДетальныеЗаписи.НежилаяПлощадь;
    ГодПостройки = ВыборкаДетальныеЗаписи.ГодПостройки;
КонецЕсли;
КонецПроцедуры

&НаСервере
Процедура ЗаполнитьСведенияОПомещении()
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     СведенияОПомещении.ПлощадьОбщая КАК ПлощадьОбщая,
        |     СведенияОПомещении.ПлощадьЖилая КАК ПлощадьЖилая,
        |     СведенияОПомещении.Этаж КАК Этаж,
        |     СведенияОПомещении.Подъезд КАК Подъезд,
        |     СведенияОПомещении.Проживающих КАК Проживающих,
        |     СведенияОПомещении.Комнат КАК Комнат
        | ИЗ
        |     РегистрСведений.СведенияОПомещении          КАК
СведенияОПомещении
        | ГДЕ
        |     СведенияОПомещении.IdПользователя           =
&IdПользователя";
    Запрос.УстановитьПараметр ("IdПользователя",
Объект.IdПользователя);
    РезультатЗапроса = Запрос.Выполнить ();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать ();
    Если ВыборкаДетальныеЗаписи.Следующий() Тогда
        ПлощадьОбщаяПомещения =
ВыборкаДетальныеЗаписи.ПлощадьОбщая;
        ПлощадьЖилаяПомещения =
ВыборкаДетальныеЗаписи.ПлощадьЖилая;
        ЭтажПомещения = ВыборкаДетальныеЗаписи.Этаж;
        ПодъездПомещения = ВыборкаДетальныеЗаписи.Подъезд;
        ПроживающихПомещение =
ВыборкаДетальныеЗаписи.Проживающих;
        КомнатПомещение = ВыборкаДетальныеЗаписи.Комнат;
    КонецЕсли;
КонецПроцедуры
#КонецОбласти

```

## Листинг Б.10 – Код формы «ФормаНачисления» обработки «ГлавнаяСтраница»

```

#Область СтандартныеОбработки
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

    Параметры.Свойство("IdПользователя", Объект.IdПользователя);

```

```

        Параметры.Свойство ("ТекущаяЗадолженность",
Объект.ТекущаяЗадолженность) ;
КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии (Отказ)
    Отбор =
СписокНачислений.Отбор.Элементы.Добавить (Тип ("ЭлементОтбораКомпо
новкиДанных")) ;
    Отбор.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных ("IdПользователя") ;
    Отбор.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
    Отбор.Использование = Истина;
    Отбор.РежимОтображения =
РежимОтображенияЭлементаНастройкиКомпоновкиДанных.Недоступный;
    Отбор.ПравоеЗначение = Объект.IdПользователя;
КонецПроцедуры
#КонецОбласти

```

#### Листинг Б.11 – Код формы «ФормаСчетчики» обработки «ГлавнаяСтраница»

```

#Область СтандартныеОбработки
&НаСервере
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)
    Параметры.Свойство ("IdПользователя", Объект.IdПользователя) ;

КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии (Отказ)
    Отбор =
СписокСчетчиков.Отбор.Элементы.Добавить (Тип ("ЭлементОтбораКомпон
овкиДанных")) ;
    Отбор.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных ("IdПользователя") ;
    Отбор.ВидСравнения =
ВидСравненияКомпоновкиДанных.Равно;
    Отбор.Использование = Истина;
    Отбор.РежимОтображения =
РежимОтображенияЭлементаНастройкиКомпоновкиДанных.Недоступный;
    Отбор.ПравоеЗначение = Объект.IdПользователя;
КонецПроцедуры
#КонецОбласти

```

#### Листинг Б.12 – Код формы «ФормаЗаписи» регистра сведений «СведенияОСчетчиках»

```

#Область КомандыФормы
&НаКлиенте
Процедура ПолучитьПоказанияЗаПериод (Команда)
    ДатаПоказаний = Неопределено;

```

Если ВвестиДату(ДатаПоказаний, "Введите дату, на которую нужно получить показания счетчика", ЧастиДаты.Дата) Тогда

```

    Ответ =
    АРІПроцедурыИФункции.ПолучитьПотреблениеСчетчика(Запись.IdСчетчи
    ка, ДатаПоказаний);
    Если ТипЗнч(Ответ) = Тип("Структура") Тогда
        Потребление = Неопределено;
        Ответ.Свойство("consumption", Потребление);
        Сообщить("Текущее потребление = " + Потребление);
    КонецЕсли;
КонецЕсли;
КонецПроцедуры

```

&НаСервере

```

Процедура ПолучитьПоказанияНаСервере()
    ДатаПоказаний = Неопределено;
    Ответ =
    АРІПроцедурыИФункции.ПолучитьПоказанияСчетчика(Запись.IdСчетчика
    );
    Если ТипЗнч(Ответ) = Тип("Структура") Тогда
        НаборЗаписей =
        РегистрыСведений.ПоказанияСчетчиков.СоздатьНаборЗаписей();
        НаборЗаписей.Отбор.IDПользователя.Установить(Запись.IDПользо
        вателя);
        НаборЗаписей.Отбор.IDСчетчика.Установить(Запись.IdСчетчика);
        НаборЗаписей.Прочитать();
        НаборЗаписей.Очистить();
        НомерСтроки = 1;
        Для Каждого СтрокаПоказания Из Ответ.data Цикл
            ЗаписьНабора =
            НаборЗаписей.Добавить();
            ЗаписьНабора.IDПользователя =
            Запись.IdПользователя;
            ЗаписьНабора.IDСчетчика =
            Запись.IdСчетчика;
            ЗаписьНабора.IDСтроки =
            НомерСтроки;
            ЗаписьНабора.ПериодПоказания =
            Дата(СтрЗаменить(СтрокаПоказания.period, "-", ""));
            ЗаписьНабора.ФлагНачальногоПоказания =
            ?(СтрокаПоказания.is_initial = "1", Истина, Ложь);
            ЗаписьНабора.ПоказаниеСчетчика =
            СтрокаПоказания.value;
            ЗаписьНабора.НомерСтроки =
            НомерСтроки + 1;
        КонецЦикла;
        НаборЗаписей.Записать(Истина);
    КонецЕсли;
КонецПроцедуры

```

&НаКлиенте

```

Процедура ПолучитьПоказания(Команда)
    ПолучитьПоказанияНаСервере();

```

```

        ПараметрыФормы      =      Новый      Структура ("IDПользователя,
IDСчетчика", Запись.IdПользователя, Запись.IdСчетчика);
        ОткрытьФорму ("РегистрСведений.ПоказанияСчетчиков.Форма.Форма
Списка", ПараметрыФормы);
    КонечПроцедуры
#КонечОбласти

```

### Листинг Б.13 – Код формы «ФормаСписка» регистра сведений «ПоказанияСчетчиков»

```

&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    Параметры.Свойство("IdПользователя", IDПользователя);
    Параметры.Свойство("IDСчетчика", IDСчетчика);
КонечПроцедуры

&НаКлиенте
Процедура ПриОткрытии(Отказ)
    Отбор =
    Список.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанны
х"));
        Отбор.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных("IdПользователя");
        Отбор.ВидСравнения =
ВидСравненияКомпоновкиДанных.Равно;
        Отбор.Использование = Истина;
        Отбор.РежимОтображения =
РежимОтображенияЭлементаНастройкиКомпоновкиДанных.Недоступный;
        Отбор.ПравоеЗначение = IDПользователя;
        Отбор =
Список.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанны
х"));
        Отбор.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных("IDСчетчика");
        Отбор.ВидСравнения =
ВидСравненияКомпоновкиДанных.Равно;
        Отбор.Использование = Истина;
        Отбор.РежимОтображения =
РежимОтображенияЭлементаНастройкиКомпоновкиДанных.Недоступный;
        Отбор.ПравоеЗначение = IDСчетчика;
КонечПроцедуры

```