

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информационно-коммуникационных технологий в образовании

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОМПАНИИ ООО «ЭЛЕФАНТ»

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСИТ-1501
Института физики, математики,
информатики и технологий
Арасланов Е.Р.

Руководитель: д.п.н., заведующий
кафедрой ИКТО
Стариченко Б.Е.

Работа допущена к защите
« ____ » _____ 2019 г.
Зав. кафедрой _____

Екатеринбург – 2019

Реферат

Арасланов Е.Р. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОМПАНИИ ООО «Элефант», выпускная квалификационная работа: 46 стр., рис. 7, библи. 43 назв., приложений 2.

Ключевые слова: МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ИНФОРМАЦИОННАЯ СИСТЕМА, ПРОЕКТИРОВАНИЕ МП, РЕАЛИЗАЦИЯ МП.

Предмет разработки - мобильное приложение для информационной системы компании ООО «Элефант».

Цель работы - разработка мобильного приложения, автоматизирующего функции расчёта и ведения учёта балконных конструкций в рамках существующей информационной системы компании.

В работе описаны результаты проектирования и программной реализации мобильного приложения, автоматизирующего процессы расчёта необходимых материалов для изготовления балконных конструкций, стоимости этих материалов, а также хранение данных расчётов.

Мобильное приложение реализовано для операционной системы Android на языке программирования Kotlin. Для реализации функций хранения использовались стандартные возможности операционной системы и СУБД SQLite. Для построения запросов использован язык SQL.

Приложение внедрено и прошло апробацию в компании ООО «Элефант».

Оглавление

ВВЕДЕНИЕ.....	4
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ	6
1.1 МОБИЛЬНОЕ ПРИЛОЖЕНИЕ КАК РЕШЕНИЕ ДЛЯ МАЛОГО БИЗНЕСА.....	6
1.2 ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РАЗРАБОТКИ.....	11
1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ МОБИЛЬНОГО ПРИЛОЖЕНИЯ	15
ГЛАВА 2. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОМПАНИИ ООО «ЭЛЕФАНТ».....	20
2.1 МОДЕЛЬНЫЕ ПРЕДСТАВЛЕНИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ	20
2.2 РЕЗУЛЬТАТЫ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ.....	29
2.3 ИТОГИ АПРОБАЦИИ.....	38
ЗАКЛЮЧЕНИЕ	42
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	43
ПРИЛОЖЕНИЯ	48
<i>ПРИЛОЖЕНИЕ 1.....</i>	<i>48</i>
<i>ПРИЛОЖЕНИЕ 2.....</i>	<i>49</i>

Введение

Современные компании, независимо от размера, имея перед собой множество задач, предоставляемых их клиентами, пользователями услуг, заказчиками, стремятся автоматизировать повторяющиеся процессы: передачу однотипных данных, расчёты рабочих формул и подобные. Малый бизнес, не имея в свободном распоряжении значительные денежные ресурсы, предпочтительным имеют использование базовых и ситуативных решений в плане автоматизации. К таким решениям можно отнести ведение учёта расходов в табличных редакторах (LibreOffice Calc, Google Sheets и пр.) и составление отчётов в текстовых редакторах (LibreOffice Writer, Google Docs соответственно), что показывает разрыв в автоматизации между этими двумя процессами, но и является автоматизацией отдельно взятых процессов учёта и отчётности.

Однако, иные компании, в виду своей специфики, множество подходящих под автоматизацию функций выполняют в удалённости от персональных компьютеров и невозможности использования ноутбуков, что приводит к вопросу о мобильных решениях для автоматизации данных функций. Несмотря на существующие аналоги указанных решений (для Google) для мобильных платформ, происходящие процессы не всегда дублируют те, решения для которых у компании уже готовы для настольных устройств.

В данной работе предметом разработки является мобильное приложение, а целью — разработка мобильного приложения для малого бизнеса, занимающегося монтажом балконных конструкций. Конечный продукт разработки — мобильное приложение для автоматизации процесса расчёта размеров и стоимости балконных конструкций для информационной системы компании ООО «Элефант». Исходя из цели работы, выделены были нижеизложенные задачи.

Проанализировать предметную область компании с выделением значимых для реализации указанных функций объектов и дальнейшей

проекции выделенных объектов на формулы расчёта компании в рамках мобильного приложения.

Разработать логический (функциональный) и графический дизайн мобильного приложения совместно с графическими элементами, в него входящими. Под функциональной структурой понимается принцип навигации в приложении и отображения элементов на экране пользователя.

Разработать мобильное приложение, реализующее функции расчёта размеров и стоимости балконных конструкций по формулам расчёта, предоставленных компанией-заказчиком. Согласно сути функции расчёта стоимости, реализована должна быть возможность динамической (без изменения программного кода и перекомпиляции) смены цены на элементарные единицы, формирующие итоговую стоимость.

Провести ввод в эксплуатацию с дальнейшей оценкой его успешности, полноты реализации и формировании рекомендаций по возможным модификациям используемого приложения, согласно опыту разработки и эксплуатации.

Глава 1. Теоретические вопросы разработки мобильного приложения

1.1 Мобильное приложение как решение для малого бизнеса

Решение о разработке мобильного приложения для того или иного бизнеса исходит из различных факторов, среди которых масштаб компании, сфера её деятельности, особенности основных бизнес-процессов и управления ими. Для среднего и крупного бизнеса отмечается наличие множества готовых решений, что возможно к настройке для нужд каждого из подразделений, равно как и для компании в целом; этот факт отражается и в сфере мобильных решений [12]. Перспективы в применении мобильных устройств для малого бизнеса существуют, например, в аналитической сфере [30]. Отмечается и значимость разработки специализированных приложений и систем, заточенных именно под задачи малых компаний, что имеют свою специфику, будь то малая площадь поставки услуг или их узкая специализация [8].

Специфика компании ООО «Элефант», для информационной системы которой создаётся мобильное приложение, заключается в малом штате сотрудников и географической разрозненности задач, выполняемых в рамках предоставления услуги их заказчику. Самой услугой является изготовление и установка алюминиевых балконных конструкций по системам PROVEDAL C640 [27] и PROVEDAL P400 [28]. Изготовление подразумевает проведение замеров, расчёт размеров и стоимости алюминиевых профилей и наполнения для створок, на основе чего производится закупка и изготовление конечных конструкций. Таким образом, обоснованием для разработки специализированного мобильного приложения будут упрощение рабочих процессов внутри фирмы и «мобильность» решения [22].

Исходя из значимости расчёта параметров на основании замеров выходит определение главной задачи для автоматизации: разработка мобильного приложения, возможного к включению в информационную систему компании, что решало бы задачу автоматизации процесса расчёта параметров для

алюминиевых конструкций по формулам компании с дальнейшей возможностью использования рассчитанных данных в бизнес-процессе компании ООО «Элефант».

Перед тем, как выделить конкретные задачи на проектирование, разработку и внедрение мобильного приложения, обозначим суть каждого из этих этапов. После обоснования разработки и принятия решения о необходимости создания автоматизированной системы, необходимо проанализировать существующую у заказчика информационную систему и, проведя обсуждение особенностей будущего продукта и требований к нему [21], составить на основании вышесказанного техническое задание на разработку автоматизированной системы на основании соответствующего стандарта [6]. Следом за формализацией требований в техническом задании идёт этап проектирования, включающий в себя разработку архитектуры приложения как с функциональной (работа с информацией), так и визуальной (дизайн пользовательского интерфейса). На основании спроектированной архитектуры начинается этап разработки, включающий, соответственно, разработку мобильного приложения и создание необходимых графических элементов для пользовательского интерфейса с учётом особенностей их интеграции на конкретной мобильной платформе. Этап введения в эксплуатацию имеет свои особенности, потому подробно рассмотрен будет далее.

При проектировании узкоспециализированного мобильного приложения для малого бизнеса необходимо учесть следующие особенности [3]:

- большое разнообразие специфик предметной области;
- возможность последующего расширения функций;
- целостность и завершённость проекта, минимум риска;
- простота эксплуатации;
- минимум затрат на обучение пользователей.

Заметим, что простота эксплуатации и минимум затрат на обучение пользователей связаны и следует их реализацию возложить на дизайн графического интерфейса пользователя и составление на его основе неформальной инструкции пользователя. Для целостности проекта и возможности расширения функционала в будущем полагается выбор подходящих инструментов для разработки [4], равно как и самого языка. Отметим, что на роль языка, реализующим современные решения, упрощающие мета-программирование, подходит язык программирования Kotlin [17]. Таким образом, для этапа проектирования формулируются две задачи:

1. Проанализировать предметную область компании, выделить в ней значимые для реализации необходимых функций объектов и спроецировать выделенные объекты на формулы расчёта компании в рамках мобильного приложения.
2. Разработать функциональный и графический дизайн мобильного приложения и графические элементы для него.

В процессе анализа предметной области значимым будет рассмотрение существующих решений внутри компании по автоматизации расчётов и ведения учёта, что позволит провести сопоставление значимых объектов и формул расчёта таким образом, что произведённая проекция предметной области на структуры данных и программный код будет наиболее эффективной в решении поставленных задач. Результатом данного этапа проектирования будут структуры данных для оперируемых в предметной области явлений: алюминиевого профиля, балконной створки и так далее; а также предполагаемые способы их хранения на устройстве (как таблицы базы данных, в отдельных файлах или иным образом).

При проектировании функционального дизайна, как уже было отмечено, стоит обратить особое внимание реализации возможности последующего расширения функционала, что может быть достигнуто за счёт слабой связанности объектов, либо же использования элементов структурного и

функционального проектирования (для чего, в свою очередь, подходит ранее выделенный язык программирования Kotlin). Требованием к графическому дизайну и его элементам является неперегруженность пользовательского интерфейса и максимально возможная информативность используемых изображений.

Учитывая специфику разработки, этапы разработки, тестирования и внедрения решено было совместить, сократив нагрузку на ручное тестирование через частый выпуск обновлений, включающих как новый функционал, так и исправление замеченных во время пользования недоработок. Также подобное смешение призвано улучшить процесс обучения работе с новым решением [10]. Таким образом, на данный этап выводятся две задачи:

1. Разработать мобильное приложение, реализующее функции расчёта размеров и стоимости балконных конструкций по формулам расчёта, предоставленных компанией-заказчиком. При разработке учесть требование к динамической корректировке цены на элементы, составляющие итоговую стоимость.
2. Провести ввод в эксплуатацию с дальнейшей оценкой его успешности, полноты реализации и формировании рекомендаций по возможным модификациям используемого приложения, согласно опыту разработки и эксплуатации.

Этап разработки включает в себя фактическую реализацию проекта приложения, созданного на предыдущем этапе: написание программного кода, создание иконок и изображений для графического интерфейса в соответствующих редакторах. Для успешного завершения этапа необходимо определение особенностей версии операционной системы, используемой на целевом продукте [32], её возможностей и ограничений, а также выборе соответствующего инструментария, что позволит сократить время на разработку. Выбор инструментов для разработки должен основываться на

необходимости минимальных денежных затрат на разработку, что является особенностью разработки мобильных приложений для малого бизнеса.

Перед выпуском первой версии, что будет передана в тестовое пользование заказчику, необходимо провести проверку адекватности работы формул, чтобы предотвратить возможные потери на первых этапах ввода приложения в эксплуатацию. Далее необходимо дорабатывать проблемные места, обнаруженные в ходе реализации новых функций, а также обнаруженные заказчиком в ходе тестовой эксплуатации. Итоговый вариант приложения должен быть представлен в соответствующие техническому заданию сроки.

Оценка эффективности внедрения информационной системы и, соответственно, качества созданного продукта это сложная задача, к которой существуют методики [5], применимые к крупным компаниям с большими рисками. Несмотря на это, в сфере малого бизнеса проблемой является низкое качество производимых решений [31]. Опираясь на исходную причину и основную задачу разработки мобильного приложения, а также специфику условий его создания, качественно исполненным приложением будем считать удовлетворённость конечного пользователя результатом, а также достижение всех поставленных выше задач.

Таким образом, по итогам разработки необходимо провести не только оценку разработчиком своего продукта, но и обработать результаты введения в эксплуатацию. Так как конечный пользователь это один человек, то его мнение полагается быть узанным через интервью, состоящем из вопросов по использованию значимых элементов приложения, а также освещающих потенциально проблемные или неоднозначные места в разработанном мобильном приложении. По результатам проведённого интервью должны быть сделаны выводы об успешности разработки и внедрения мобильного приложения, а также выведены рекомендации по возможной модернизации.

Подводя итог вышесказанному, для выполнения поставленных четырёх задач необходимо:

1. Подготовить техническое задание, согласно требованиям заказчика.
2. Подготовить проект системы, включающей функциональный и графический дизайн.
3. Выбрать подходящие инструменты для реализации поставленных задач.
4. Создать необходимые графические элементы.
5. Разработать мобильное приложение.
6. Провести анализ результатов разработки и внедрения, сформулировать рекомендации к модернизации мобильного приложения.

1.2 Выбор программного обеспечения для разработки

Набор программного обеспечения, необходимого для разработки конечного продукта, будет зависеть от множества факторов, как влияющих на минимальный набор программного обеспечения, так и на дополнения для него, обеспечивающие саму возможность или удобство использования. Среди этих факторов:

- платформа, под которую разрабатывается приложение;
- языки программирования, для этого используемые;
- библиотеки и фреймворки, необходимые для решения поставленных задач;
- необходимость проектирования структуры приложения, его сложность и глубина детализации;
- необходимость создания элементов графического интерфейса.

Разрабатываемое приложение целевой платформой имеет операционную систему Android и не имеет зависимостей от каких-либо существующих ресурсов, будь то приложения для настольных платформ или веб-сайты. Таким образом, разрабатываемое приложение может быть классифицировано как нативное приложение [20] для мобильных устройств под управлением

операционной системы Android. Иные мобильные операционные системы [23] не используются заказчиком в работе, что определяет множество инструментов, используемых для разработки [14].

Разработка под выбранную мобильную платформу возможна на множестве языков программирования, некоторые из которых являются основными, или официальными, языками, поддержка других же реализована сторонними разработчиками [38]. К первой группе относятся Java, C++ [33], Kotlin. Ко второй можно отнести языки Python (проект Kivy), C# (Xamarin [11]), Basic (B4X [35]).

Использование сторонних проектов, таких как Xamarin или Kivy, могут иметь проблемы с размером файлов [7] приложения и сложностью оптимизации под целевую платформу [19]. Вместе с тем, корпорация Google, являющаяся разработчиком Android, объявила язык Kotlin [39] официально поддерживаемым в их операционной системе в 2017 году [26].

Kotlin имеет возможность работать на основе Java Virtual Machine [43], что позволяет использовать продукты экосистемы Java, но отличается лаконичностью синтаксиса, в отличие от Java. Kotlin, при этом, защищён от NullPointerException [40] и связанных с этим проблем в тестировании. Для этапа разработки и проектирования также ценны классы данных [36] и перечислений [37], направленными на написание более компактного кода и структурирования отношений между данными с помощью элементов перечисления в качестве маркеров типа вместо перегруженной системы наследования. Таким образом, Kotlin является более предпочтительным языком для разработки под Android, чем Java. C++ хоть и является частью экосистемы Android, но используется для специфических задач [24], к которым определённая в рамках данной работы не относится.

Фреймворки являются инструментами для упрощения работы с кругом часто встречающихся задач, либо же решениями специфических вопросов, компенсирующими недостатки языка или платформы. Выбор фреймворков для

разработки программного продукта, как и выбор менее комплексных решений — библиотек, обусловлен фактором временной выгоды, происходящей также от уровня владения ответственными разработчиками рассматриваемым инструментом. В случае разработки данного мобильного приложения, исходя из навыков разработчика и предполагаемой модели разработки, подразумевающей наиболее быстрый релиз первой работоспособной версии (рабочего прототипа), нет необходимости в использовании каких-либо фреймворков.

Для мобильной операционной системы Android, в зависимости от классификации, существует три [25] или четыре [13] варианта работы с данными из приложения: внешнее и внутреннее хранилища (как отдельные файлы), Shared Preferences (хранимые настройки приложения), система управления базами данных SQLite [42] (файл базы данных). Для основного функционала приложения (работы с заказами) подходит последний вариант, когда данные хранятся в базе данных под управлением SQLite, так как обозначенные выше возможности языка Kotlin (а именно Data Class) хорошо переносятся на базу данных при её структурном проектировании [18]. Для хранения цен и коэффициентов для расчёта конечной стоимости заказов и для логического отделения данных параметров от самих параметров заказов предполагается использовать Shared Preferences, как способ их хранения. При реализации экспериментальных функций, связанных с хранением текстовых заметок, вариант с хранением в виде отдельных файлов является адекватным, ибо позволяет, при необходимости, модернизировать внутреннюю структуру текстовых файлов для создания шаблонов для различных типов заметок и прочих манипуляций с хранимым в них текстом.

Дизайн графического интерфейса представляет из себя принятие решений [16] о реализации формирующих факторов [15], среди которых: единство дизайна, визуальная иерархия, стилистическое единство иконок, цветовая гамма [2]. Визуальная иерархия связана с навигацией, потому главным

критерием успешности дизайна для неё будет являться доступность всех функций. Единства дизайна предполагается достигнуть за счёт разработки набора иконок специально для приложения. Для гибкости при работе с цветом созданных иконок, они должны быть в формате векторных изображений (SVG [41]), что делает возможным конвертацию во внутреннее представление векторных изображений Android, как с привлечением сторонних сервисов, так и возможностями Android Studio [34]. Помимо формата, необходимо уточнить, что в итоговый набор разработанных иконок должны входить подгруппы, имеющие внутреннюю логику дизайна, отличающую их друг от друга по назначению. Так, иконкам верхней панели следует применить дизайн, отличный от иконок нижней панели, а им, в свою очередь, от навигационных иконок.

Под основные мобильные операционные системы существуют собственные среды разработки (Integrated Development Environment, IDE), предоставляющие необходимые инструменты для разработки [1]. Для операционной системы Android существуют различные IDE: IntelliJ IDEA, Eclipse, Android Studio. Последняя среда рекомендуется Google в качестве среды для разработки под платформу Android [29].

Android Studio является совместным продуктом компаний JetBrains (разработчиков IntelliJ IDEA и языка Kotlin) и Google, представляя собой модифицированную версию продукта JetBrains. В то же время, IntelliJ может быть модифицирована с помощью плагинов для разработки Android-приложений. Eclipse также может быть настроен для мобильной разработки, но исходная версия слабо адаптирована для данной цели. Учитывая обозначенные факторы и факт длительного использования разработчиком именно IntelliJ IDEA, настроенную под Kotlin и Android разработку, выбор производится в пользу данной IDE, так как освоение нового инструмента, равно как и его настройка, не являются необходимыми или окупаемыми временными затратами.

Помимо написания программного кода, необходим также инструмент для создания векторных изображений, иконок. Существует несколько решений, позволяющих работать с векторными изображениями, как платных (Corel Draw, Adobe Illustrator), так и бесплатных, предлагающих реализацию аналогичных функций. Одним из таких бесплатных решений является Inkscape. Данный редактор кроссплатформенный, работает в режиме оффлайн. Интерфейс стандартный, подобен офисным решениям LibreOffice, где имеется верхняя панель, панель инструментов слева и скрываемое меню дополнительных настроек справа, из чего следует удобство данного решения в возможности настройки рабочего пространства без множества отдельных окон, регулярно перекрывающих рабочую область (подобные решения были в старых версиях растрового графического редактора GIMP). Таким образом, Inkscape является достаточным решением в качестве инструмента создания иконок для разрабатываемого приложения.

Сформулировав цели и определившись с необходимыми инструментами и основными технологическими решениями для будущего проекта, невыбранной остаётся методология разработки. Проект отличается небольшим объёмом запланированных работ, необходимость раннего ввода в эксплуатацию, хорошая делимость задач на функциональные группы. Таким образом, наиболее подходящей является «инкрементная модель» [9], хотя возможно и определение методологии как каскадная модель с элементами инкрементной. Вне зависимости от формального обозначения, процесс проектирования, разработки и ввода в эксплуатацию предполагается следующий:

1. Разработка первой рабочей версии с функцией расчёта конструкций, параллельно создаётся дизайн отдельных сцен, разрабатывается набор иконок.
2. Отладка функционала первой рабочей версии после ввода её в эксплуатацию, создание следующей версии, основанной на разработанном дизайне.

3. Добавление новых функций и корректировка численных значений графического интерфейса до состояния реализации всех заявленных функций.

Таким образом, все обозначенные в начале пункта 1.2 факторы, влияющие на выбор программного обеспечения и технологий, были рассмотрены. В качестве инструментов были выбраны IntelliJ IDEA в роли IDE, а для работы с графическими задачами — Inkscape. После выбора инструментов было дано описание модели разработки на основе поставленных задач, выбранных инструментов и особенностей разработки.

1.3 Техническое задание на разработку мобильного приложения

На разработку мобильного приложения «Elephant Next»

1. Общие сведения

- 1.1. Заказчик: ООО «Элефант» по сборке и монтажу балконных конструкций.
- 1.2. Название продукта: Elephant Next.
- 1.3. Продукт: мобильное приложение для автоматизации процесса расчёта размеров и стоимости производимых компанией балконных конструкций, ведения их учёта.
- 1.4. Начало разработки установить на 18 июля 2018 года, окончательную дату внедрения основной версии на 1 февраля 2019 года.

2. Характеристика области применения продукта

- 2.1. Мобильное приложение Elephant Next, согласно описанному в пункте 1.3. функционалу, предполагается к применению в ходе управленческих процессов компании. Мобильное приложение связывает процесс получения информации о заказе (размеров, типа конструкции, их количестве), процесс оформления договора о

предоставляемых услугах (расчёт стоимости заказа) и процесс предоставления услуг (соотношение имеющихся в наличии элементов конструкции с необходимыми для реализации заказа).

2.2. Мобильное приложение рассчитано на пользование одним человеком, получать информацию, полученную посредством приложения, имеют возможность третьи лица по воле пользователя мобильного приложения. Квалификация основного пользователя оценивается как удовлетворительная для работы со стандартными мобильными приложениями (не менее трёх лет использования мобильных устройств).

3. Требования к продукту разработки.

3.1. Конечный продукт разработки - мобильное приложение с определённым в ходе обсуждения с заказчиком функционалом. Основным функционал был определён следующий:

3.1.1. Расчёт расходных материалов (профилей и заполнения), необходимых для изготовления конструкции определённого типа (2-х, 3-х, 4-х створок и глухой) и стоимости всей конструкции (с учётом стоимости изготовления). Размеры выводятся в мм., стоимость в рублях РФ.

3.1.2. Хранение заказов, в которые включаются рассчитанные конструкции, с возможностью учёта в них дополнительных материалов (специальных профилей, не учитывающихся для отдельной конструкции) и расчётом общей стоимости заказа. Должна быть предусмотрена возможность редактирования существующих заказов.

3.1.3. Установка стоимости материалов конструкции (суммарно за все) и работы, стоимости отдельных профилей, дополнительных для заказов.

- 3.1.4. Передача информации о входящих в заказ конструкциях (тип, общие размеры, размеры заполнения) и дополнительных материалах в текстовом формате через подходящие приложения, установленные на устройстве пользователя (прим. мессенджеры) работникам (сборщикам, монтажникам).
- 4. В тестовом режиме предполагается реализация ведения простых текстовых заметок.
- 3.2. Продукт разрабатывается для смартфонов. Оптимизация под иные мобильные платформы не предусмотрена.
- 3.3. Приложение разрабатывается для использования на операционных системах семейства Android не младше 5 версии. Для использования функции отправки данных о заказах (пункт 3.1.4) необходимы соответствующие установленные приложения на устройстве пользователя.
- 3.4. Разработка будет произведена на языке программирования Kotlin. Для проектирования и разработки используется свободно распространяемое программное обеспечение.
- 3.5. Приложение не рассчитано на создание доступных извне файлов. Данные о заказах хранятся в локальной базе данных на устройстве пользователя.
- 3.6. Так как в рамках приложения не используется закрытая корпоративная информация, то мер защиты со стороны приложения не предусматривается.
- 4. Требования к пользовательскому интерфейсу.
 - 4.1. Пользовательский интерфейс включает в себя подразделы, соответствующие функционалу (работа с новыми конструкциями, управление заказами и пр.), навигация между ними происходит, исходя из производимых действий (прим. после расчёта

конструкции, становится возможным добавление её к существующим заказам или формирование из него нового с переходом в соответствующий раздел), предоставляя, где возможно, доступ к разделам разными способами (навигационная панель, действия с элементами раздела).

- 4.2. Экран делится вертикально на три части: верхняя навигационная панель, рабочая область, нижняя навигационная панель. Верхняя панель состоит из иконок-кнопок трёх разделов: заметок, конструкций, управления. Основной раздел, размещённый по центру, включает в себя все операции с заказами и конструкциями. Раздел управления (правая кнопка) предоставляет доступ к изменению стоимости конструкций; здесь же предполагается добавление функций по хранению и обмену данными, если такой функционал будет создаваться в последующих проектах. Кнопка раздела заметок находится слева, предоставляя доступ к работе с заметками. В случае обнаружения ненужности функционала заметок по итогам опыта использования приложения заказчиком, рекомендуется занять этот раздел управлением хранимыми заказами. В рабочую область подгружаются элементы графического интерфейса, соответствующие текущему разделу. Нижняя навигационная панель содержит кнопки навигации между большими разделами (перемещение по верхней панели вправо-влево), а также до трёх активных кнопок, иконки и функционал которых зависит от текущего контекста приложения (сосчитать конструкцию, сохранить настройки и пр.). Цвета для интерфейса должны сочетаться, не напрягать зрение пользователя. Элементы графического интерфейса не должны быть перегружены, но подчинены общему стилю приложения.

5. Требования к документированию.

В дальнейшем, составляется итоговая структура данных в визуальном представлении, руководство пользователя, включающее скриншоты работающего приложения, а также последовательность действий для выполнения той или иной задачи.

6. Порядок сдачи-приёмки продукта.

Первая рабочая версия должна быть передана заказчику до 1 сентября 2018 года без составления дополнительных документов о передаче. Каналом для передачи выбрана электронная почта. Минорные версии передаются по тому же каналу, когда реализуются или дорабатываются функции по плану их введения или по просьбе заказчика (прим. изменение форматирования передаваемых третьим лицам сообщений). В период до 1 февраля 2019 включительно должен быть заключён договор приёмки окончательной версии мобильного приложения, подтверждающий успешное выполнение требований заказчика. По согласию заказчика, могут быть также составлены документы об использовании приложения в работе заказчика, содержащих степень удовлетворённости продуктом и проч.

Глава 2. Мобильное приложение для информационной системы компании ООО «Элефант»

2.1 Модельные представления мобильного приложения

При разработке мобильного приложения необходимо выделить значимых элементов предметной области для собирания их в классы, что будут функциональными элементами в рамках программного кода. После анализа актуальной для компании-заказчика предметной области, были выделены классы (см. Рисунок 1): профиль, конструкция, сделка. Таким образом, минимальной структурной единицей является профиль, а сделка представляет из себя композицию конструкций и профилей: именно в формате сделки хранится информация, связанная непосредственно с предметной областью. Основным функциональным элементом в приложении является конструкция, так как от её характеристиками детерминировано количество необходимых профилей и их длина.

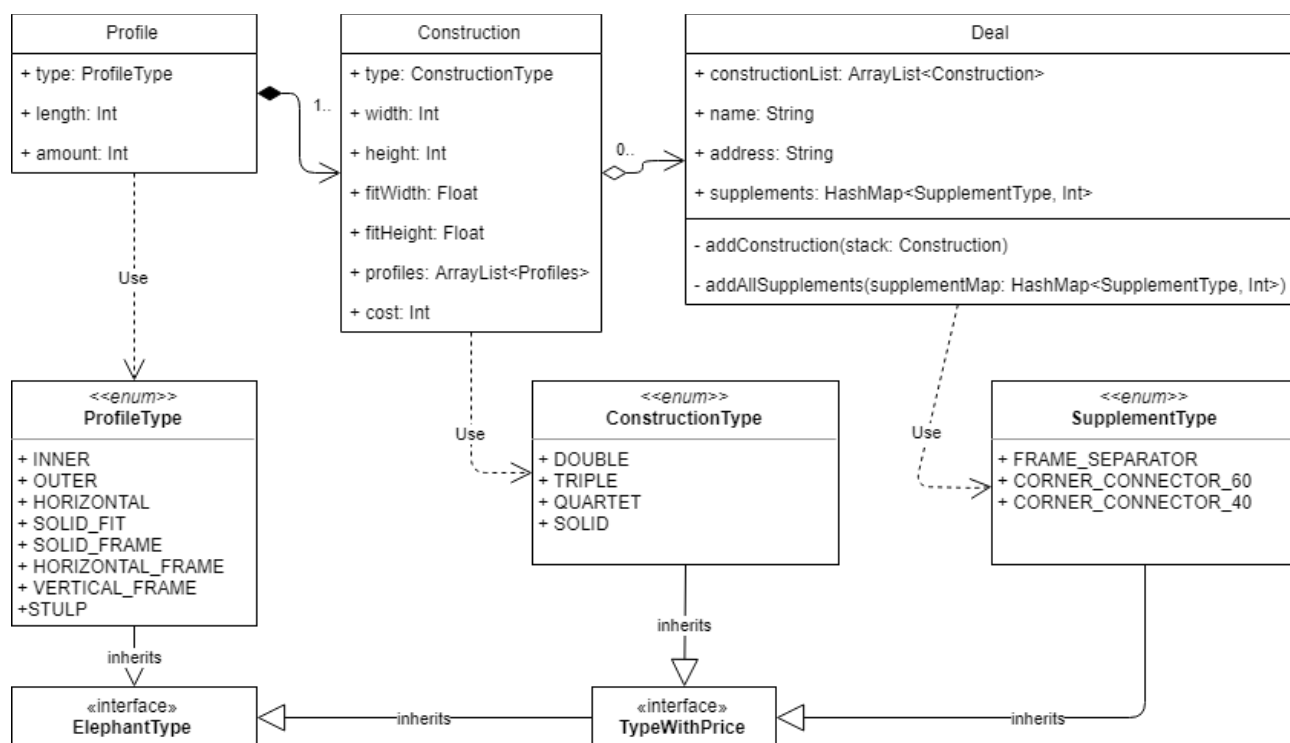


Рисунок 1- Диаграмма классов

Значимой характеристикой для профилей и конструкций является их тип. Профили делятся на конструкционные и комплектные. Первые полностью

зависят от того, в какого типа конструкцию включены и какие у неё параметры ширины и длины. Вторые включаются в сделки как композиционный элемент и в рамках разрабатываемого приложения, имеют значимой лишь характеристику количества. Конструкции также имеют тип, называющийся вслед за количеством створок, из которых они состоят.

По итогу реализации предметной области в программном коде, были созданы три класса-перечисления для типов профилей, что наследовали два интерфейса-маркера для достижения необходимого, с точки зрения чистоты кода, уровня абстракции:

- интерфейс `ElephantType`, которым помечаются (наследуют от него) все типы;
- интерфейс `TypeWithPrice` для типов, у которых должна быть реализована функция расчёта стоимости;
- перечисление (enum) `ProfileType`, описывающее конструкционные профили и имеющий возможные значения: `INNER` (внутренний створки), `OUTER` (внешний створки), `HORIZONTAL` (горизонтальный створки), `SOLID_FIT` (внутренний глухой), `SOLID_FRAME` (глухая рама), `HORIZONTAL_FRAME` (горизонтальная рама), `VERTICAL_FRAME` (вертикальная рама), `STULP` (штуп);
- перечисление `SupplementType` (`TypeWithPrice`), описывающее комплектные профили, с возможными значениями: `FRAME_SEPARATOR` (разделитель рам), `CORNER_CONNECTOR_60` (соединительный угол на 60), `CORNER_CONNECTOR_40` (соединительный угол на 40);
- Перечисление `ConstructionType` (`TypeWithPrice`) со значениями, соответствующими типам конструкций: `DOUBLE` (конструкция 2-х створок), `TRIPLE` (3-х створок), `QUARTET` (4-х створок), `SOLID` (глухая конструкция).

Для абстракций профиля и конструкции были созданы классы данных (data class), предоставляющие необходимый функционал по хранению и предоставлению данных. Все поля были объявлены константными значениями (val). Для класса профиля установлены поля, отвечающие за тип, длину и количество. Для конструкции: тип, ширина конструкции, высота конструкции, ширина заполнения, высота заполнения, профили (списком), стоимость. Сделка определена как класс с изменяемыми (mutable) полями названия, адреса; неизменяемыми (immutable) полями списка конструкций и словаря (Map) дополнительных профилей. Из методов, помимо конструктора, определены добавление конструкции и добавление множества дополнительных профилей. Дополнительные профили задаются не классом профиля, а парой тип-количество, так как для них не имеет значение длина, являющаяся параметром в определении обычного профиля.

Упомянутые классы включены в пакет (package) core, в котором собраны определения основных типов данных в формате классов и перечислений, используемых для внутренней работы приложения. Дополнительно к ним, отражающим предметную область, добавлены были классы данных заголовка сделки и заметки, используемые в ходе работы приложения.

Пакет engine содержит два файла с расширением .kt, отвечающими за расчёты (Calculator) и преобразование (Converter). В каждом из них определены соответствующие методы, необходимые для реализации функционала. В Calculator публичными являются методы расчёта стоимости сделки, стоимости дополнений в сделке, количества конструкций в сделке по типу, всей конструкции. Согласно принципам функциональной парадигмы, возвращаемые результаты полностью зависят от входных параметров. Прочие функции являются вспомогательными и имеют модификатор private. Converter реализует различные методы преобразования конструкций, сделок в строки с различным форматированием (для вывода общей информации, для диалога и пр.) и типов в их названия и наоборот.

Пакеты `core` и `engine` отвечают, как уже отмечалось ранее, за внутреннее функционирование мобильного приложения. За работу с графическими ресурсами, что находятся в директории `res`, стандартной для проектов для операционной системы Android, отвечает пакет `view` и файл `ViewManager` в пакете `managers`.

В отличие от пакета `view`, предоставляющего утилитарные функции для инициализации элементов графического интерфейса и смены сцен, пакет `managers` содержит менеджеры локального хранилища (`LocalManager`), настроек (`SettingsManager`), визуальной составляющей (`ViewManager`) и состояния приложения (`AppManager`). `ViewManager`, как и большинство файлов пакета `view`, работает с графическими элементами, формируя словари (`map`) с элементами, необходимыми для каждой из сцен. `SettingsManager` реализует работу с классом данных `Prices` и предоставляет функционал для получения всех цен и коэффициентов, либо для конкретной группы. Через этот же менеджер идёт изменение и сохранение цен, для чего используется стандартное средство работы с настройками в Android — класс `SharedPreferences`. `AppManager` служит для координации действий пользователя с хранимыми данными, потому в нём реализованы функции для взаимодействия с базой данных и записями через `LocalManager` и отслеживания времени изменений в целях оптимизации количества операций чтения-записи. Помимо этого, `AppManager`, являясь, как и остальные менеджеры, `object` (реализацией паттерна `singleton` в языке Kotlin), хранит данные о текущей рассчитанной конструкции, идентификаторе текущей редактируемой сделки, идентификаторе текущей редактируемой заметки, что позволяет хранить эти данные в единственном экземпляре, не допуская десинхронизации в работе программы. `LocalManager`, в свою очередь, предоставляет необходимые методы для работы как с локальной базой данных (в реализации `SQLite`), так и с записями, хранимыми в текстовом формате. Структура базы данных, запросов к ней, а также особенности обработки записей рассмотрены далее в работе.

Учитывая описанную структуру отображения предметной области на иерархию классов и основные моменты её реализации в программном коде, основной класс мобильного приложения MainActivity задаёт следующий алгоритм работы (см. Рисунок 2).

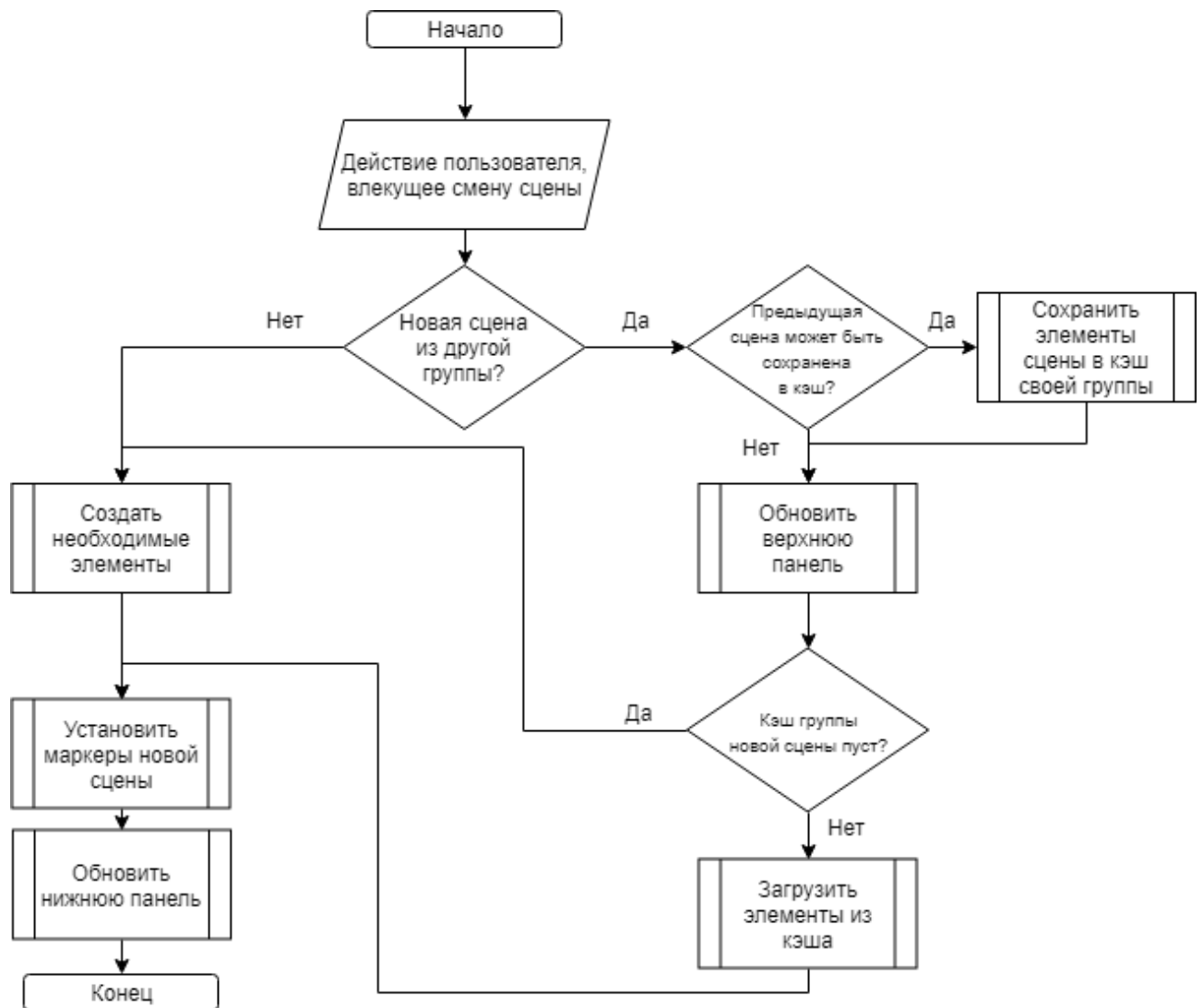


Рисунок 2- Алгоритм работы MainActivity

Проверяется, является ли вызываемая сцена последней сохранённой в «закладке» (для сцен, что сохраняются), после чего словарь элементов загружается из кэша, хранимого в companion object MainActivity, либо создаётся вызовом метода утилитарных классов. Из словаря в ходе выполнения метода соответствующей сцены передаются нужные элементы в группу центральных элементов. Переопределяются кнопки на нижней панели, которым задаются соответствующие иконки и функционал.

Выполнение действий кнопок нижней панели вызывает иную цепочку действий. Если нажатие на кнопку меняет сцену, вызывается метод `changeScene`, регулирующей не связанные с конкретной сценой эффекты (сохранение сцен в кэш и смена активных кнопок на верхней панели) при смене «закладки». Завершается этот метод инициализацией сцены и установкой идентификатора его как текущего активного. Подобные действия, а также расчёты конструкций и сохранение изменений, требуют обращения к методам `AppManager` для вывода необходимой информации на экран и изменения текущего состояния приложения. Методам, связанным с ценами, передаются в качестве параметров результаты обращений к методам `SettingsManager`.

Таким образом, основной поток приложения взаимодействует с тремя менеджерами (кроме `LocalManager`), через которые и получает данные для обработки уже функциями пакета `engine` типов из пакета `core`. Подобная структура работы основного класса приложения позволяет в явном виде работать со всеми элементами проекта, что исключает сильную связанность самих элементов друг с другом, позволяя, в перспективе, тестировать их функционал отдельно, либо же заменить поток исполнения приложения без необходимости изменять сами классы и их функционал.

Данные приложения хранятся на мобильном устройстве пользователя в трёх видах: как файл настроек (назначение которого описано выше), как база данных, как текстовые файлы. Заметки хранятся в формате текстовых файлов, что располагаются в директории «notes» в отведённом под это месте в рамках дискового пространства устройства пользователя. Название каждого файла заметок содержит его название и дату последнего сохранения, что позволяет выводить список хранимых заметок по заголовкам. Такой подход выбран в связи с достаточностью в отношении экспериментальной функции, не вызывающий дополнительных расходов на изменение базы данных. База данных, в свою очередь, сосредоточена на хранении информации о сделках, потому состоит из следующих таблиц (схема на Рисунок 3):

- deals, что представляет сами сделки через идентификатор, имя, адрес и дату сохранения;
- supplements, где хранится информация о дополнительных профилях, привязанных к сделке по идентификатору сделки;
- constructions с конструкциями, привязанными также по идентификатору сделки.

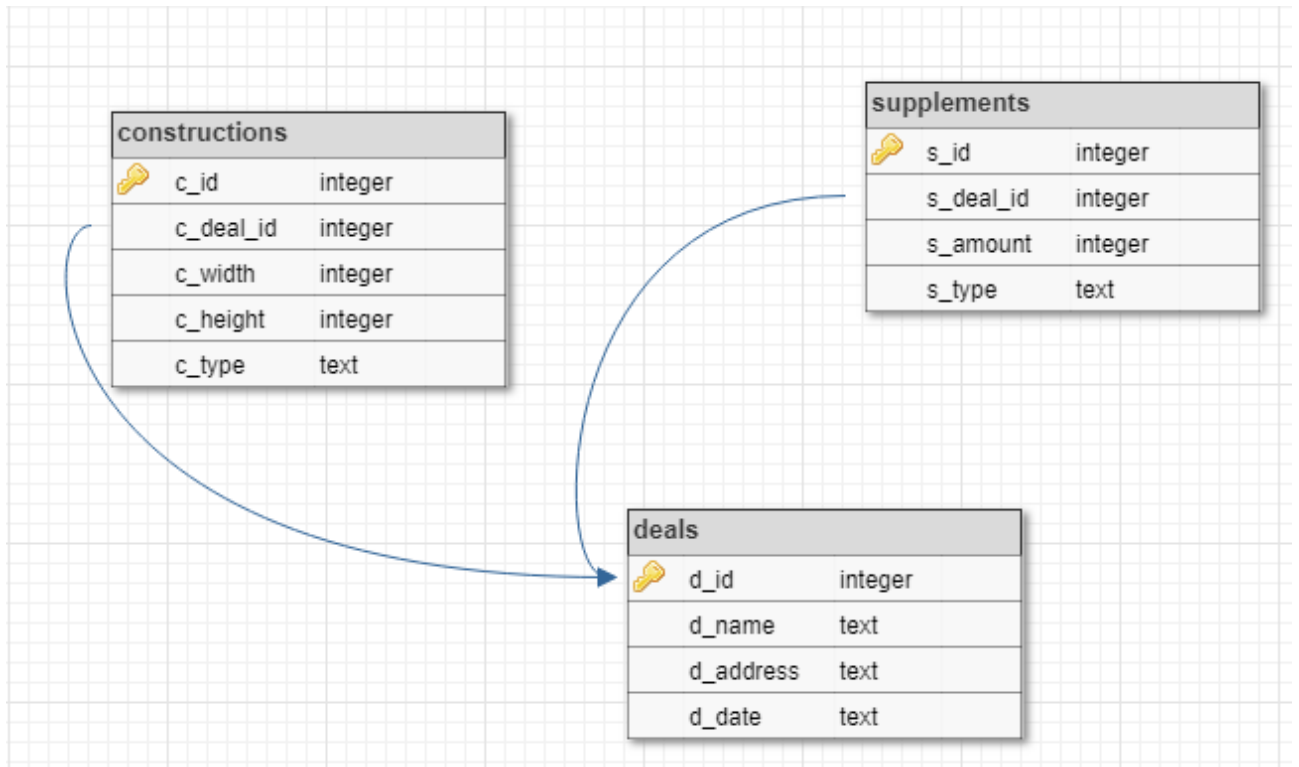


Рисунок 3 - Структура базы данных

Несмотря на почти идентичные поля классов данных дополнительных профилей и конструкций с представлением в базе данных, для реализации работы с базой данных используется структурный подход, а не объектный, так как при таком количестве таблиц, он является достаточным в плане соотношения сложности использования и скорости разработки. Также LocalManager реализует необходимые функции по созданию, обновлению удалению сделок и их составляющих через прямые SQL-запросы. В этом же менеджере происходят преобразования в соответствующие сущности для дальнейшей обработки в ходе работы приложения.

Пользовательский графический интерфейс является наиболее значимой частью мобильного приложения, потому что важно как спроектировать его адаптивным, так и избежать запутанности и отсутствия целостности. Исходя из обозначенных предпосылок и предписанного к реализации функционала, был разработан общий дизайн графического интерфейса для приложения. Визуально видимый пользователем экран разделён вертикально на три части: центральную область, верхнюю и нижнюю панели.

Верхняя панель служит для навигации между «закладками», что представляют собой наборы функционально взаимосвязанных сцен. Это наиболее статическая часть, чья роль состоит в информировании пользователя о текущей «закладке», а также существовании иных групп функций, что напрямую не связаны с его текущей деятельностью. «Закладки» также являются отображением логических разделов, подробное описание работы которых располагается в пункте 2.2. Нижняя панель предусмотрена для навигации в рамках одной «закладки» и является главным интерактивным элементом. Центральная область наиболее динамический элемент из трёх: работа с обрабатываемой приложением информацией происходит на ней, составляющие её зависят от обрабатываемой в конкретный момент функцией.

Таким образом, графический интерфейс логически разбит на 3 функциональных блока («закладки»), в каждом из которых доступна самостоятельная навигация. Подобная структура позволяет обращение к разным группам функций параллельно, без потери состояния текущей группы. Поскольку переход между группами функций предполагается редким, расположение иконок с обозначением активной «закладки» на верхней панели является рациональным. Вместе с тем, при необходимости, функциональные блоки могут быть добавлены, что отразится на верхней панели, также как она, при сокращении функциональных блоков, может быть скрыта без ущерба для работоспособности приложения и навигации в нём.

Расположение основных контекстных интерактивных элементов на нижней панели обусловлено удобством манипуляции приложением одной рукой без необходимости обращения к удалённым элементам для решения задач той же частоты использования, что и основные. Предполагается, что размещение более трёх изменяемых кнопок, совместно с крайними двумя, что облегчают навигацию между функциональными группами, является разумным ограничением на количество контекстуальных элементов, что реализуют основные функции той или иной сцены, представленной пользователю в конкретный момент времени.

Остальные же взаимодействия, такие как ввод данных, либо чтение хранимой информации, привязаны к центральной области, наполнение которой зависит от каждой сцены в отдельности. Добавление новой функции в приложении предполагает основным этапом создание элементов для центральной области, так как её содержание не ограничено кнопками, как содержание панелей. Подобная архитектура графического интерфейса пользователя после своей реализации в программном коде не препятствует его модернизации, являясь при этом логически целой.

Процесс разработки приложения, в целом, соответствует каскадной модели разработки программного обеспечения. Отличительными чертами является создание прототипа базового функционала приложения (расчёта конструкций) до этапа дизайна, а также начало внедрения приложения на ранних версиях.

Прототип, также являющийся первой переданной заказчику для тестирования версией, реализовал функцию расчёта конструкций. Так как разработчик приложения один, то развитие наработок, что представляли собой прототип, не представлялось проблемным в совмещении с этапом проектирования итогового графического интерфейса. Разработка прототипа также позволяет оценить объём работы, а также задаёт отправную точку для формирования структурированной кодовой базы.

Раннее введение приложения в эксплуатацию, обозначенное в техническом задании, позволяет специализировать приложение для работы под конкретную аппаратно-программную платформу, коей является мобильное устройство заказчика, ибо конечным пользователем подразумевается он один. Недочёты в работе графического интерфейса или отображения текстовой информации, а также минорная корректировка расчётных формул возможны через обозначенный подход. Также, он упрощает общий процесс тестирования, что, в рамках разработки приложения одним человеком, является преимуществом.

2.2 Результаты разработки мобильного приложения

В ходе реализации проекта мобильного приложения «Elephant Next» было создано несколько промежуточных версий, которые передавались на пользование заказчику для использования и тестирования. Соответственно, каждая из промежуточных версий имела больший функционал, чем предшествующая, имея, помимо этого, обнаруженные заказчиком недочёты исправленными. После того, как все заявленные в проекте приложения функции были реализованы, была выпущена релизная версия под номером 1.0.0, которая и является конечной целью разработки данного проекта. Все изменения, возможные в исходном коде версии 1.0.0, являются косметическими, не затрагивающими основной функционал приложения, потому результаты разработки мобильного приложения будут рассматриваться для указанной версии.

При запуске приложения, пользователь видит сцену расчёта конструкции по параметрам ширины, высоты и типа конструкции (по центру на Рисунок 4). На поля ввода ширины и высоты указывают текстовые подсказки (hint), пропадающие при вводе данных. Тип конструкции задаётся через выбор нужного значения в выпадающем списке. Названия типов отображаются как сокращения, известные в данной сфере производства: 2-х, 3-х, 4-х, Гл. Они,

соответственно, отвечают за конструкции с двумя, тремя, четырьмя створками, а также за глухую конструкцию, створок не имеющую. Помимо элементов для установки входных параметров, первая сцена имеет текстовое поле для вывода полученных значений, изначально имеющее надпись «Нет конструкции».

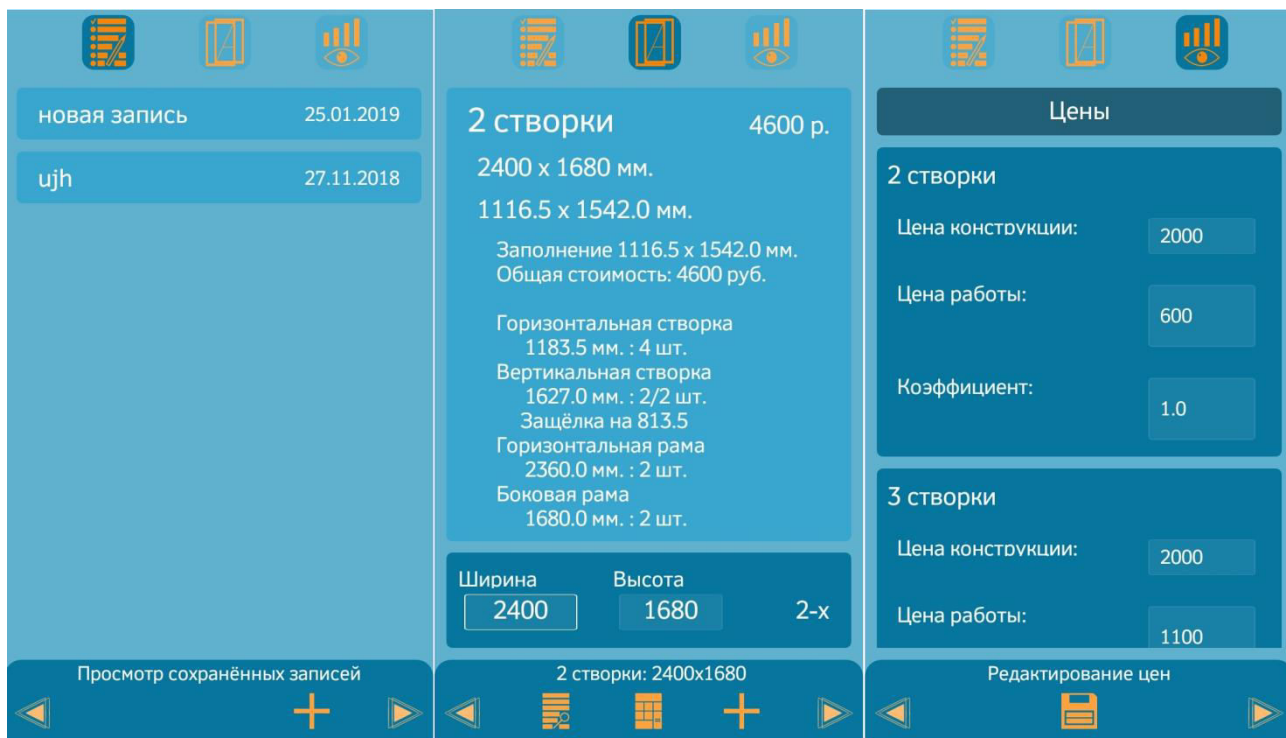


Рисунок 4 - Начальные сцены каждой «закладки»

Расчёт конструкции и вывод результатов осуществляется после нажатия на кнопку со схематическим изображением калькулятора, находящуюся на нижней панели, описание функционала и изменчивости которой будет приведено ниже. В случае поступления некорректных для расчёта данных, текстовое поле выведет сообщение «Неправильный ввод данных», иначе будут показаны результаты расчётов. В список выводимых параметров входят:

- тип конструкции;
- цена конструкции;
- ширина и высота конструкции;
- ширина и высота заполнения одной створки (либо глухой конструкции);
- длина профилей горизонтальных и вертикальных, если они есть, их количество;

- длина профилей горизонтальных и вертикальных рам, их количество.

Нижняя панель содержит кнопки управления и строку состояния. Строка состояния выводит контекстуальные подсказки о текущем действии или хранимом в памяти состоянии (например, о рассчитанной конструкции для добавления к существующим заказам). Крайние кнопки служат для навигации между «вкладками» верхней панели. Центральные три меняют функцию в зависимости от активной сцены и действий пользователя и могут находиться в активированном и деактивированном состоянии. Основное назначение левой кнопки это возвращение на предыдущую сцену, отмена действия. Центральная соответствует основной функции сцены. Правая отвечает за добавление, создание и подобное.

Конкретно для начальной сцены назначение и визуальное представление кнопок следующее (соответственно их порядку слева-направо):

- просмотр сохранённых заказов в виде лупы на фоне списка;
- расчёт конструкции в виде калькулятора с выделенным знаком равенства;
- сохранение рассчитанной конструкции в виде «+», что появляется, следуя из названия, при удачном расчёте конструкции.

Сцена сохранения рассчитанной конструкции состоит из списка сохранённых заказов, каждый из которых представлен титульной информацией о заказе (адрес и название), количеством включенных в него конструкций разных типов, стоимостью конструкций и общей стоимостью заказа. При нажатии на один из элементов списка, рассчитанная конструкция будет добавлена к нему. Нижняя панель имеет активными левую и правую кнопки, отвечающие, соответственно за возврат к расчёту конструкции (в виде ломаной стрелки, указывающей налево) и сохранение рассчитанной конструкции как нового заказа с дальнейшим возвратом на предыдущую сцену.

Следующая сцена отвечает за управление сохранёнными заказами и конструкциями в них (Рисунок 5). Она представляет из себя список заказов,

аналогичный описанному выше. Длительное нажатие на элемент списка заказов вызывает диалог выбора действия:

- переход к сцене редактирования заказа;
- отправка информации о заказе в текстовом формате через стороннее приложение;
- удаление заказа.



Рисунок 5 - Сохранённые сделки

Выбор одного из элементов списка путём простого нажатия приводит к раскрытию его в новый список, состоящий из информации о входящих в заказ конструкциях; также на элемент списка заказов добавляется информация о дополнительных профилях, необходимых для данного заказа. Элемент списка, отображающий информацию о конструкции, содержит информацию о:

- типе конструкции (цифрой, соответствующей количеству створок, либо буквой «Г»);
- размере конструкции;
- размере заполнения для створки, либо конструкции;
- цене конструкции.

Нажатие на элемент конструкции вызывает информационное диалоговое окно, содержащее информацию о конструкции, аналогичную той, что выводится при первоначальном расчёте конструкции. Помимо закрытия диалога, доступна функция перехода к заказу (то же, что и редактирование в диалоге длительного нажатия). Нижняя панель схожа с панелью сцены сохранения рассчитанной конструкции, но правая кнопка отвечает за добавление нового заказа.

Сцена редактирования заказа (см. Рисунок 6) предоставляет наибольший набор функций для изменения параметров заказа. На ней доступно:

- удаление и добавление конструкций;
- удаление и добавление дополнительных профилей;
- изменение названия заказа;
- изменение адреса заказа.

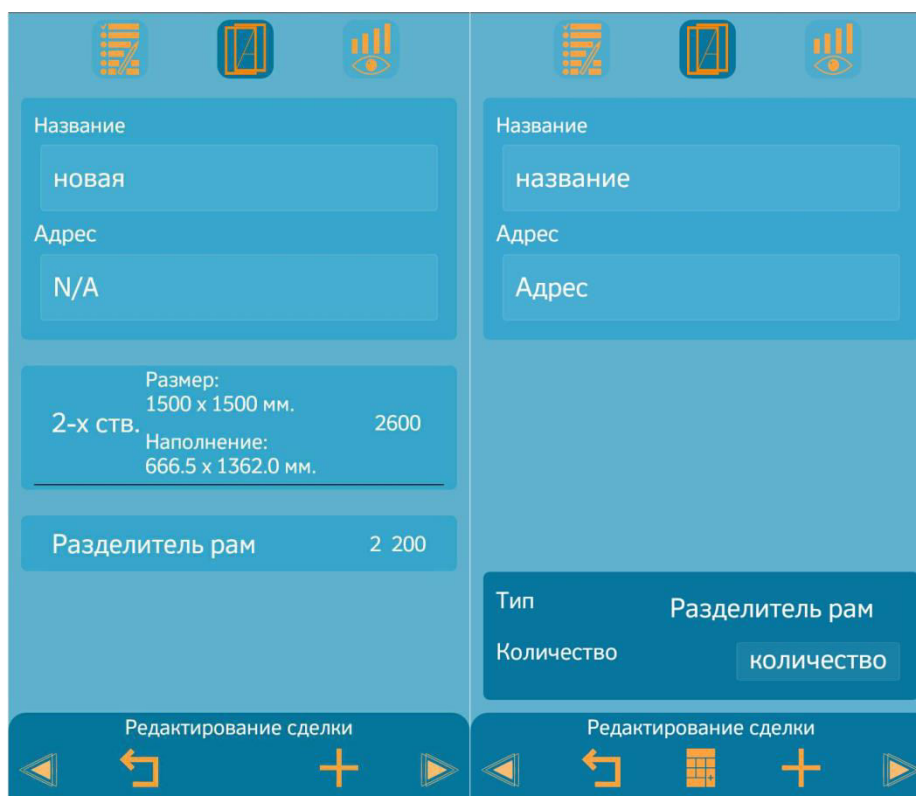


Рисунок 6 - Сцены создания и редактирования сделок

Для реализации описанного функционала, имеются поля для редактирования названия и адреса заказа, установленные по умолчанию на текущие значения соответствующих параметров, списки конструкций и

дополнительных профилей с возможностью удаления их элементов посредством выбора соответствующего действия через диалог, появляющийся при нажатии на элемент. Для добавления конструкций и профилей возможен вызов соответствующей панели добавления, который осуществляется посредством диалога, появляющемся при нажатии на правую кнопку нижней панели. Скрыть панель можно в том же диалоге. Для добавления конструкции вводится ширина, длина и тип, для дополнительных профилей - тип профиля и его количество. Левая кнопка нижней панели вызывает диалог выбора между сохранением (с дальнейшим выходом) и выходом без сохранения.

Следующая рассматриваемая сцена располагается в левой «вкладке», которая отвечает за управление текстовыми записями. Добавление функции ведения записей было введено в тестовом режиме, потому связанные сцены максимально просты по своей структуре.

Как и сцены, отвечающие за хранящиеся заказы, сцена просмотра записей состоит из списка сохранённых элементов, которые определяются названием и датой последнего сохранения. Записи с одинаковым названием и датой последнего сохранения перезаписываются. Нажатие на элемент списка вызывает диалог, предоставляющий выбор между опциям редактирования и удаления с подтверждением. Нижняя панель содержит одну активную кнопку с изображением знака «+» для создания новой записи, что переводит пользователя на сцену, аналогичную сцене редактирования записей.

Сцена редактирования записей в центральной части содержит одно текстовое поле для редактирования, куда, при открытии уже существующей записи, помещается её текст, в ином случае содержит подсказку (hint) «текст заметки». Содержание данного поля, соответственно, записывается в файл в том виде, в котором оно находилось на момент вызова функции сохранения, даже если не содержала ни единого символа. Нижняя панель имеет активными кнопки с ломаной стрелкой и дискетой, отвечающих за возврат и сохранение. Центральная кнопка с дискетой вызывает диалог, предлагающий задать

непустую строку в качестве названия записи (для новых записей), либо отредактировать имеющееся; после сохранения, возвращает на сцену просмотра хранимых на устройстве записей.

Последней из рассматриваемых будет сцена редактирования цен, расположенная в правой «вкладке», отвечающей за функции управления, но на данный момент с единственной уже обозначенной функцией. Так как на этапе обсуждения с заказчиком обсуждалась возможность управления заявками, оставленными посредством веб-сайта, через приложение для оперативного отклика, данная сцена состоит из раскрывающегося списка, где на данный момент находится элемент «Цены». Сам элемент состоит из вложенного списка из 5 элементов, где 4 отвечают за цены на типы конструкций с отдельными полями для редактирования цены конструкции, цены работы и коэффициента на конструкцию. Пятым является элемент для изменения цен на дополнительные профили, доступные к добавлению в заказы в данном приложении; сами цены указаны за штуку. Нижняя панель имеет одну активную кнопку с изображением дискеты, отвечающей за сохранение ценовых настроек, заданных на момент её нажатия. Информационное текстовое поле сообщает об успехе операции сохранения.

Верхняя панель, как уже было отмечено, включает в себя три элемента, обозначающие «вкладки», тематически разбитые наборы сцен: записи, конструкции и заказы, управление. Активная «вкладка» имеет более тёмный фон и более насыщенное изображение, привязанное к ней. Переключение между элементами данной панели происходит по выбору элемента, отличного от выбранного, либо по использованию навигационных кнопок на нижней панели, выбирающих элемент верхней панели по ту или иную сторону от активного, делая переход с правого на левый при «шаге» вправо, равно как и в обратной ситуации: с левого активного элемента на правый при «шаге» влево. При смене активного элемента, активная сцена сохраняется, чтобы при переключении на данный элемент в дальнейшем восстановить своё состояние и

последнюю активную сцену. Сцена сохранения рассчитанной конструкции не сохраняется, вместо неё происходит переключение на сцену расчёта конструкции с информацией о последней рассчитанной конструкции. По тому же принципу сцена редактирования записи не сохраняется, заменяемая на сцену просмотра сохранённых записей. При запуске приложения активной является центральная «вкладка» и базовая для неё сцена расчёта конструкции. Для левой «вкладки», согласно её тематике, базовой является сцена просмотра записей; для правой — сцена редактирования цен.

Для поддержания единства графического решения мобильного приложения, были созданы иконки для активных элементов на верхней и нижней панелях (см. Рисунок 7). Все изображения векторного формата (формат файла .svg), что позволяет масштабировать их напрямую, не храня несколько изображений одного и того же объекта для разных разрешений экрана. В самом же мобильном приложении используются преобразованные из обозначенных файлы формата xml в форматировании, предназначенном для использования в Android проектах.

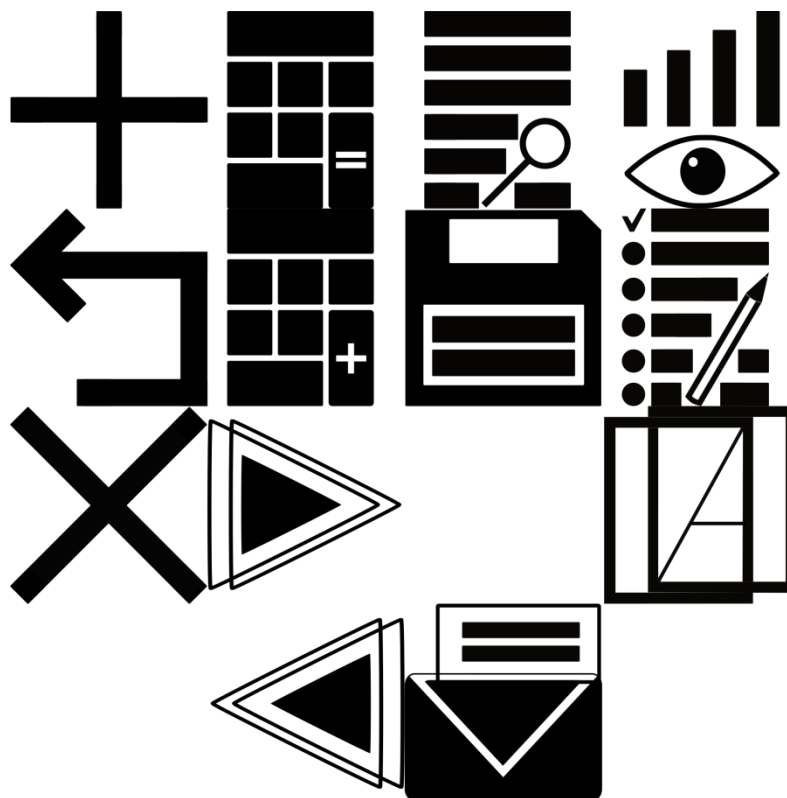


Рисунок 7 - Набор разработанных иконок

Среди всех разработанных иконок можно выделить три стилистических группы, критерием разделения которых является их роль в приложении:

- элементы верхней панели;
- функциональные элементы нижней панели;
- навигационные элементы нижней панели.

Элементы верхней панели остаются на своих местах и являются наиболее крупными из всех. Они должны быть хорошо различимы и воспринимаемы, для чего при их создании делался акцент на свободном пространстве между составляющими иконки, а также использовании декоративных элементов для уточнения функционала группы, в которую ведёт каждая конкретная иконка на верхней панели. Например, иконка для центральной группы изображает две створки, внутри которых имеется две прямых; прямые эти символизируют замеры и расчёты, производимые в этой группе.

Функциональные элементы нижней панели — наиболее многочисленная группа из обозначенных, ибо в неё входят иконки для всех контекстуальных действий, что меняются в зависимости от текущей сцены. Их дизайн подчинён максимальной информативности и узнаваемости: лупа для обозначения просмотра, знак равенства для расчёта, плюс для создания и так далее. Также для данной группы характерна высокая плотность элементов, либо их размер (для иконки «возврата» и «добавления»).

Последняя группа состоит из двух иконок, составленных из одного отзеркаленного изображения. Служат они для навигации между «закладками» вправо и влево. Имея стандартный размер квадрата, в который они вписаны, иконки навигации визуально выглядят вытянутыми, что тематически отличает их от прочих, подсказывая действие, ими олицетворяемое.

Цветовое решение отправной точкой имело оранжевый цвет, как неформальный цвет города, в котором работает заказчик. Чтобы не перегружать интерфейс множеством цветов, но при этом сохранить

возможность установки визуальных акцентов, было решено выбрать контрастное соотношение. Таким образом, получен был набор оттенков оранжевого и синего. Как более яркий и привлекающий внимание, оранжевый был определён цветом интерактивных элементов. При нажатии (нижняя панель) или нахождении в неактивном состоянии (верхняя панель) оттенки самой иконки и её подложки меняют оттенок или инвертируют цвет по выбранной цветовой схеме (оттенки оранжевого и синего были проиндексированы попарно от светлых к тёмным), чтобы обозначить изменённое состояние. Большая часть каждой сцены состоит из элементов синих оттенков, текст установлен белым, как нейтральный и не напрягающий зрение. Акценты делаются через более тёмные оттенки фона элементов при повышении его значимости.

2.3 Итоги апробации

Как только итоговая версия приложения, содержащая реализацию всех функций, обозначенных в техническом задании, и инструкция пользователя были введены в бизнес-процесс компании, стало возможным выделение основных аспектов, которые значимы для подведения итогов разработки мобильного приложения, его интеграции в бизнес-процесс компании, перспектив модернизации [Приложение 1, Приложение 2]. Как уже было сказано, оценку приложения пользователем (заказчиком) было решено провести в формате беседы.

Для проведения беседы о результатах включения мобильного приложения в бизнес-процесс компании были составлены вопросы, затрагивающие основные элементы разработанного продукта, нацеленные на создание структуры беседы. Среди них есть вопросы о раннем вводе приложения в эксплуатацию, оценке структурного и визуального дизайна пользователем, работе с основными функциями, работе с экспериментальными функциями. Таким образом, получился следующий список вопросов:

1. Постепенный ввод в эксплуатацию создавал неудобства, связанные с функционированием приложения?
2. Является ли выбранная цветовая гамма подходящей для данного приложения?
3. Удобна ли выбранная модель навигации, разделение по «вкладкам»?
4. Приложение работает корректно с точки зрения расчётов и представления информации?
5. Оцените удобство пользования основной функцией расчётов конструкций. Какие есть предложения или замечания по её реализации?
6. Как часто использовалась тестовая функция работы с текстовыми заметками? Кажется ли она перспективной в рамках данного приложения?
7. Как часто использовалась функция хранения заказов? Необходимы ли изменения в её представлении (информационном, графическом) в приложении?
8. Предоставленное руководство пользователя проясняет особенности работы с приложением, в достаточной мере рассказывает о реализованных в нём функциях?
9. Имеются ли общие пожелания и замечания по работе приложения, выбранному способу его интеграции?

После проведения беседы, полученная информация была обобщена и проанализирована с точки зрения возможных рекомендаций. По итогам, выбранный способ ввода в эксплуатацию заказчиком был признан удачным, редкие неудобства были связаны с мелкими недочётами в программном коде, не влекущими за собой неработоспособность основных функций. Решение дизайна было названо адекватным, цвета не мешающими работе с приложением, разделение на «вкладки» понятным. Расчёты проводятся корректно, информация выводится в нужном виде. Тестовая функция не использовалась после её реализации, заказчик отмечает, что ведение текстовых заметок перспективно, но, на данный момент, необходимости в электронном

варианте классических текстовых записей нет. Функция хранения заказов также была используется недостаточно для вывода решения об эффективности её реализации, однако отмечена значимость подобной возможности. Руководство заказчик, предоставленное после передачи конечной версии приложения, характеризуется заказчиком как предоставляющее необходимую информацию о функционировании приложения и основах работы с ним.

На основе данных, полученных в ходе беседы с конечным пользователем (заказчиком) и собственного анализа проделанной работы, были подведены итоги об успешности выполнения поставленных задач. Разработанное мобильное приложение выполняет все заявленные функции, имеет целостный дизайн и навигацию, не препятствующую использованию всех функций приложения. Ввод в эксплуатацию проведён успешно с незначительными затруднениями, предусмотренными выбранным методом интеграции продукта в информационную систему компании. Однако позднее предоставление цельного руководства пользователя затруднило своевременное ознакомление пользователя со всем функционалом. Так как функционал добавлялся по ходу выпуска новых «малых» версий, то стоит рекомендовать в дальнейшем прилагать записки в текстовой форме, словесно описывающие новые возможности по мере их реализации, что как не будет оказывать заметного негативного влияния на загруженность разработчика, так и полезно для обучения конечного пользователя и тестирования нововведений.

Также были сформулированы возможные направления по модернизации и преобразованию разработанного мобильного приложения, в случае возникновения такой необходимости:

- реструктуризация графического интерфейса;
- добавление управленческих функций;
- изменение тестовой функции заметок.

Под изменением функции заметок подразумевается создание шаблонизированных действий, например, установка напоминаний от

приложения о важных событиях или контактов клиентов. Подобная рекомендация выходит из низкой частоты пользования заметками на текущем этапе развития данной функции. Отметим также, что решение о добавлении возможности хранения списка контактов повлечёт за собой требование о защите персональных данных.

Дополнительная защита конфиденциальных и деловых данных является требованием и к рекомендации по добавлению управленческих функций. В итоговой версии продукта к управлению относилась возможность регулировки цен на составные элементы, однако, при дальнейшем расширении информационной системы компании возможно подключение облачных и клиент-серверных технологий для подачи и подтверждения заявок на услуги посредством интернета.

Согласно частоте использования, можно утверждать, что наиболее значимыми возможностями приложения являются расчёт размеров и стоимости, а также хранение заказов. По этой причине и в случае реализации вышесказанных рекомендаций, выводится перспектива реструктуризации графического интерфейса. Центральным самостоятельным элементом должна остаться сцена расчёта размеров и стоимости, сохраняя последовательность действий от этой сцены, позволяющая добавить рассчитанную конструкцию к хранимым заказам. Однако управление хранимыми заказами стоит выделить в самостоятельную структурную группу, возможно, объединив с текстовыми заметками. Если же значительного расширения управленческих функций не произойдёт, то возможно объединение доработанных заметок с управленческой группой. Таким образом, будет поставлен акцент на управлении хранимыми данными, как на отдельной возможности, не связанной исключительно с расчётом конструкций; в то же время, это может облегчить взаимодействие с основной сценой.

Таким образом, на основе проведённой беседы с заказчиком и анализа процессов разработки и внедрения мобильного приложения были сделаны

выводы об успешности выполнения поставленных задач и выведены рекомендации по возможным способам модернизации мобильного приложения.

Заключение

Целью данной работы было создание мобильного приложения для информационной системы ООО «Элефант»: проектирование, разработка и описание его. Для её достижения были выполнены поставленные задачи.

Была проанализирована предметная область компании, выделены объекты, значимые для реализации указанных в техническом задании функций, и спроецированы на формулы расчёта компании в рамках программного кода мобильного приложения для операционной системы Android. Был разработан функциональный и графический дизайн мобильного приложения, элементы, а также входящие в него элементы.

После решения задач проектирования было разработано мобильное приложение, реализующее функции расчёта размеров и стоимости балконных конструкций, согласно формулам расчёта компании-заказчика. Проведён ввод указанного приложения в эксплуатацию.

По результатам разработки и внедрения продукта в бизнес-процесс компании было проведено оценивание успешности достижения поставленных целей с помощью беседы с заказчиком. Из неё следует, что поставленные цели были достигнуты, продукт соответствует техническому заданию, требованиям и ожиданиям заказчика.

Также было разработано и передано в пользование заказчику руководство пользователя, были сформулированы рекомендации по возможным способам модернизации разработанного продукта.

Таким образом, в ходе данной работы было спроектировано, разработано и введено в эксплуатацию мобильное приложение, отвечающее всем требованиям, сформулированным в техническом задании.

Список информационных источников

1. Аксенов К. В. Обзор современных средств для разработки мобильных приложений // Новые информационные технологии в автоматизированных системах. 2014. №17. URL: <https://cyberleninka.ru/article/n/obzor-sovremennyh-sredstv-dlya-razrabotki-mobilnyh-prilozheniy> (дата обращения: 27.04.2019).
2. Балдина Е. А., Геворгян Н. Р., Ткалич С. К. "Незаметный" дизайн мобильного приложения // Научные исследования: теория, методика и практика. Чебоксары: Центр научного сотрудничества "Интерактив плюс", 2017.
3. Боковой Юрий Владимирович Особенности методологии проектирования информационных систем для малого и среднего бизнеса // Прикладная информатика. 2006. №5. URL: <https://cyberleninka.ru/article/n/osobennosti-metodologii-proektirovaniya-informatsionnyh-sistem-dlya-malogo-i-srednego-biznesa> (дата обращения: 28.04.2019).
4. Бужинская Н. В., Стародумова Д. С. Проектирование мобильных приложений для смартфонов на базе операционной системы Android // Современная техника и технологии. 2017. №4.
5. Глушак Н. В. Методика оценки эффективности внедрения информационной системы для реализации задач контроллинга // Известия РГПУ им. А.И. Герцена. 2008. №58. URL: <https://cyberleninka.ru/article/n/metodika-otsenki-effektivnosti-vnedreniya-informatsionnoy-sistemy-dlya-realizatsii-zadach-kontrollinga> (дата обращения: 28.04.2019).
6. ГОСТ 34.602-89 Межгосударственный стандарт "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы" от 24.03.89 №661 // Государственным комитетом СССР по стандартам. 1989 г.
7. Достоинства и недостатки Xamarin // habr URL: <https://habr.com/ru/company/microsoft/blog/415833/> (дата обращения: 27.04.2019).

8. Елисафенко Михаил Евгеньевич, Васючкова Татьяна Сергеевна Особенности разработки систем управления предприятием для малого бизнеса // Universum: технические науки. 2015. №3 (16). URL: <https://cyberleninka.ru/article/n/osobennosti-razrabotki-sistem-upravleniya-predpriyatiem-dlya-malogo-biznesa> (дата обращения: 28.04.2019).
9. Ещё раз про семь основных методологий разработки // habr URL: <https://habr.com/ru/company/edison/blog/269789/> (дата обращения: 27.04.2019).
10. Ильина Н. И. Целесообразность внедрения мобильного приложения на предприятии // Научные исследования. 2017. №4 (15). URL: <https://cyberleninka.ru/article/n/tselesoobraznost-vnedreniya-mobilnogo-prilozheniya-na-predpriyatii> (дата обращения: 27.04.2019).
11. Инструменты Visual Studio для Xamarin // Microsoft URL: <https://visualstudio.microsoft.com/ru/xamarin/?rr=https%3A%2F%2Fwww.google.com%2F> (дата обращения: 27.04.2019).
12. Итоги 2017 года в мобайле. Чего ожидать от 2018? Отчёт Unilead News // Unilead News URL: <https://news.unilead.net/2017/12/19/itogi-2017-v-mobajle/> (дата обращения: 27.04.2019).
13. Казакова А. Н. Исследование способов хранения информации в Android приложениях // Материалы конференций ГНИИ "Нацразвитие". Декабрь 2017. М.: ГНИИ "Нацразвитие", 2018.
14. Какие языки программирования нужно знать, чтобы разрабатывать приложения под Android? // Medium URL: <https://medium.com/nuances-of-programming/какие-языки-программирования-нужно-знать-чтобы-разрабатывать-приложения-под-android-317b05fe1214> (дата обращения: 27.04.2019).
15. Ким В. Ю. Особенности разработки дизайна пользовательского интерфейса для мобильного приложения // Новые информационные технологии в автоматизированных системах. 2015. №18. URL:

- <https://cyberleninka.ru/article/n/osobennosti-razrabotki-dizayna-polzovatelskogo-interfeysa-dlya-mobilnogo-prilozheniya> (дата обращения: 27.04.2019).
16. Костюк Л. В., Марченко М. Н. Теоретические основы проектирования дизайна интерфейсов мобильных приложений // Дизайн-образование: проблемы и перспективы. Краснодар: Кубанский государственный университет, 2016.
 17. Ленкин А. В. Обзор языка программирования Kotlin // Постулат. 2018. №1.
 18. Методы проектирования БД // Студопедия URL: <https://studopedia.org/2-71693.html> (дата обращения: 27.04.2019).
 19. Мобильная разработка на Python: обзор двух фреймворков // proglib URL: <https://proglib.io/p/mobile-python/> (дата обращения: 27.04.2019).
 20. НГУЕН Б. Х., НГУЕН Д. Ч., НГУЕН Н. Х., ЛАЙ Т. Л., НГУЕН Ч. Л., БУЙ М. Д. Виды мобильных приложений // Синергия наук. 2017. №7.
 21. Обоснование создания мобильного приложения // Studbooks URL: <https://studbooks.net/1174177/informatika/obosnovanie-sozdaniya-mobilnogo-prilozheniya> (дата обращения: 27.04.2019).
 22. Образцова В.С., Долгова Т.Г. Мобильные приложения для бизнеса // Актуальные проблемы авиации и космонавтики. 2014. №10. URL: <https://cyberleninka.ru/article/n/mobilnye-prilozheniya-dlya-biznesa> (дата обращения: 27.04.2019).
 23. Операционные системы для мобильных устройств // StudFiles URL: <https://studfiles.net/preview/3053772/> (дата обращения: 27.04.2019).
 24. C++ в Android. Часть 1 - Введение // SPARK URL: <https://spark.ru/startup/mobiledev/blog/33292/s-v-android-chast-1-vvedenie> (дата обращения: 27.04.2019).
 25. Сидора А. А. Способы хранения данных в приложениях Android OS // Решетневские чтения. 2015. №19. URL: <https://cyberleninka.ru/article/n/sposoby-hraneniya-dannyh-v-prilozheniyah-android-os> (дата обращения: 27.04.2019).

- 26.Созданный россиянами язык Kotlin получил поддержку на Android // Ведомости URL: <https://www.vedomosti.ru/technology/articles/2017/05/18/690452-kotlin-podderzhku-android> (дата обращения: 27.04.2019).
- 27.Технические характеристики алюм. профиля "PROVEDAL C640" // Praktika URL: http://www.profti.ru/razdv_ostekl/xap_provedal_c640.html (дата обращения: 28.04.2019).
- 28.Технические характеристики алюм. профиля "PROVEDAL P400" // Praktika URL: http://www.profti.ru/razdv_ostekl/xap_provedal_p400.html (дата обращения: 28.04.2019).
- 29.Ургенчев А. Р., Байбактина А. Т. Анализ современных средств для разработки мобильных приложений под ОС Android // Научный журнал. 2017. №3.
- 30.Целых А. Н., Целых Лариса Анатольевна Мобильные приложения для информационных систем поддержки принятия решений в малом бизнесе // Известия ЮФУ. Технические науки. 2013. №11 (148). URL: <https://cyberleninka.ru/article/n/mobilnye-prilozheniya-dlya-informatsionnyh-sistem-podderzhki-prinyatiya-resheniy-v-malom-biznese> (дата обращения: 27.04.2019).
- 31.Шполянская Ирина Юрьевна Информационные системы для малого бизнеса: проблема качества // Вестник РГЭУ РИНХ. 2006. №22. URL: <https://cyberleninka.ru/article/n/informatsionnye-sistemy-dlya-malogo-biznesa-problema-kachestva> (дата обращения: 28.04.2019).
- 32.Эренчин, Т.Р. Технологии баз данных в мобильном приложении : выпускная квалификационная работа / Т.Р. Эренчин ; Тувинский Государственный Университет, Кафедра информатики. - Кызыл : , 2017. - 56 с. : табл., ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=492830>(27.04.2019).

- 33.Add C and C++ code to Your project // Android Developers URL:
<https://developer.android.com/studio/projects/add-native-code> (дата обращения:
27.04.2019).
- 34.Add multi-density vector graphics // Android Developer URL:
<https://developer.android.com/studio/write/vector-asset-studio#svg> (дата
обращения: 27.04.2019).
- 35.B4X URL: <https://www.b4x.com> (дата обращения: 27.04.2019).
- 36.Data Classes // Kotlin URL: <https://kotlinlang.org/docs/reference/data-classes.html> (дата обращения: 27.04.2019).
- 37.Enum Classes // Kotlin URL: <https://kotlinlang.org/docs/reference/enum-classes.html> (дата обращения: 27.04.2019).
- 38.GitHub URL: <https://github.com/kivy/python-for-android> (дата обращения:
27.04.2019).
- 39.Kotlin URL: <https://kotlinlang.org> (дата обращения: 27.04.2019).
- 40.Null Safety // Kotlin URL: <https://kotlinlang.org/docs/reference/null-safety.html>
(дата обращения: 27.04.2019).
- 41.SCALABLE VECTOR GRAPHICS // W3C URL:
<https://www.w3.org/Graphics/SVG/> (дата обращения: 27.04.2019).
- 42.SQLite URL: <https://www.sqlite.org/docs.html> (дата обращения: 27.04.2019).
- 43.The Java Virtual Machine Specification // Oracle URL:
<https://docs.oracle.com/javase/specs/jvms/se8/html/> (дата обращения:
27.04.2019).

Приложения

Приложение 1.

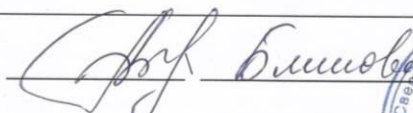
Акт приема–передачи программного обеспечения

«01» февраля 2019 г.

Блинова Наталия Леонидовна, в лице директора ООО «Элефант», действующего на основании устава и **Арасланова Егора Руслановича** составили настоящий акт о нижеследующем:

1. **Арасланов Егор Русланович** передал, а **Блинова Наталия Леонидовна** приняла программное мобильное приложение “Elephant Next” в полной комплектации и стороны претензий к друг другу не имеют.

Директор ООО «Элефант» _____




Арасланов Егор Русланович _____

 \ Арасланов Е.Р.

М.П.

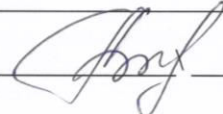
Акт согласования технического задания

«17»июля 2018 г.

Блинова Наталия Леонидовна, в лице директора ООО «Элефант», действующего на основании устава и **Арасланова Егора Руслановича** составили настоящий акт о нижеследующем:

1. Арасланов Егор Русланович согласовал с **Блиновой Наталией Леонидовной** содержание технического задания на разработку автоматизированной системы: мобильное приложение “Elephant Next”.

Директор ООО «Элефант»




Арасланов Егор Русланович

 \ Арасланов Е.Р.

М.П.