Содержание

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Назначение и цели разработки | 4 |
| 2 Разработка технического проекта на основе анализа требований | 5 |
| 2.1 Определение спецификаций программного обеспечения | 5 |
| 2.2 Проектирование модели данных и диаграммы классов | 9 |
| 2.3 Конструирование прототипа | 12 |
| 3 Реализация | 17 |
| 3.1 Обоснование выбора средств разработки | 17 |
| 3.2 Описание программных модулей | 18 |
| 3.3 Описание разработанного приложения | 21 |
| 4 Тестирование программных модулей | 28 |
| 4.1 Интеграционное тестирование | 28 |
| 5 Разработка эксплуатационной документации | 30 |
| 5.1 Руководство пользователя | 30 |
| 5.2 Руководство программиста | 31 |
| Заключение | 32 |
| Список использованных источников | 33 |
| Приложение А Техническое задание. Требования к программным модулям | 35 |
| Приложение Б Программный код | 41 |
| Приложение В Тестирование | 42 |

Введение

В настоящее время, в мире непрерывного потока информации, все большую актуальность приобретают процессы автоматизации работ рутинного характера и работ, трудно поддающихся формализации. В связи с этим ежегодно разрабатываются автоматизированные информационные системы, призванные решить возникающие проблемы. С увеличением доступности мобильного интернета все большей популярностью стали пользоваться мобильные приложения.

Компания ООО «Зодиак-Электро» монтажными работами. Одной из важнейших проблем, существующих в данной компании, является автоматизация сотрудничества менеджеров и специалистов по монтажу. Менеджер должен выдавать задачи специалисту, а специалист в свою очередь оперативно получать и выполнять назначенные задачи, сопровождая их фотоотчетом.

Выпускная квалификационная работа включает в себя проектирование и разработку локальной базы данных и мобильного приложения для операционной системы Android в рамках информационной системы FixPhoto, предназначенной для автоматизации работы менеджеров и специалистов по монтажу.

1 Назначение и цели разработки

Техническим заданием предусмотрена разработка мобильного приложения для информационной системы FixPhoto. Разработанное приложение предназначено для автоматизации работы специалиста по монтажу в ООО «Зодиак-Электро», а также для сбора, обработки и хранения информации о выполняемых пользователем задачах.

Автоматизация позволит получить следующие преимущества по сравнению с нынешней реализацией данных задач:

- быстрый сбор статистики по выполненным задачам за выбранный период;

- оперативное выполнение назначенных задач на специалиста по монтажу;

- ускорение процесса работы специалистов по монтажу, благодаря удобству фиксации выполненной работы.

Данное приложение, интегрированное с системой FixPhoto, позволит максимально сократить присутствие человеческого фактора в ряде важных задач, таких как фиксация выполненной работы, выполнение назначенных задач. Помимо этого, система в целом полностью автоматизирует сотрудничество между менеджерами и специалистами по монтажу, а также формирует статистику и отчетность по выполненным задачам.

2 Разработка технического проекта на основе анализа требований

2.1 Определение спецификаций программного обеспечения

Рассмотрим определение вариантов использования (прецедентов).

Мобильное приложение для системы FixPhoto предназначено для специалистов по монтажу. Соответственно основной прецедент (вариант использования) для разрабатываемого приложение будет – выполнение задачи.

Диаграмма вариантов использования, созданная средствами MS Visio, для проектируемой системы представлена на рисунке 1.

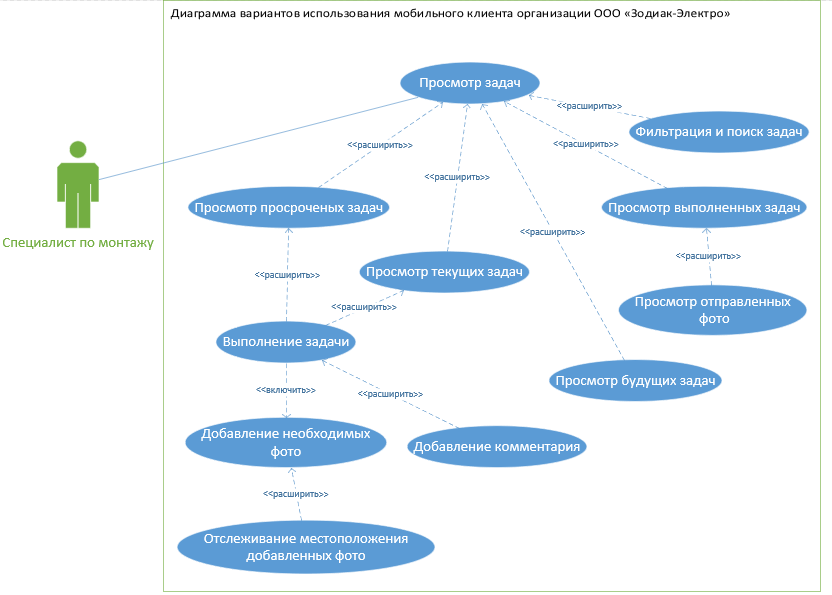


Рисунок 1 – Диаграмма вариантов использования мобильного приложения FixPhoto

В таблице 1 представлено описание главного раздела сценария прецедента (варианта использования) «Выполнение задачи».

Таблица 1 - Главный раздел сценария «Выполнение задачи»

|  |  |
| --- | --- |
| Вариант использования | Выполнение задачи |
| Актеры | Специалист по монтажу |
| Краткое описание | Специалист по монтажу делает необходимое количество фотографий к задаче, добавляет комментарий при необходимости и выполняет задачу |
| Цель | Выполнить назначенную задачу |
| Тип | Базовый |
| Ссылки на другие варианты использования | Включает в себя добавление фотографий и комментария |

В таблице 2 описана последовательность действий специалиста по монтажу, приводящая к успешному выполнению прецедента (варианта использования) «Выполнение задачи»

Таблица 2 - Сценарий успешного выполнения варианта использования «Выполнение задачи»

|  |  |
| --- | --- |
| Действия актеров | Отклик приложения |
| 3. Выбор задачи из списка | 1. Получение списка задач, назначенных на пользователя.  2. Вывод задач на экран  4. Открытие нового экрана выбранной задачи с подробным описанием |
| 5. Добавление фотографий к задаче и нажатие кнопки «Выполнить»  Исключение 1. Добавлено недостаточное количество фотографий | 6. Приложение сжимает фотографии и отправляет выполненную задачу на сервер |

В таблице 3 представлен сценарий обработки исключительных ситуаций для варианта использования «Выполнение задачи».

Таблица 3 - Обработка исключительных ситуаций для варианта использования «Выполнение задачи»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| Исключение 1. Добавлено недостаточное количество фотографий | |
|  | 6. Приложение отображает всплывающее сообщение о необходимости добавления дополнительных фотографий |

С помощью диаграммы деятельности, представленной на рисунке 2, описан алгоритм реализации описанных сценариев для варианта использования «Выполнение задачи».

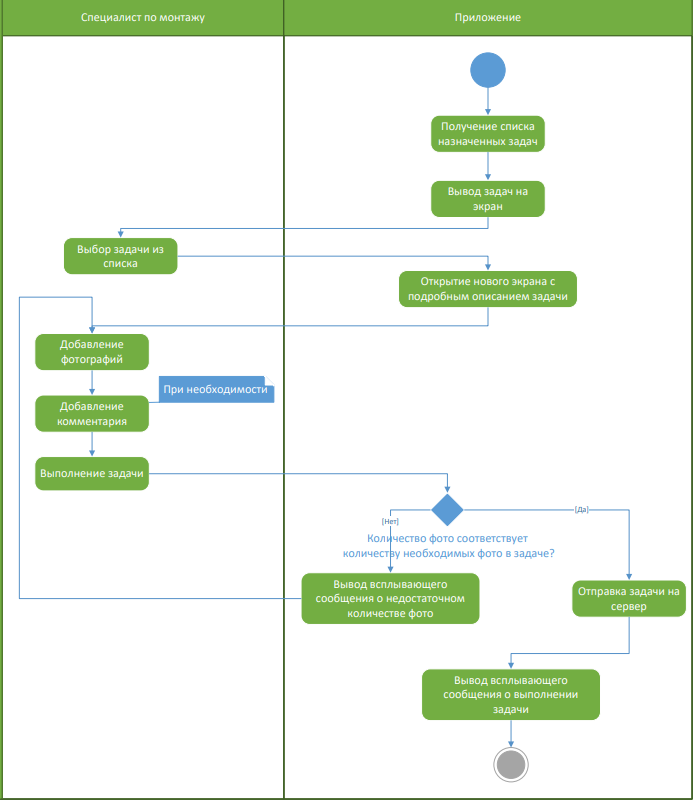


Рисунок 2 – Диаграмма деятельности для варианта использования «Выполнение задачи»

На диаграмме последовательностей, представленной на рисунке 3, описаны требования реализации сценария «Выполнение задачи» с учетом динамики во времени на уровне сообщений.

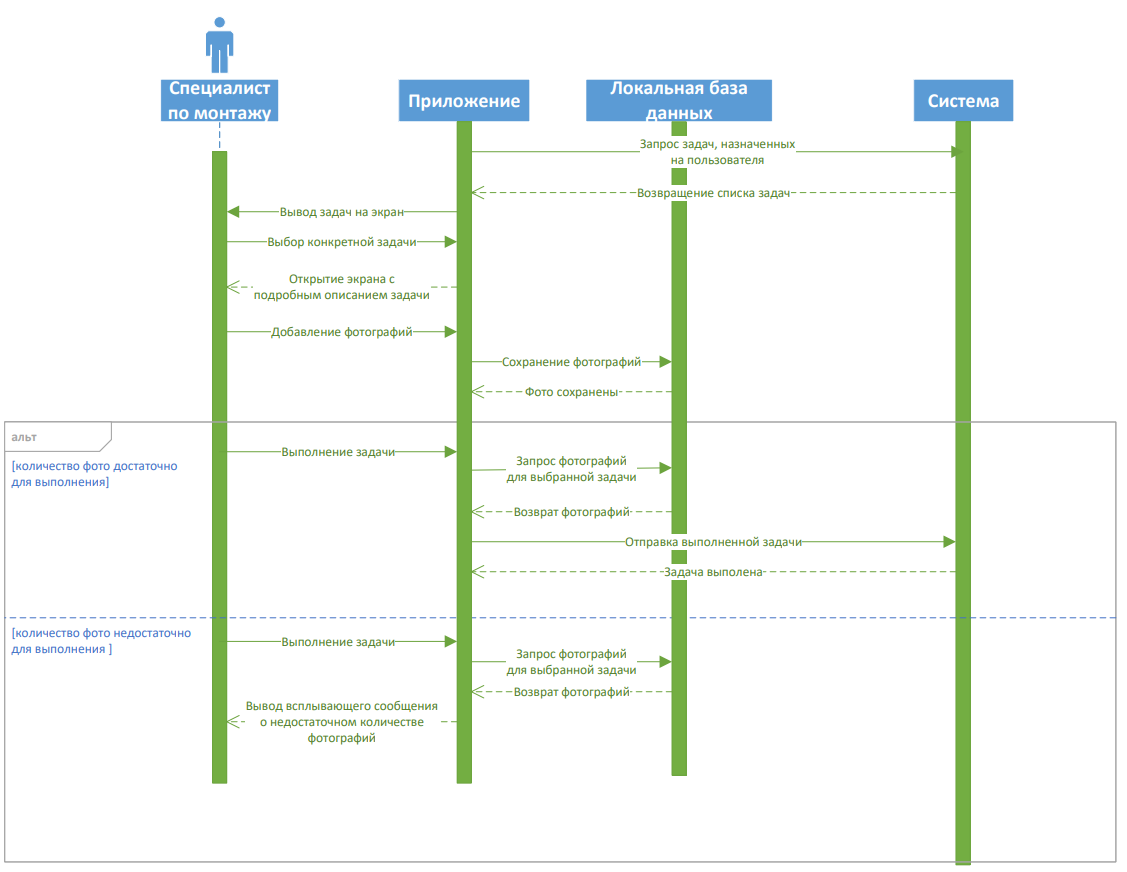


Рисунок 3 - Диаграмма последовательности для варианта использования «Выполнение задачи»

Разработанные спецификации программного обеспечения на языке UML средствами MS Visio показывают основные действия пользователя мобильного приложения и алгоритмы их выполнения.

2.2 Проектирование модели данных и диаграммы классов

На основе анализа требований заказчика к функциям мобильного приложения, требованиям к организации входных и выходных данных, с учетом спроектированных требований к реализации функций, описанных в диаграммах вариантов использования, деятельности и последовательности, была разработана модель данных приложения, которая будет использоваться локально, и описана в виде ER-модели, позволяющая четко описать требования к представлению логической структуры данных, на основе которой в последующем будет разработана физическая структура данных для хранения во внутренней памяти. Так как мобильное приложение будет ориентировано на интеграцию с системой и работать непосредственно через интерфейс программирования приложений, расположенном на сервере, где хранится основная база данных, то локально необходимо будет хранить только фотографии, а также их дату создания и местоположение, еще не выполненных задач, то локальная база данных будет состоять из 1 таблицы.

На рисунке 4 представлена ER–модель для локальной базы данных мобильного приложения.

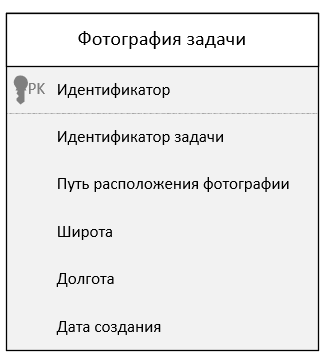


Рисунок 4 – ER-модель для локальной базы данных мобильного приложения

В таблице 4 представлены сведения о сущности ER-модели из рисунка 4.

Таблица 4 – Характеристики сущности ER-модели для локальной базы данных мобильного приложения

|  |  |
| --- | --- |
| Атрибут | Значение |
| Идентификатор | Первичный ключ |
| Идентификатор задачи | Идентификатор задачи, полученной с сервера |
| Путь расположения фотографии | Содержит информацию о местоположении файла в системе Android |
| Широта | Широта местоположения сделанной фотографии |
| Долгота | Долгота местоположения сделанной фотографии |
| Дата создания | Дата и время создания фотографии |

На следующем этапе была спроектирована диаграмма классов, которая описывает требования к внутренней организации проектируемого программного обеспечения на уровне интерфейсов и классов, их кооперации, атрибутов и методов классов.

На рисунке 5 представлена диаграмма классов для локальной базы данных приложения.

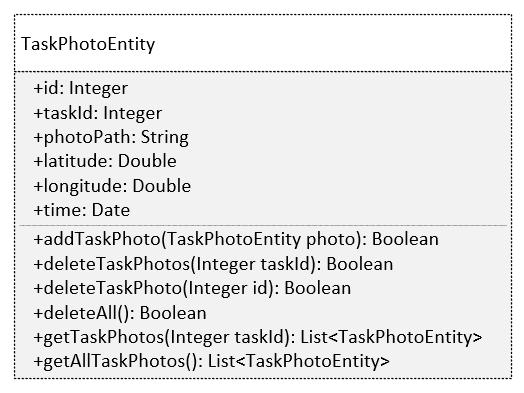


Рисунок 5 –Диаграмма классов для локальной базы данных приложения

Данная диаграмма классов описывает объект TaskPhotoEntity локальной базы данных, а также отображаются операции класса.

В таблице 5 представлены сведения о структуре сущности, свойствах, а также её методах.

Таблица 5 - Характеристика диаграммы классов для локальной базы данных

|  |  |  |
| --- | --- | --- |
| Имя класса | Атрибуты класса | Методы класса |
| TaskPhotoEntity | +id – содержит целочисленный код фотографии;  +taskId – содержит целочисленный код задачи, получаемой из системы;  +photoPath – содержит путь расположения фотографии в операционной системе Android;  +latitude – содержит широту местоположения сделанной фотографии;  +longitude – содержит долготу местоположения сделанной фотографии  +time – содержит дату и время создания фотографии | +addTaskPhoto(TaskPhotoEntity photo) – добавить фотографию к задаче в локальную базу данных.  + deleteTaskPhotos(Integer taskId) – удаляет все фотографии к определенной задачи. Вызывается при отправке задачи на сервер.  + deleteTaskPhoto(Integer id) – удаляет определенную фотографию из локальной базы данных.  + deleteAll() – удаляет все записи из локальной базы данных. Вызывается при выходе из учетной записи.  + getTaskPhotos(Integer taskId) – возвращает список фотографий к определенной задаче.  + getAllTaskPhotos() – возвращает весь список фотографий из локальной базы данных ко всем задачам. |

Данная диаграмма классов и ее описание служат для представления статической структуры модели локальной базы данных приложения.

2.3 Конструирование прототипа

Для построения прототипа первоначально были составлены требования к макету приложения.

Все компоненты должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

* разметка и дизайн (разметка должна быть масштабируема, так как устройства могут различаться размером дисплея);
* группировка элементов (в логические категории);
* использование соответствующих элементов управления;
* расположение и выравнивание элементов;
* общая компоновка логична, понятна и проста в использовании;
* последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении;
* соответствующий заголовок на каждом окне приложения.

Основные требования руководства по стилю:

* для приложения должна быть установлена иконка, полученная от заказчика, изображенная на рисунке 6;
* тип шрифта – Roboto;
* цветовая схема предусматривает использование в качестве основного фона – белый цвет RGB (255, 255, 225), в качестве дополнительных – синий цвет RGB (0, 82, 204) и красный цвет RGB(255, 0, 0) для акцентирования внимания пользователя.



Рисунок 6 – Логотип приложения

С учетом требований к макету и руководству по стилю, для обеспечения требуемых функций, были разработаны макеты экранов приложения, в последствии утвержденные заказчиком. Для разработки макетов была использована Figma – онлайн-сервис для дизайнеров, разработчиков и маркетологов, предназначенный для создания прототипов сайтов или приложений, иллюстраций и векторной графики.

На рисунке 7 представлен прототип экрана авторизации для пользователей.

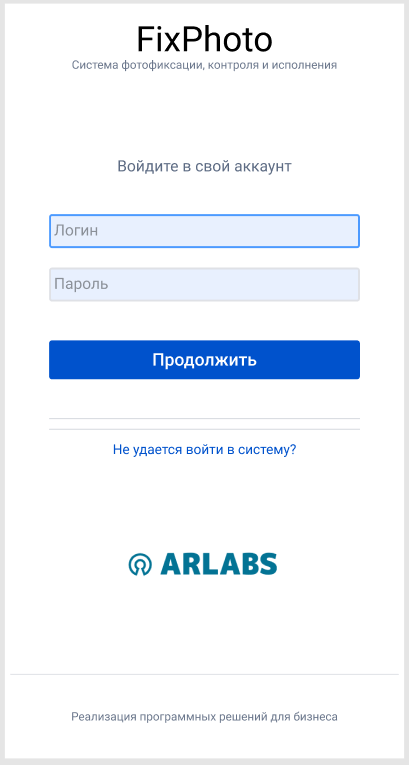


Рисунок 7 – Прототип экрана авторизации для пользователей

На рисунке 8 представлен прототип главного экрана, с которым пользователь будет работать и видеть назначенные на него задачи. На экране находится меню, расположенное внизу, и основной контент в виде списка задач и строки поиска.

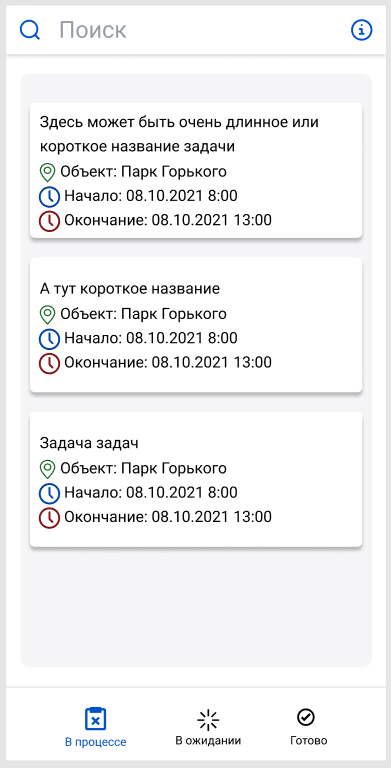


Рисунок 8 – Прототип главного экрана

При нажатии на какую-то задачу открывается новый экран с подробным описанием выбранной задачи. Прототип экрана с подробным описанием задачи представлен на рисунке 9.

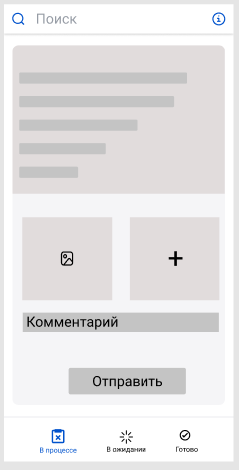


Рисунок 9 – Прототип экрана с описанием задачи

При нажатии на кнопку «+» открывается новый экран с изображением с основной камеры. В правом верхнем углу отображается количество сделанных фотографий. Внизу находятся три кнопки: кнопка для включения/отключения вспышки, кнопка для создания фотографии, кнопка для прикрепления фотографий к задаче. Данный экран представлен на рисунке 10.



Рисунок 10 – Прототип экрана для создания фотографий

Данные прототипы основных экранов отображают главный функционал для специалистов по монтажу, а именно просмотр назначенных задач, добавление и удаление фотографий и комментария, а также выполнение задачи. Эти макеты были утверждены заказчиком перед началом разработки.

3 Реализация

3.1 Обоснование выбора средств разработки

В таблице 6 представлены некоторые возможные средства разработки для разработки мобильных приложений.

Таблица 6 – Возможные средства разработки

|  |  |  |
| --- | --- | --- |
| Технология | Плюсы | Минусы |
| Flutter (язык Dart) | Кроссплатформенность, интерфейс легко разбивается на отдельные модули, собственный графический движок | Нестабильность, библиотек меньше, чем для базовой разработки, интерфейс создается с помощью кода |
| Xamarin (язык C#) | Производительность близка к нативной, простая поддержка, полный пакет инструментов разработки | Задержки с обновлениями платформ, необходимость писать нативный код в определенных ситуациях |
| Нативная разработка на Java | Большая часть Android приложений написана на Java, большое количество библиотек и официальной документации | Более низкая производительность при работе с большими объемами информации |

Так как мобильное приложение ориентировано на операционную систему Android, то необходимость в кроссплатформенности отсутствует, поэтому выбирать Flutter не стоит, так как он нестабилен и не имеет достаточное количество библиотек. Xamarin не подходит, так как требует написание нативного кода для некоторых модулей, например, модуль работы с камерой, поэтому выбор останавливается на нативной разработке на Java, обладающей большим количеством библиотек и документации.

3.2 Описание программных модулей

В мобильном приложение реализована так называемая «чистая архитектура» и паттерн MVP. Задача архитектуры программного обеспечения — минимизация человеческих ресурсов, необходимых для создания и обслуживания требуемой системы. Проект Android приложения требует разделения понятий между пользовательским интерфейсом, логикой функционирования и моделью данных. Поэтому приложение разделено на 3 слоя: домен, данные и слой представления.

Домен: содержит определения логики функционирования приложения, модели данных сервера, абстрактное определение репозиториев и определение сценариев использования. На рисунке 11 представлена структура слоя «domain».

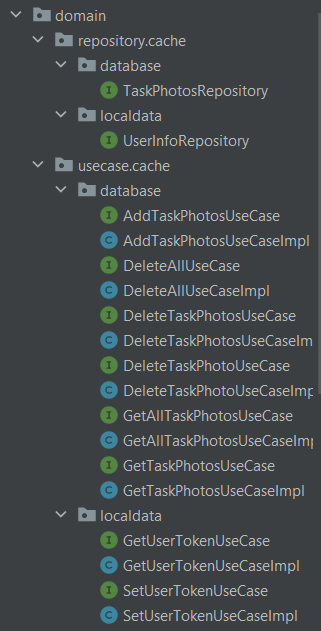


Рисунок 11 – Структура слоя «domain»

Данные: содержит реализацию абстрактных определений доменного слоя. Он содержит репозитории и реализации источников данных, определение базы данных и ее DAO (Data Access Object - паттерн проектирования, смысл которого в создании специальной прослойки, которая будет отвечать исключительно за доступ к данным), определения сетевых API, некоторые средства преобразования для конвертации моделей сетевого API в модели базы данных и наоборот. На рисунке 12 представлена структура слоя «data».

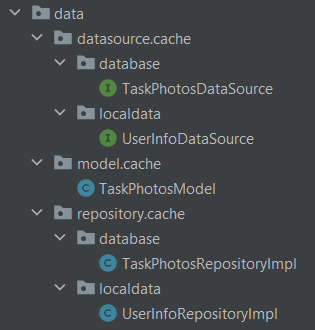


Рисунок 12 – Структура слоя «data»

Слой представления: содержит фрагменты, модели представления, адаптеры, активности, а также реализацию паттерна MVP представленную в пакетах «repositories», «presenters» и «contracts». На рисунке 13 представлена структура слоя представления.

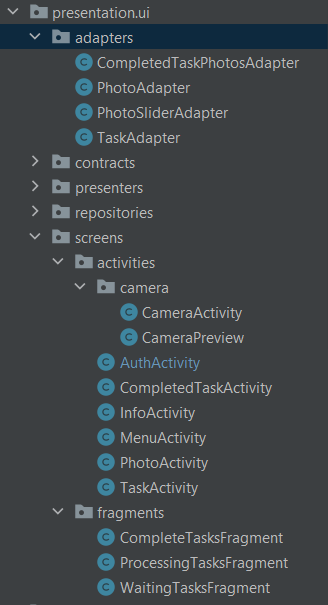


Рисунок 13 – Структура слоя представления

Кроме трех слоев приложения, также присутствует модуль «cache» для работы с локальным хранилищем и базой данных. На рисунке 14 представлена структура модуля «cache».

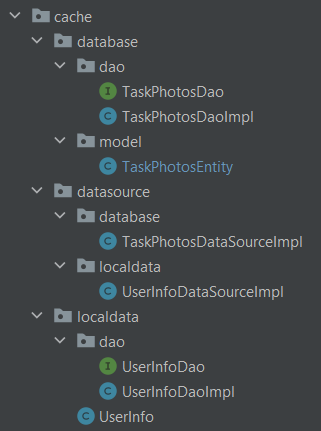


Рисунок 14 – Структура модуля «cache»

В таблице 7 представлено описание назначение пакетов модуля «cache».

Таблица 7 – Назначение пакетов модуля «cache»

|  |  |
| --- | --- |
| Название пакета | Назначение |
| database | Содержит все классы, взаимодействующие с локальной базой данных |
| database.dao | Хранит прослойки, которые отвечают за доступ к данным в локальной базе данных |
| database.model | Хранит модель данных, хранящейся в локальной базе |
| datasource | Отвечает за связь модуля «cache» и слоя «data» |
| localdata | Содержит все классы, взаимодействующие с постоянным хранилищем платформы Android |
| localdata.dao | Хранит прослойки, которые отвечают за доступ к данным, находящимся в постоянном хранилище |

Таким образом, приложение независимо от всего: от интерфейса, от работы с локальной базой данных, от работы сервера и от других элементов окружения. Независимость архитектуры от окружения очень важна, так как это позволяет менять различные компоненты окружения без изменения самой архитектуры.

3.3 Описание разработанного приложения

В процессе разработки мобильного приложения, некоторые макеты экранов, утвержденных заказчиком, были видоизменены в целях удобства для пользователей и расширения функционала. Таким образом, при первом открытии приложения открывается экран авторизации, представленный на рисунке 15. На данном экране специалист по монтажу вводит логин и пароль, полученные от менеджера

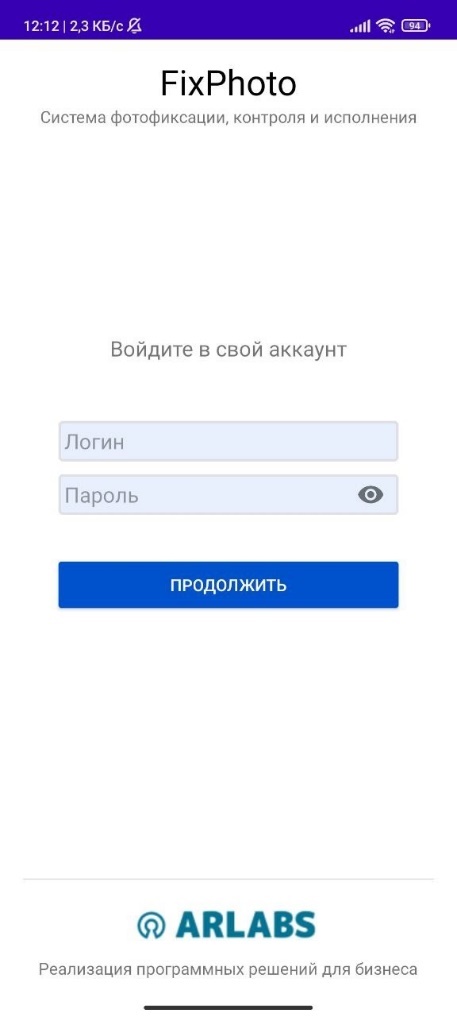


Рисунок 15 – Экран авторизации

При успешной авторизации, открывается основной экран со списком задач. Если учетные данные неверные, то появляется всплывающее уведомление о том, что необходимо ввести правильные данные. На рисунке 16 изображен главный экран, на котором есть возможность просмотра, поиска и фильтрации назначенных на пользователя задач, которые находятся в процессе выполнения. Красным цветом выделяются просроченные задания. Фильтрация осуществляется по объектам, к которым привязаны задачи. Поиск работает по названию и адресу задач.

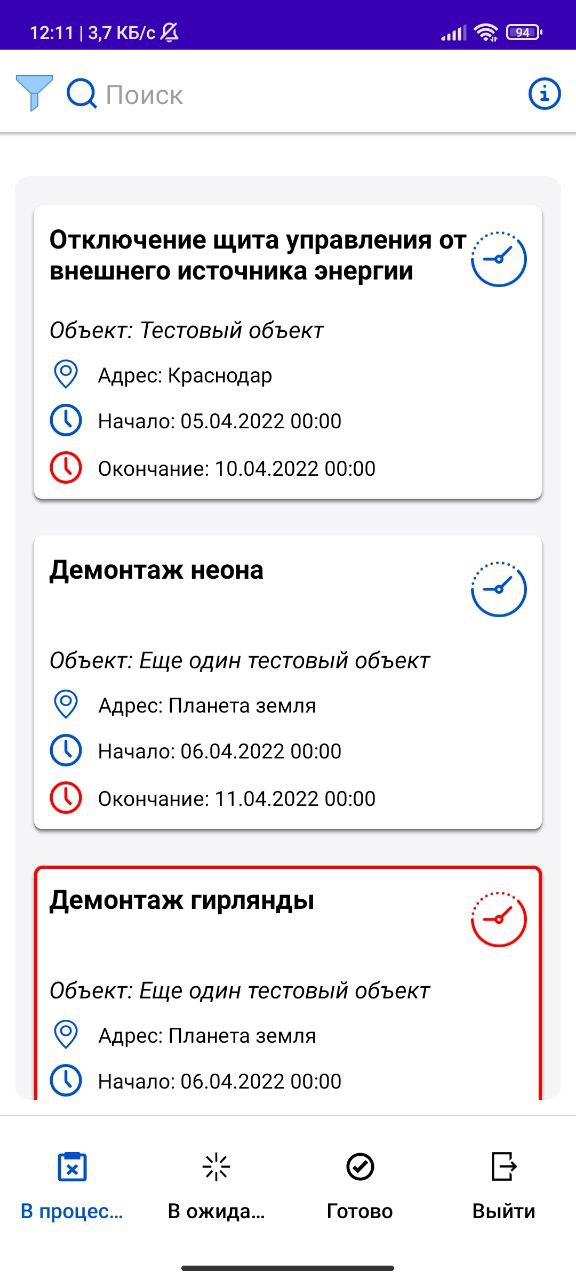
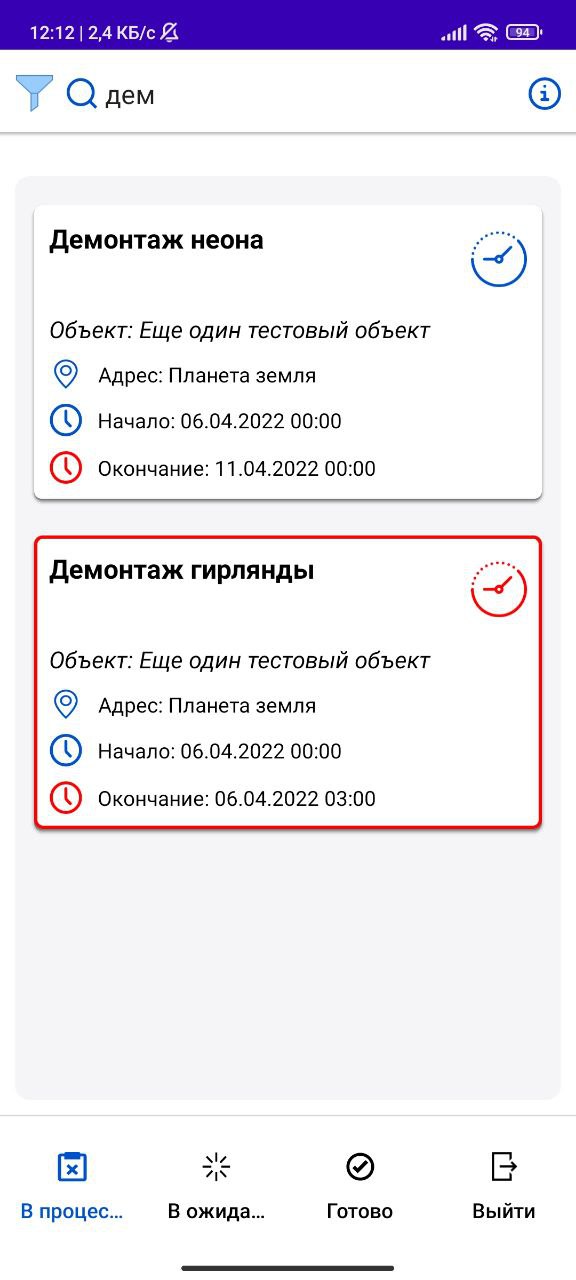
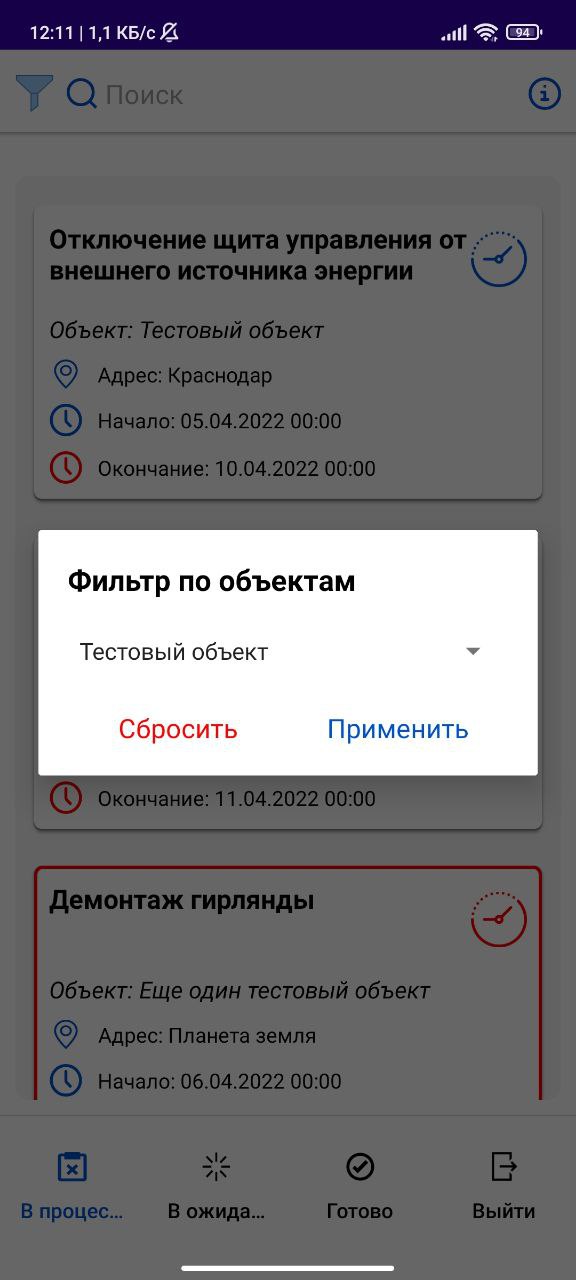
  

Рисунок 16 – Экран с задачами, которые находятся в процессе выполнения

В правом верхнем углу рядом со строкой поиска находится кнопка, при нажатии на которую открывается новый экран, представленный на рисунке 17, с информацией о том, как пользоваться приложением, а также требования к прикладываемым к задачам фотографиям и версия текущего приложения.

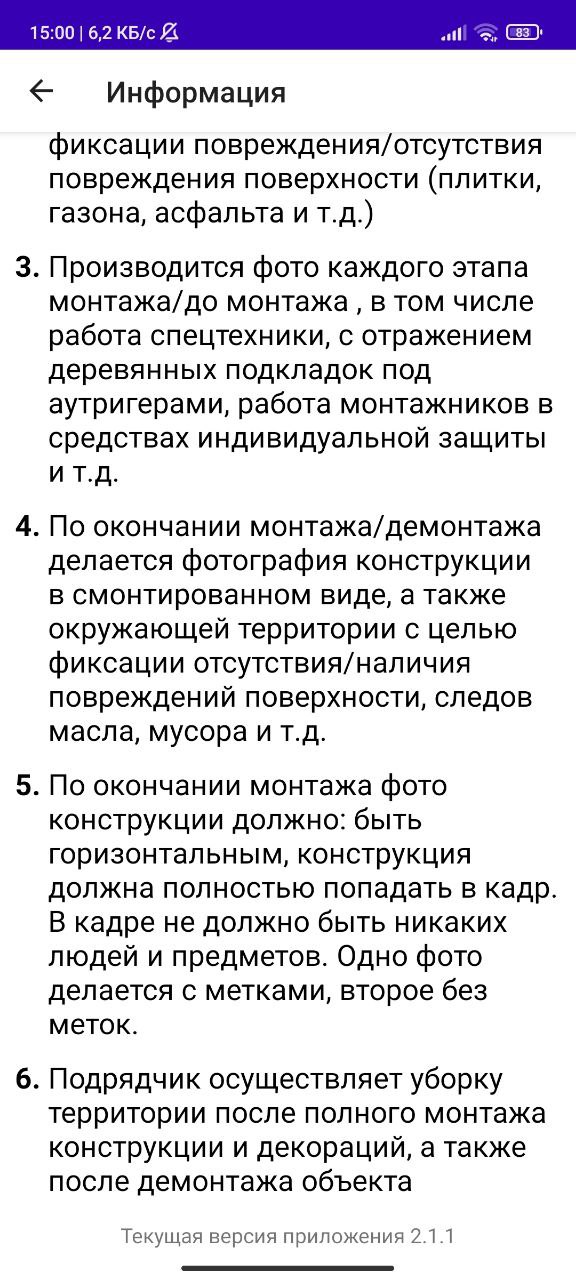
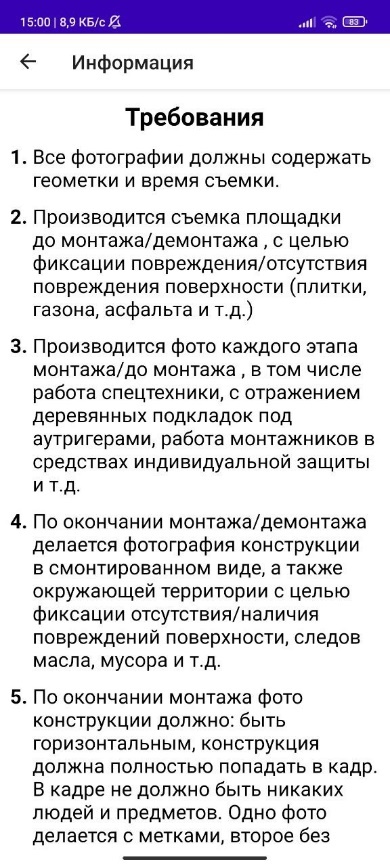


Рисунок 17 – Экран с информацией для пользователя

При нажатии на задачу, которая находится в процессе выполнения, открывается новый экран с подробным описанием задачи и возможностью добавления фотографий и комментария. На рисунке 18 представлен экран задачи.

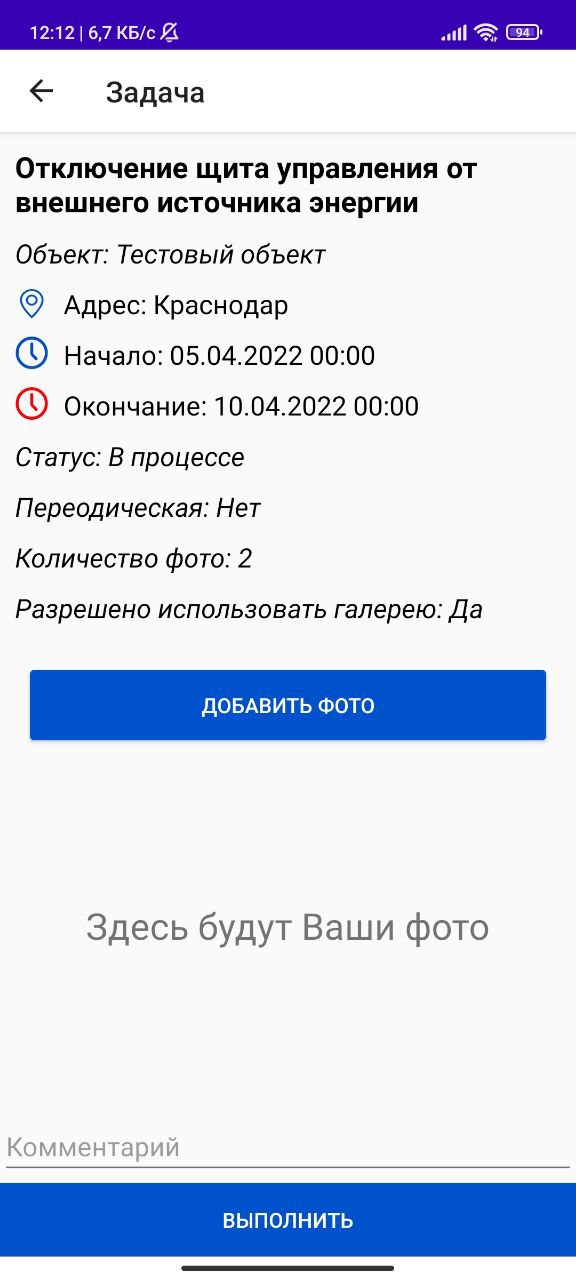


Рисунок 18 – Экран с подробным описанием задачи

На рисунке 18 видно, что есть поле «Разрешено использовать галерею», данный функционал был дополнительно добавлен для возможности прикреплять фотографии из галереи для некоторых задач. Если использование галереи разрешено, то при нажатии на кнопку «Добавить фото» появится диалоговое окно с выбором галереи или камеры. Если использование галереи запрещено, то при нажатии на ту же кнопку, сразу открывается экран с камерой, представленный на рисунке 19. На данном экране в верхнем правом углу отображается количество прикрепленных фотографий к задаче и необходимое количество фотографий, внизу слева направо три кнопки: кнопка работы вспышки, кнопка «сфотографировать» и кнопка подтверждения прикрепления фотографий к задаче.



Рисунок 19 – Экран для работы с камерой

После прикрепления фотографий, снова открывается экран с описанием задачи, только теперь туда добавлены прикрепленные фотографии. При нажатии на крестик фотография удаляется. Если выйти из приложения, то прикрепленные фотографии сохранятся, так как они сохраняются в локальную базу данных. Специалист по монтажу может добавить комментарий в поле ввода при необходимости. После нажатия на кнопку «Выполнить» задача отправляется на сервер, после чего выводится уведомление о том, что задача выполнена, и переходит в статус «Готово». Если количество прикрепленных фотографий меньше, чем требуется в задаче, то выводится всплывающее сообщение о том, что необходимо добавить больше фотографий. На рисунке 20 представлен экран задачи с прикрепленными фотографиями.

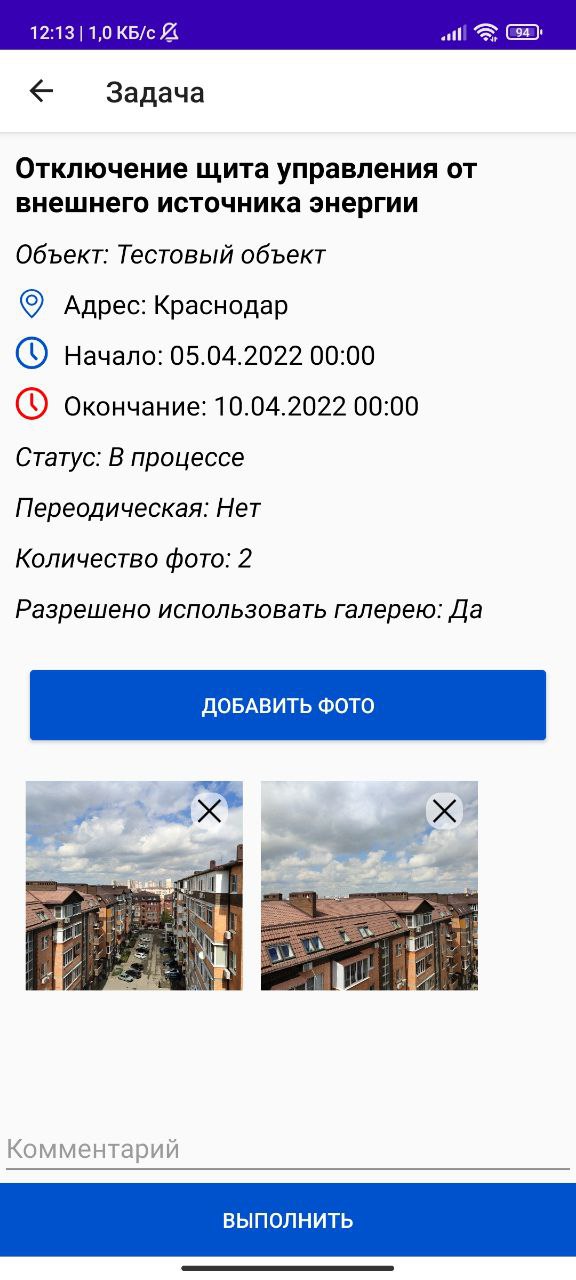


Рисунок 20 – Экран задачи с прикрепленными фотографиями

Разработанное мобильное приложение удовлетворяет всем требованиям технического задания, сформулированным совместно с заказчиком.

4 Тестирование программных модулей

4.1 Интеграционное тестирование

Необходимо протестировать способность мобильного приложения FixPhoto обрабатывать различные ответы сервера, ведь сервер может отключится на некоторое время, поэтому приложение должно быть устойчиво к сбоям со стороны сервера. Сбои трудно эмулировать с реальными серверами, поэтому я воспользовался библиотекой «MockWebServer», которая позволяет легко проверить, правильно ли приложение делает вызовы HTTP и HTTPS. Это позволяет указать какие ответы возвращать, а затем проверить, что запросы были сделаны так, как ожидалось. Таким образом были составлены два ответа JSON. На рисунке 21 представлен пример ответа сервера.

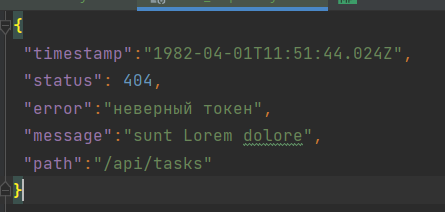


Рисунок 21 – Пример ответа сервера, используемый в тестировании

Были составлены тесты, которые позволяют эмулировать работу сервера и проверять различные коды состояния HTTP. На рисунке 22 приведены примеры тестирования положительного и отрицательного случаев.



Рисунок 22 – Примеры положительного и отрицательного тестов

Тест-кейсы представлены в Приложении Д.

5 Эксплуатационная документация

5.1 Руководство пользователя

При первом открытии приложения необходимо дать доступ к запрашиваемым разрешениям: доступ к камере, доступ к местоположению и доступ к хранилищу данных. После этого появляется окно авторизации, где необходимо ввести учетные данные, полученные от менеджера. После успешной авторизации открывается главный экран. Внизу данного экрана располагается меню с вкладками: «В процессе», «В ожидании» и «Готово», а также кнопка «Выйти». На вкладках появляется список задач, которые находятся в работе, в ожидании и уже выполнены соответственно. Для удобной работы со списком задач присутствует возможность фильтрации по объектам и поиск по названию и адресу. Для того чтобы выполнить задачу необходимо нажать на нее в списке, после чего откроется новый экран с подробным описанием задачи и возможностью добавить или сделать фотографии и написать комментарий при необходимости. Выполненная задача переводится в статус «Готово». Требования к фотографиям объектов:

- производится съемка площадки до монтажа/демонтажа, с целью фиксации повреждения/отсутствия повреждения поверхности (плитки, газона, асфальта и т.д.);

- производится фото каждого этапа монтажа/демонтажа, в том числе работа спецтехники, с отражением деревянных подкладок под аутригерами, работа монтажников в средствах индивидуальной защиты и т.д.;

- по окончании монтажа/демонтажа делается фотография конструкции в смонтированном виде, а также окружающей территории с целью фиксации отсутствия/наличия повреждений поверхности, следов масла, мусора и т.д.;

- по окончании монтажа фото конструкции должно: быть горизонтальным, конструкция должна полностью попадать в кадр. В кадре не должно быть никаких людей и предметов.

5.2 Руководство программиста

Заключение

В результате выполнения выпускной квалификационной работы было разработано мобильное приложение «FixPhoto» с помощью средств разработки языка программирования Java и среды разработки Android Studio.

Мобильное приложение соответствует требованиям заказчика и обладает следующими функциональными возможностями:

- просмотр назначенных задач, в том числе просроченных;

- просмотр выполненных задач;

- выполнение задач, включающее в себя отправку фотографий объекта и комментария;

- поиск и фильтрация задач;

- просмотр справочной информации об использовании приложения;

- возможность добавить фотографии к задаче и сохранить их, для выполнения задачи через время.

Разработанное мобильное приложения можно рассматривать как удобное средство для автоматизации работы специалиста по монтажу.

Список использованных источников

1. ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением N 1) = Unified system for program documentation. Technical specifications for development. Requirements to contents and form of presentation: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3351: введен впервые: дата введения 1980-01-01. – Москва: Стандартинформ, 2010. – 4 с. – Текст непосредственный.

2. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов: руководящий документ по стандартизации: издание официальное: утверждены и введены в действие Постановлением Государственного комитета СССР по управлению качеством продукции и стандартами от 27 декабря 1990 г. № 3380: дата введения 1992-01-01 / Разработан Министерством электротехнической промышленности и приборостроения СССР. – Москва.: ИПК Издательство стандартов, 2002 г. – 27 с. –Текст непосредственный.

3. Мюллер Джон Пол, Семпф Билл, Сфер Чак. С# для чайников.: Пер. с англ. - СПб.: ООО "Диалектика", 2019. - 608 с. – (Профессиональное образование). - ISBN 978-5-907144-43-9. - Текст: непосредственный.

4. Семакин, И.Г. Основы алгоритмизации и программирования: учебник для студ. учреждений среднего профессионального образования / И.Г. Семакин, А.П. Шестаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-6228-3. – Текст: непосредственный.

5. Федорова, Г.Н. Разработка, администрирование и защита баз данных: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 288 с. – (Профессиональное образование). – ISBN 978-5-4468-6994-7. – Текст: непосредственный.

6. Онлайн-школа IT профессий и сообщество программистов. [сайт] – Текст: электронный. – URL: https://itproger.com/ (дата обращения 15.02.2022).

7. Портал для помощи программистам. Текст: электронный. – URL: https://codernet.ru/ (дата обращения 27.03.2022).

8. Сайт о программировании. Полное руководство по языку программирования С# 9.0 и платформе .NET 5: [сайт] – Текст: электронный. – URL: https://metanit.com/sharp/tutorial (дата обращения 20.02.2022).

9. IT Курсы программирования онлайн – обучение программированию, видео уроки. [сайт] - Текст: электронный. Видеоматериалы. – URL: https://itvdn.com/ru (дата обращения 21.03.2022).

10. [UML-диаграммы классов. Программирование. [сайт] – Текст: электронный. – URL: https://prog-cpp.ru/uml-classes (дата обращения 17.03.2022).](https://prog-cpp.ru/uml-classes)

11. Сообщество IT-специалистов [сайт] – Текст: электронный. – URL: <https://habr.com/> (дата обращения 07.04.2022)

12. Android for Developers [сайт] – Текст: электронный. – URL: <https://developer.android.com/> (дата обращения 05.04.2022)

13. Освой программирование играючи [сайт] – Текст: электронный. – URL: <http://developer.alexanderklimov.ru/android/index.php> (дата обращения 01.04.2022)

Приложение А

(обязательное)

Техническое задание. Требования к программным модулям

1 Введение

Настоящее техническое задание распространяется на разработку мобильного клиента для системы «FixPhoto», предназначенной для автоматизации работы менеджеров и специалистов по монтажу. Использовать данное мобильное приложение будут специалисты по монтажу организации ООО «Зодиак-Электро».

Мобильное приложение FixPhoto предоставляет графический интерфейс для сбора, хранения, обработки и отправки информации на сервер о текущих и выполненных задачах, назначенных на пользователя.

Подобная автоматизация повышает эффективность управленческой деятельности и позволяет создать единую систему, состоящую из веб-приложения и мобильного приложения, работающую как слаженный механизм, своевременно анализировать процессы работы организации и повысить качество и скорость выполнения работ по монтажу.

2 Основания для разработки

Основанием для разработки является заказ системы FixPhoto для автоматизации работы сбора фотографий, их обработка и формирование отчетов для компании ООО «Зодиак-Электро» и согласовано с директором Блиновым Е.А.

Наименование темы разработки – «Разработка мобильного клиента для специалистов по монтажу оборудования организации ООО «Зодиак-Электро»».

3 Назначение разработки

Приложение должно быть интегрировано с системой FixPhoto. Оно предназначено для автоматизации работы специалистов по монтажу:

- сбор фотографий объектов;

- контроль выполнения назначенных задач;

- учет времени выполнения каждой задачи;

- контроль состояния объекта после выполнения задачи;

- контроль состояния площадки после выполнения задачи.

4 Требования к программе или программным модулям

4.1 Требования к функциональным характеристикам

После запуска приложения пользователю отображается окно авторизации с формой для ввода логина и пароля. Данные для входа специалисты по монтажу получают от менеджеров. При авторизации приложение получает токен, действующий две недели, пока действует токен, приложение не должно запрашивать логин и пароль у пользователя.

Для специалиста по монтажу программа предоставляет следующие возможности:

- просмотр назначенных задач, в том числе просроченных;

- просмотр выполненных задач;

- выполнение задач, включающее в себя отправку фотографий объекта и комментария;

- поиск и фильтрация задач;

- просмотр справочной информации об использовании приложения;

- возможность добавить фотографии к задаче и сохранить их, для выполнения задачи через время.

4.2 Требования к надежности и безопасности

Приложение должно собирать всю статистику об ошибках в сервис Firebase Crashlytics, а также следить за производительностью при помощи сервиса Firebase Performance Monitoring.

Приложение должно соответствовать современному уровню требований к надежности программного обеспечения:

- предусматривать контроль вводимой информации и блокировку некорректных действий пользователя при работе с мобильным приложением;

- обеспечивать целостность информации, хранящейся в локальной базе данных;

- обеспечивать корректную работу при отправке больших объемов информации;

- токен пользователя необходимо хранить в зашифрованном виде, для обеспечения безопасности всей системы;

- обрабатывать все возможные исключение и сообщать о них пользователю.

4.3 Условия эксплуатации

Приложение запускается на мобильном телефон с операционной системой Android. При первом запуске приложения, необходимо принять все запрашиваемые разрешения на использование хранилища, камеры и геолокации. Телефон должен обладать устойчивым интернет-соединением для выполнения задач.

Специальные климатические условия не требуются.

Приложение не требует проведения каких-либо видов обслуживания.

4.4 Требования к составу и параметрам технических средств

Требования к телефону специалиста по монтажу представлены в таблице А.1.

Таблица А.1 – Требования к телефону специалиста по монтажу

|  |  |
| --- | --- |
| Компонент | Требование |
| Основная камера | От 10 Мп |
| Размер дисплея | От 5 дюймов |
| Оперативная память | Не менее 1 ГБ |
| Стандарт связи | 3G или больше |
| Операционная система | Android 5.0 или выше |

4.4 Требования к информационной и программной совместимости

На мобильном телефоне специалиста по монтажу должна быть установлена операционная система Android 5.0 или выше, а также приложение FixPhoto.

Приложение Б

(обязательное)

Программный код

Приложение В

(обязательное)

Тестирование