



МУП АгроИнвест

Спецификация требований к ПО (СТП)

Версия: 1.3.1

Подготовлено: Александр Полосин

Утверждено: Петр Орлов

Дата создания: 15-Сентябрь-2019

Дата обновления: 28-Сентябрь-2019

ООО «Айти ДАО»

ул. Кооперативная, 97

г. Ульяновск, Россия, 432001

+7 927 818 88 00

mail@daotech.ru

daotech.ru

1. Введение

1.1. Назначение

Целью данного документа является описание мобильного приложения “АгроИнвест” для овощевода (MVP)

1.2. Термины и определения

- MVP - Minimal Viable Product
https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE_%D0%B6%D0%B8%D0%B7%D0%BD%D0%B5%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82
- МУП – Система мобильного управления персоналом на базе 1С ЕРП
- NFC-метка – электронное устройство, аналогичное по размерам электронному пропуску. Содержит информацию о номере ряда/номере тележки.
- А – агроном
- АЗ – агроном-защитник
- АР – агроном рассадного отделения
- МБ – мастер бригады
- БУ - бухгалтер по расчету заработной платы
- ГА – главный агроном
- ГД – Генеральный директор
- З - защитник
- ОВ – овощевод
- ОВР – овощевод рассадного отделения
- РОК – руководитель Отдела кадров
- ОКК – сотрудник Отдела контроля качества
- РОКК – руководитель Отдела контроля качества

1.3. Предполагаемая аудитория и предложения по чтению

Документ предназначен для широкого круга читателей. Различным типам читателей, которым адресован этот документ, следует обратить особое внимание на разделы данного документа:

- Владелец проекта - общее описание системы, требования по качеству и производительности
- Проектный менеджер - весь документ
- Администратор системы - требования к инфраструктуре системы
- Разработчик - функциональные и нефункциональные требования
- Специалист по качеству - требования по качеству
- Системный аналитик/архитектор - нефункциональные требования, общее описание

Данный документ не предназначен для конечных пользователей системы. Для них предусмотрена пользовательская документация, описывающая непосредственно использование системы.

1.4. Границы проекта

На данный момент мобильная составляющая МУП это закрытое приложение на JAVA, которое показало свою необходимость и ценность для АгроИнвеста, но является неподдерживаемым и нерасширяемым

Для дальнейшего развития необходимо реализовать новое приложение, которое позволит в связке с 1С ЕРП достигнуть поставленной бизнес-цели

Разработка бизнес-логики на базе 1С ЕРП снижает стоимость проекта, а так же увеличивает скорость разработки и дальнейшего развития

Детально оцифрованные данные по процессам и автоматизация рутинных функций позволят сократить себестоимость выпускаемой продукции и увеличить эффективность производства

Автоматизация и оцифровка бизнес-процессов на максимально детальном уровне позволит принимать правильные управленческие решения по оптимизации и развитию производственных процессов.

Данный документ содержит в себе требования к мобильному приложению овощевода, которое является одним из множества рабочих процессов МУП АгроИнвест

1.5. Ссылки

В подготовке данного документа были использованы:

- Концепция проекта МУП_v12.pptx
- МУП_Инструкции сотрудников по должностям v13_корп.docx
- Памятка овощеводу_20170619.docx

Прочие ссылки:

- Смартфон Blackview BV6000s 2GB RAM/16GB ROM - <https://4pda.ru/forum/index.php?showtopic=766281&st=3440>

2. Общее описание

2.1. Взаимодействие продукта

Мобильное приложение должно взаимодействовать Middleware-сервером на базе Camunda BPM, которое в свою очередь взаимодействует с ERP системой (1С:Предприятие 8.3, 1С:ERP Управление предприятием 2), получать данные на входе и передавать данные на выходе по средствам API Camunda

Бизнес логика и мастер-система - 1С:Предприятие 8.3 (8.3.15.1565),

1С:ERP Управление предприятием 2 (2.4.8.92)

Camunda BPM 7.11.0

2.2. Общий взгляд на продукт

Мобильное приложение овощевода должно обеспечивать выполнение следующего бизнес-процесса

Овощевод получает с помощью смартфона плановые или срочные задачи на участках агрокомплекса, согласно своему рейтингу и опыту

Его основная функция — найти нужное место и выполнить задание согласно приложенному описанию, сопроводить отчетом заданной формы

Выполнение задачи начинается только после того, как овощевод сканирует NFC метку на участке, который указан в задании

Все данные мобильного приложения через Camunda передаются на сервер (1С ERP) где производится формирование рабочих задач и анализ результата их выполнения

Наиболее важными требованиями являются:

- Работа с NFC – рабочее пространство размечено метками
- Распознавание штрих и QR-кодов – авторизация и некоторые операции производятся с помощью камеры

- Работа в режиме offline, поскольку покрытие Wi-Fi не 100%
- Требования по производительности
- Требования по качеству и отказоустойчивости

2.3. Классы и характеристики пользователей

В проекте выделены две группы пользователей:

1. Руководители (агрономы), которые работают за ПК в 1С ЕРП с настройками и отчетами
2. Мастера участков (также: бригадиры, мастера бригады), которые также работают за ПК в 1С ЕРП с настройками
2. Работники теплиц (овощеводы), которые работают с мобильными устройствами

Агрономы определяют планы и показатели.

Бригадиры — менеджеры среднего звена, на которых агроном делегирует часть своих функций. Помогают агроному обеспечить заданный сверху результат

Агрономы, по желанию, могут выполнять функции бригадира. Например, процесс обучения новичков и их первичная оценка/квалификация — прерогатива бригадира, но ответственные агрономы часто принимают участие и сами оценивают итоги обучения и присваивают квалификацию.

Типичный сценарий использования приложения овощеводом находится в Приложении 4 данного документа.

2.3.1. Структура пользователей с точки зрения предприятия в целом

Агро-комплекс (**главный агроном**)

- Теплица 1 (**агроном теплицы**)
 - Бригада 1 (**мастер бригады**)
 - Овощеводы

Всего на данный момент (Сентябрь 2019) 9 теплиц, в каждой 12 гектаров

Внутри каждой теплицы работает три бригады по 35 овощеводов

Агроном отвечает за всю теплицу, бригадир за участок в теплице, но функции у него схожи.

2.4. Операционная среда

Уже закупленное и эксплуатируемое устройство - Смартфон Blackview BV6000s 2GB RAM/16GB ROM

Операционная система Android 7 "Nougat" или выше

2.4.1. Операционная среда

Требования:

- ЦПУ: Xeon ≥ 2.5 GHz 2 CPU(4 ядра в каждом)
- ОЗУ: 32 Gb с ECC(защита от битых секторов)
- Диски: 2 x SAS 300 Gb
- Материнская плата: обязательно наличие SAS-портов, поддержка DDR4, сетевой интерфейс удаленного подключения, 1 Gbit LAN
- Сетевые интерфейсы:
 - Интерфейс сети передачи данных: 1 Gbit
 - Интерфейс сети управления: любой

Для обеспечения отказоустойчивости необходимо наличие 3-х серверов данной конфигурации.

2.4.2. NAS

Требования:

- ЦПУ: Xeon $\geq 1.5 \leq 2.5$ GHz 4 ядра
- ОЗУ: 4 Gb
- Диски:
 - Файловое хранилище: 4 шт. SAS 500 Gb
 - Системное хранилище: 2 шт. SSD 300 Gb

- Сетевые интерфейсы:
 - Интерфейс сети передачи данных: 1 Gbit
 - Интерфейс сети управления: любой

2.4.3. Сервер бэкапирования

Требования:

- ЦПУ: Xeon $\geq 1.5 \leq 2.5$ GHz 4 ядра
- ОЗУ: 4 Gb
- Диски:
 - Файловое хранилище: 4 шт. SAS 500 Gb
 - Системное хранилище: 2 шт. SSD 300 Gb
- Сетевые интерфейсы:
 - Интерфейс сети передачи данных: 1 Gbit
 - Интерфейс сети управления: любой

2.4.4. Коммутатор

Необходимо наличие 8 портов с пропускной способностью 1 Gb

2.4.5. Маршрутизатор

Необходимо наличие 10 LAN портов с пропускной способностью 1 Gb

2.5. Ограничения дизайна и реализации

Мобильное приложение должно быть разработано на Java или Kotlin с использованием Android SDK.

В качестве архитектуры приложения необходимо использовать архитектурный паттерн MVP (Model-View-Presenter) Для помощи в поддержании архитектурного подхода приложения нужно использовать

библиотеку Mosby (<https://github.com/sockeqwe/mosby>). Обязательно использовать вариант с возможностью управления состояниями view (то есть с дополнением ViewState)

Данный архитектурный подход в мобильных приложениях зарекомендовал себя как гибкое, масштабируемое и надежное решение, которое очень хорошо тестируется благодаря разбивке на зоны ответственности выполняемого кода и помогающее реализовать один из столпов SOLID принципа о единой ответственности компонента

Для реализации паттерна внедрения зависимостей необходимо использовать инструмент Dagger 2 (<https://github.com/google/dagger>)

Паттерн “Внедрение зависимостей” упрощает управление оперативной памятью, уменьшает риск возникновения утечек памяти, уменьшает потребление оперативной памяти и облегчает тестирование приложения в разы

Для коммуникации с API приложение, которое необходимо осуществить через HTTP запросы необходимо использовать библиотеку Retrofit (<https://square.github.io/retrofit/>) в связке с Okhttp 3 (<https://square.github.io/okhttp/>) клиентом

Эти библиотеки зарекомендовали себя как гибкие, надежные и используются в подавляющем большинстве современных mobile проектах. Они обеспечивают максимально возможно качественное соединение со стороны клиента (мобильного устройства) и к тому же возможность логирования всех запросов на этапе разработки помогает сократить время отладки кода

Так как необходима работа с сокетами, то для взаимодействия с ними нужно использовать библиотеку Socket.IO

(<https://socket.io/blog/native-socket-io-and-android>)

Так же необходимо учитывать рекомендации best practices, описанные тут <https://medium.com/@mrcurious/best-practices-of-socket-io-with-android-9f9cb4eb744e>

Так как SocketIO не поддерживает интеграцию с Okhttp3, то необходимо вести логи самим. Для этого предполагается использование такой библиотеки как Timber (<https://github.com/JakeWharton/timber>)

Так же необходимо использовать эту библиотеку для ведения логов системы и отладки

SocketIO - одна из самых популярных библиотек для работы с сокетами в android, которая обеспечивает быструю разработку, минимально возможное написание кода, а так же надежное соединение с сетью

Timber - легковесная библиотека для более удобного ведения логов системы, что облегчает внедрение логов, а также уменьшает количество кода при динамических переменных лога. Кроме того имеет в себе функционал записи логов в файл, что крайне важно, если невозможно отображения логов в консоль напрямую (например, отсутствие компьютера во время теста или отладки)

Для работы с форматом передачи данных JSON необходимо использовать библиотеку GSON (<https://github.com/google/gson>)

Библиотека GSON намного упрощает работу с форматом данных JSON, помогает сделать код чище и понятнее, что хорошо сказывается на время разработки, поддержки приложения, а также понимания кода новыми членами команды

Для наиболее ресурсоемких операций (таких как работа с сетью, работа с БД и другими вычислительно сложными или медленными операциями) необходимо использовать реактивный подход с помощью библиотеки RxJava (<https://github.com/ReactiveX/RxJava>)

RXJava - это библиотека, которая предоставляет возможность переводить объемные вычисления в потоки и увеличить отзывчивость и скорость UI за счёт переноса объемных задач в фоновые потоки

Для реализации UI части необходимо описать на рекомендации от Google по Material Design (<https://material.io/>)

Дизайн, представленный Google для мобильных устройств на платформе android и рекомендованный для реализации в новых мобильных приложениях

Для уменьшения количества кода и улучшения читаемости его необходимо использовать View binding для обращения к элементам UI (<https://kotlinlang.org/docs/tutorials/android-plugin.html>)

Данная библиотека уменьшает объем повторяющегося кода и улучшает его читаемость, что хорошим образом сказывается на время реализации задач

Для распознавания бар кода нужно использовать firebase ML Kit (<https://firebase.google.com/docs/ml-kit/read-barcodes>)

Это стабильная, мощная библиотека для разработанная Google и предоставляющая возможности компьютерного зрения и обучения, что гарантирует хорошее решение задачи по распознаванию штрих кода

Для взаимодействия со встроенной базой данных мобильного приложения необходимо использовать библиотеку Room (<https://developer.android.com/jetpack/androidx/releases/room>)

Это слой абстракций над встроенной библиотекой SQLite в android, позволяющий упростить обращение к Базе Данных, что существенно снижает время, которое затрачивается на реализацию задач, связанных с работой с базой данных

Для взаимодействия с чипом nfc необходимо использовать рекомендации, описанные тут: (<https://expertise.jetruby.com/a-complete-guide-to-implementing-nfc-in-a-kotlin-application-5a94c5baf4dd>)

Для Unit Тестирования необходимо использовать Robolectric (<http://robolectric.org/>)

Для Интеграционного тестирования необходимо использовать Espresso (<https://developer.android.com/training/testing/espresso/index.html>).

2.6. Пользовательская документация

На данном этапе, для мобильного приложения в качестве пользовательской документации можно использовать Приложение 4 данного документа, Приложение 5 и Приложение 6

2.7. Допущения, предположения и зависимости

В данном документе предполагается что взаимодействие с 1С ЕРП происходит посредством выполнения REST-запросов через API, предоставляемое 1С ЕРП. Взаимодействие с API предполагается по принципу “черного ящика”

Структура и состав запросов/ответов должны быть выяснены в ходе проведения работ по созданию мобильного приложения и внесены в данный документ

Подразумевается что API 1С ЕРП содержит все необходимые методы и данные для выполнения функциональных требований данного документа

Распределение задач согласно рейтинга происходит на стороне 1С ЕРП и попадает на связку Middleware Camunda + мобильное приложение в готовом виде

Также подразумевается что доступны следующие возможности через REST API предоставляемое 1С ЕРП:

- Получение списка сотрудников
- Получение информации о сотруднике по ID
- Получение списка задач общее (с фильтрацией и пагинацией)
- Получение детальной информации о задаче по ID
- Загрузка отчета по задаче с указанием ID

3. Функции системы

Каждая функция должна быть оценена с помощью Матрицы Эйзенхауэра: оценка по двум параметрам:

- Важность
- Срочность

Соответственно, получаем 4 группы задач:

- Срочные важные

- Срочные неважные
- Несрочные важные
- Несрочные неважные

Выполняться они должны именно в таком порядке

3.1. Авторизация

3.1.1. Описание и приоритет

Важно, срочно

Для того чтобы получить список заказ-нарядов пользователь должен авторизоваться в мобильном приложении, так как данный список назначается на каждого конкретного сотрудника.

Также требуется отслеживать активность сотрудника и получать отчеты о проделанной работе. Данная информация необходима в подсчете ЗП и КРП сотрудников, для чего требуется идентификация

3.1.2. Последовательности Запуска/Ответа

Экран авторизации по штрих-коду должен быть показан при запуске мобильного приложения. После авторизации на данный экран можно вернуться только разлогинившись из приложения.

3.1.3. Функциональные требования

При помощи мобильного приложения необходимо считать специальный штрих-код (табельный номер), отослать его на сервер, получить ответ, найден данный штрих-код в системе или нет. Отрицательный результат показать на экране. Положительный результат сохранит код сотрудника внутри мобильного приложения на время сеанса работы.

Код (табельный номер) также можно ввести вручную. В поле для ввода разрешены только цифры, поэтому необходимо показывать специальную цифровую клавиатуру.

После залогинивания показать страницу с детальной информацией о сотруднике

3.2. Заказ-наряды. Работа со списком

3.2.1. Описание и приоритет

Важно, срочно

После авторизации с сервера необходимо получить список работ для сотрудника

3.2.2. Последовательности Запуска/Ответа

Список задач появляется после авторизации. Также можно перейти на экран списка нажатием кнопки “Назад” на мобильном устройстве с любого другого экрана

Также можно обновить список задач с сервера вручную, находясь на экране списка задач, потянув список вниз (свайп вниз)

3.2.3. Функциональные требования

После получения списка работ необходимо их показать на экране в виде простого списка. Задачи сгруппированы по участкам и типам, по нажатию на участок можно получить список работ на участке.

Если список работ или участков занимает более одного экрана, необходимо реализовать механизм прокрутки

3.3. Заказ-наряд. NFC-метка и работа

Разделить работу над задачей на два пункта, NFC и сама работа

3.3.1. Описание и приоритет

Важно, срочно

На странице заказ-наряда необходимо начать работу над ним, считав NFC-метку

По завершении работы необходимо предоставить отчет

3.3.2. Последовательности Запуска/Ответа

Экран заказ-наряда появляется после нажатия на элемент списка заказ-нарядов определенного участка

3.3.3. Функциональные требования

Работа над заказ-нарядом начинается автоматически после считывания соответствующей метки. Если метка не соответствует заказ-наряду, необходимо показать сообщение об ошибке

Если метка соответствует, начинается работа над заказом и стартует таймер отсчета затраченного времени, если метка содержит данные, необходимые для начала работы (например, номер ряда и тележки), они заполняются на экране автоматически

Для начала работы требуется чтобы телефон был онлайн, при этом отправился запрос на сервер проверяющий блокировку задачи, и только если задача не заблокирована, блокировать ее и начинать работу в мобильном приложении

В случае, если никакой заказ-наряд не выбран, и приложение находится в состоянии “ожидание задания” , при считывание метки внутри мобильного приложения происходит поиск подходящей задачи. Если задача найдена, то автоматически показывается экран данной задачи с которого можно начать работать над ней

В ходе работы пользователь может предоставить текстовое описание работы и загрузить в память мобильного устройства одну или несколько фотографий, согласно форме отчета необходимой для данной задачи

3.4. Заказ-наряд. Завершение работы

3.4.1. Описание и приоритет

Важно, срочно

После завершения работ над заказ-нарядом пользователь может нажать на кнопку завершения работы и завершить задачу

3.4.2. Последовательности Запуска/Ответа

После начала работ над заказ-нарядом становится доступной кнопка “Завершить работу”

3.4.3. Функциональные требования

После начала работ над заказ-нарядом и в случае если введены все необходимые для отчета по задаче данные, становится доступной кнопка “Завершить работу” по нажатию на которую:

- Останавливается таймер
- Загруженные данные отправляются на сервер
- Заказ-наряд помечается как завершённый в списке

3.5. Заказ-наряд. Оффлайн-режим

3.5.1. Описание и приоритет

Важно, срочно

Приложение должно корректно отправлять данные в условиях нестабильного сетевого соединения

3.5.2. Последовательности Запуска/Ответа

Если в ходе рабочей сессии соединение с сервером потеряно, то приложение должно перейти в оффлайн-режим и показать соответствующее сообщение на экране

3.5.3. Функциональные требования

Если в ходе рабочей сессии соединение с сервером потеряно, то приложение должно перейти в оффлайн-режим и показать соответствующее сообщение на экране.

Накопленные данные при этом не должны быть потеряны и отправлены на сервер когда соединение восстановится

3.6. Состояния приложения

3.6.1. Описание и приоритет

Неважно, срочно

Приложение в каждый конкретный момент времени может находиться в одном из нескольких состояний, определяющих доступные функциональные возможности:

- Ожидание авторизации

- Ожидание задачи
- Задача в работе

Также есть под-состояния:

- Ожидание ответа на запрос
- Соединено с сервером
- Соединение потеряно

3.6.2. Последовательности Запуска/Ответа

Состояние контролируется пользователем через взаимодействие с прочими функциональными возможностями

3.6.3. Процесс Samunda ответ на запрос мобильного приложения

Важно, срочно

Типы:

- Авторизация
- Выход
- Получения списка задач
- Получение информации о залогиненном пользователе
- Получение информации о задаче
- Отправка информации о задаче
- Синхронизация временной метки

4. Требования к данным

4.1. Логическая модель данных

4.1.1. Входные данные:

Справочники

- Сотрудники
- Должности
- Подразделения
- Виды работ
- Структура участков

Документы

- Заказ-наряды

4.1.2. Выходные данные:

1. Статус выполнения заказ-наряда
2. Прикрепленное фото
3. Комментарий к заказ-наряду
4. Мета-данные заказ-наряда (время начала-время окончания, GPS координата места аварии,)

4.2. Отчеты

Внутри мобильного приложения отчеты не предусмотрены

4.3. Получение, целостность, хранение и утилизация данных

Получение списка заказ-нарядов происходит после авторизации пользователя по штрих-коду.

Накопленные данные хранятся в мобильном устройстве и отправляются на сервер при наличии соединения.

Если данные отправлены, то после выхода из приложения, данные будут удалены.

Если данные не отправлены, то при выходе из приложения данные будут сохранены и отправлены при наличии соединения при следующем подключении к сети.

Данные, введенные другим сотрудником не будут видимы в этом случае.

5. Требования к внешним интерфейсам

5.1. Пользовательские интерфейсы

Для прототипирования приложения требуется разработать макеты экранов с помощью:

- <https://marvelapp.com>
- <https://www.sketch.com/>
- <https://zeplin.io/>

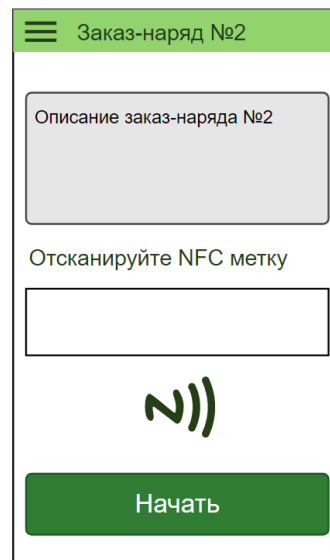
Предварительные макеты:

- <https://marvelapp.com/9ba768g/screen/61457284>

5.1.1 Авторизация

5.1.2 Задачи

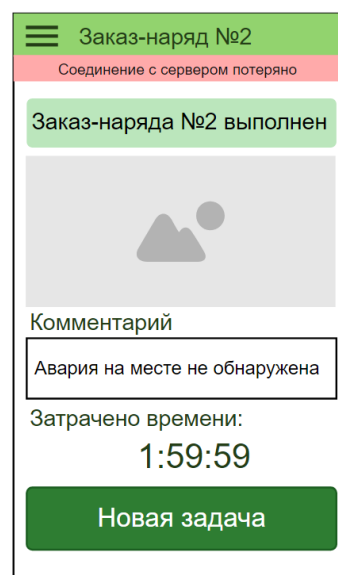
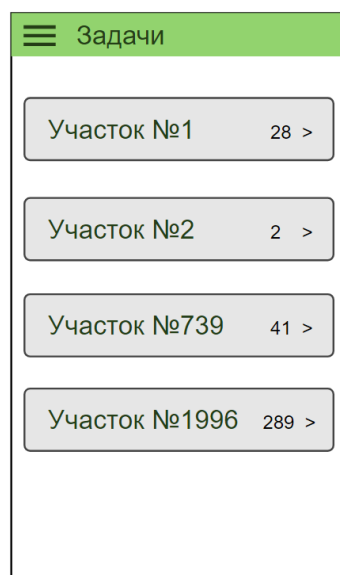
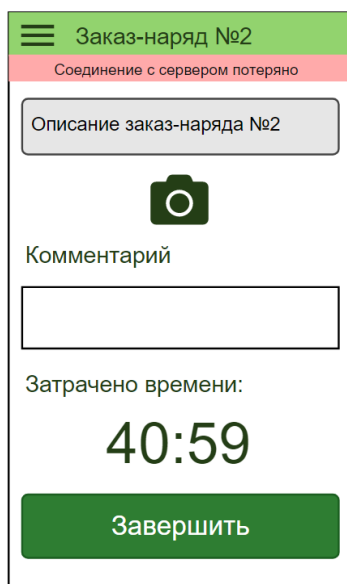
5.1.3 Заказ-наряд

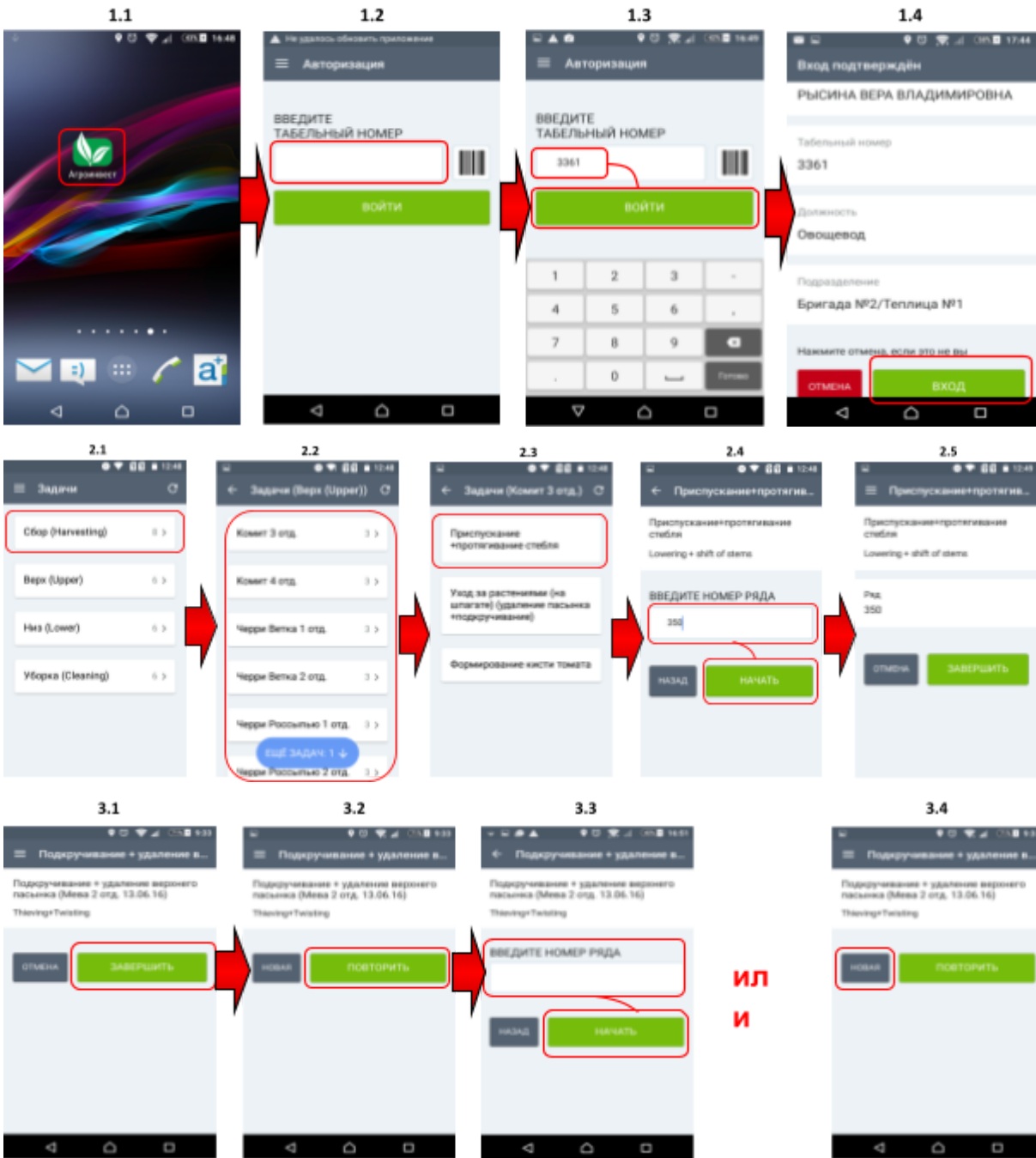


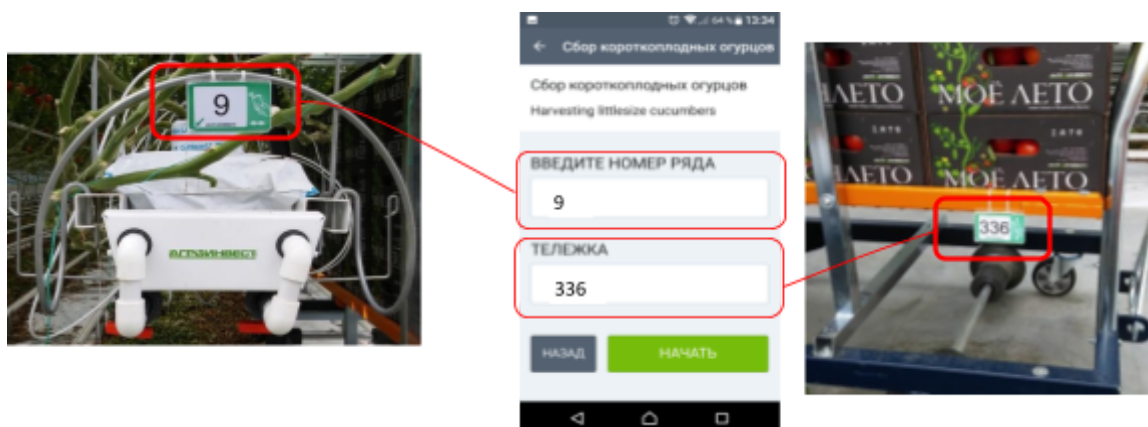
5.1.4 Работа

5.1.5 Участки

5.1.6 Завершение







5.2. Коммуникационные интерфейсы

Мобильное приложение и API 1С ЕРП будут взаимодействовать через REST-вызовы через HTTPS-протокол

6. Нефункциональные требования

6.1. Удобство использования

Важно, срочно

Дизайн должен быть спроектирован таким образом чтобы работа с приложением была интуитивно-понятна конечному пользователю системы, согласно типичному пользовательскому сценарию описанному в Приложении 4.

Приложение должно быть разработано таким образом чтобы элементы интерфейса позволяли удобную работу в перчатках

6.2. Производительность

Важно, срочно

Мобильное приложение должно быть спроектировано с учетом возможных высоких нагрузок до 2 000 – 4 000 пользователей онлайн в пиках.

Пиками считается время начала и окончания дня, а также перерывы и обед. В эти моменты пользователи мобильных приложений будут производить авторизацию и массовые запросы по новым/выполненным задачам.

Авторизация, выход из системы и загрузка данных не должны занимать более 3 секунд.

МУП работает с SLA 99% (возможная недоступность до 14 мин./сутки)

Данные требования распространяются не сколько на мобильное приложение, сколько на обслуживающий backend-сервер.

6.3. Техника безопасности

Важно, несрочно

При работе с мобильным приложением требуется соблюдать ТБ, принятую на предприятии. В частности, не снимать защитные элементы одежды (очки, перчатки).

6.4. Атрибуты качества

Важно, несрочно

Все функциональные требования должны быть покрыты автотестами на 100%

Необходимо проведение модульного(unit), интеграционного и UI тестирования для мобильного приложения.

Из всех тестов должно приходиться примерно 70% на unit тесты, 20% на интеграционные тесты и 10% на UX тесты исходя из рекомендаций Google (<https://developer.android.com/training/testing/fundamentals>)

Атомарные функции необходимо покрывать unit тестами.

После прохождения unit тестирования необходимо провести интеграционное, которое подразумевает взаимодействие между компонентами.

После успешного прохождения интеграционного тестирования, необходимо проводить UX тестирование, которое подразумевает корректную работу UX компонентов. Тестирование считается успешным если пройдено 100% тестов

6.5. Требования безопасности

Важно, срочно

Обмен данными между мобильным приложением и backend-сервером должен быть зашифрован

6.6. Требования по локализации

Локализация продукта на языках помимо русского не требуется

7. Установка

Первоначальная установка приложения на мобильные устройства может происходить при подключении устройства USB-кабелем к компьютеру и полуавтоматической загрузки с помощью специального bash-скрипта с использованием технологии adb:

- <https://developer.android.com/studio/command-line/adb>

Мобильное устройство при этом должно быть заранее переведено в режим разработчика согласно документации к устройству, а также разрешена установка приложений из сторонних источников

Примечание от составителя: можно использовать системы управления рабочим профилем, предоставляемые платформой Android (Corporate Google Play), которая предоставляет весьма широкие возможности по управлению и контролю парком мобильных устройств любого предприятия.

Однако это требует использования специальных EMM программ:

- <https://developers.google.com/android/work/>

Данные программы как правило предоставляются по подписке, например пакет G-Suite от Google, или системы Miradore, Genydeploy и прочие. Стоимость использования, от 2\$/пользователь в месяц, что с учетом потенциально возможного количества сотрудников в 4000 приводит к значительным затратам. Использование данных программ требует дальнейшей проработки по поводу целесообразности их использования.

8. Обновление

При запуске программы, она может с командного сервера запрашивать текущую версию программы и сравнить ее с номером версии, установленной на устройстве.

Если номера версий не совпадают, программа должна предложить обновление и по нажатию на кнопку подтверждения обновиться на устройстве.

Таким образом обновление программы в общем случае не требует подключения к компьютеру и ручной переустановке в отделе технического обеспечения.

9. Приложение 1: Словарь терминов

Не требуется на данном этапе

10. Приложение 2: Аналитическая модель

Не требуется на данном этапе

11. Приложение 3: Список вопросов

Список новых процессов для автоматизации в дальнейшем:

- Автоматическое распределение задач для работников теплиц.
- Автоматический расчет ЗП с учетом оценок контроля качества и аутсорсинга.
- Фиксация болезней/вредителей

12. Приложение 4: Сценарий пользователя

(для аварийного техника)

1. Я как аварийный техник, могу авторизоваться в мобильном приложении по штрих-коду, отсканированному камерой смартфона, чтоб начать рабочий день.
2. Я могу получить список заказ-нарядов с указанием участков, чтоб начать работу над конкретным заказ-нарядом в конкретной локации.
3. Получив заказ-наряд, я следую в назначенный участок к NFC метке, которая расположена на участке.
4. Я сканирую метку смартфоном и приложение позволяет мне начать работу по поиску аварий в этой локации.
5. В процессе движения я вижу в приложении индикацию состояния подключения к серверу.
6. Я оказываюсь вне зоны сети и приложение переходит в offline режим, отображая этот статус на экране.
7. Обнаружив место аварии, я могу сделать фото аварии и комментарий, которые будут сохранены в приложении.

8. Вернувшись в зону приема сети, приложение подключается к серверу и автоматически отправляет сделанные фото и комментарий на сервер. Так же отправляется статус выполнения задачи и время сделанной фотографии.
9. Я перехожу к выполнению следующей задачи (переход на шаг 3)
10. По итогам дня я завершаю рабочий сеанс и приложение фиксирует время завершения рабочего дня. Если в данный момент сервер не доступен, данные сохраняются и отправляются при следующем подключении к сети.

13. Приложение 5: Памятка Овощеводу

1. Войдите в Приложение

Запустите Приложение, Ввести свой табельный номер и нажмите кнопку «Войти»

Приложение открывает экран, на котором отображены ФИО, табельный номер, должность и подразделение сотрудника

Проверьте корректность отображенной информации и нажмите кнопку «Вход». Приложение сразу же отображает группы и количество задач на экране «Группы задач»



2. Выберите задачу и приступайте к ее выполнению

Список задач также можно вызвать из меню, нажав на кнопку в левом верхнем углу экрана

При успешной регистрации Приложение отображает список текущих (запланированных) заданий

Структура заданий представляет собой список наименований работ, которые доступны на сегодня. Работы группируются по папкам. Поэтому, сначала выбирается папка, а потом конкретное задание в папке

Например, на рисунке ниже, папка «Верх (Upper)» содержит три задания

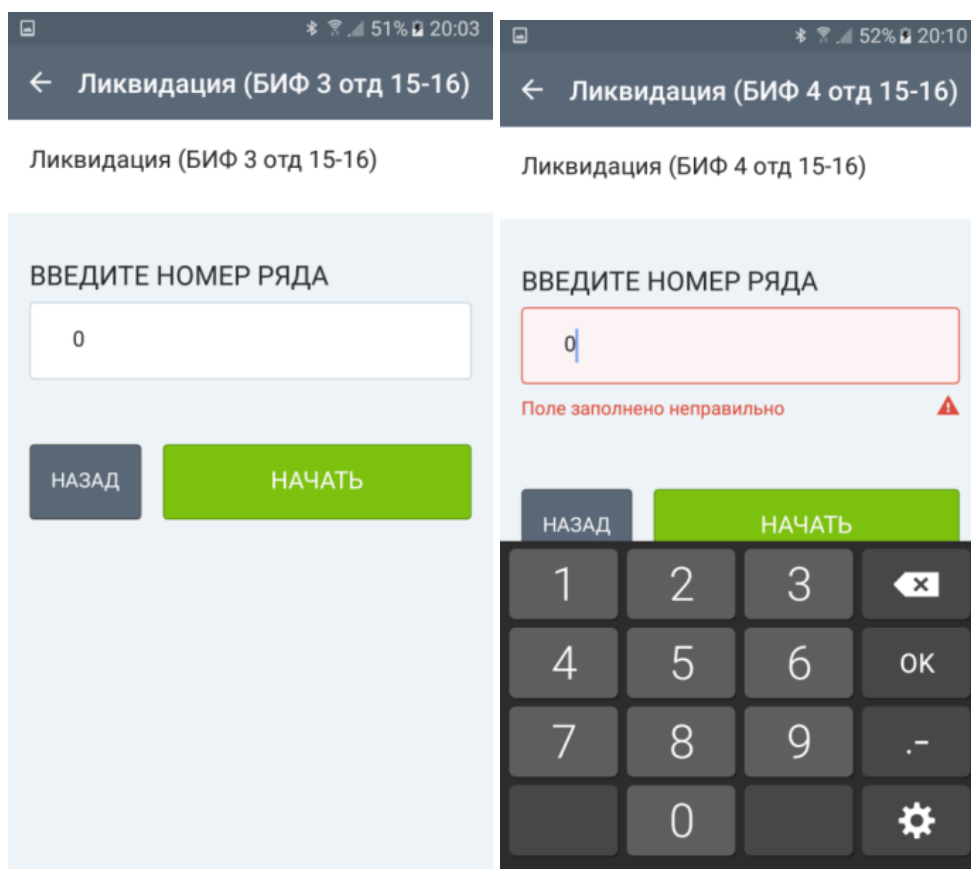
После выбора Задания необходимо ввести в Приложении параметр, обязательный для данной работы



Для каждого из типов работ настроен свой список параметров на вход. Например, для работы «Уборка листа» - необходимо ввести **номер ряда**

Внимание, если номер ряда не относится к допустимому диапазону, то Приложение выдаст ошибку. В этом случае введите правильный номер ряда

Для каждого из типов запрашиваемых данных с сервера передается проверка. Например, ряды по работам в первом отделении Томатной теплицы 1 будут проверяться, что они в диапазоне от 1 до 130



После правильного ввода всех необходимых параметров Приложение позволит нажать кнопку «Начать». После этого приступайте к выполнению задачи, выбранной в приложении. По завершению выполнения задачи нажмите кнопку «Завершить».

Далее, если есть аналогичные задачи, в Приложении можно нажать кнопку «Повторить». В этом случае, вы должны ввести следующий ряд и нажать кнопку «Начать».

Если аналогичных задач нет - нажмите в Приложении кнопку «Новая». После этого отобразится весь список «Группы задач».



3. Завершите выполненную задачу и выберите следующую

При завершении каждой задачи обязательно нажимайте кнопку «Завершить». **Без нажатия на кнопку «Завершить» новые задачи недоступны**

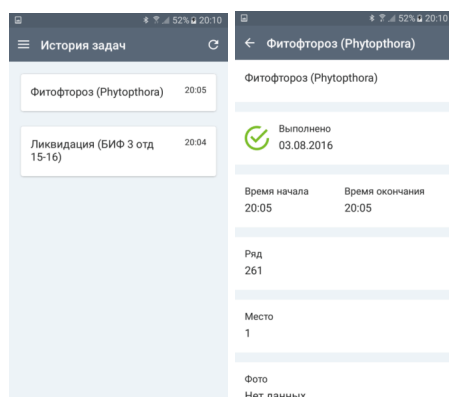
Для отметки об окончании некоторых работ (например, Сбор) введите дополнительный обязательный параметр

Например, для отметки об окончании работы по сбору, необходимо ввести в Приложении номер «Тележки», в которую осуществлялся сбор. Только после ввода данного параметра кнопка «Завершить» станет доступной




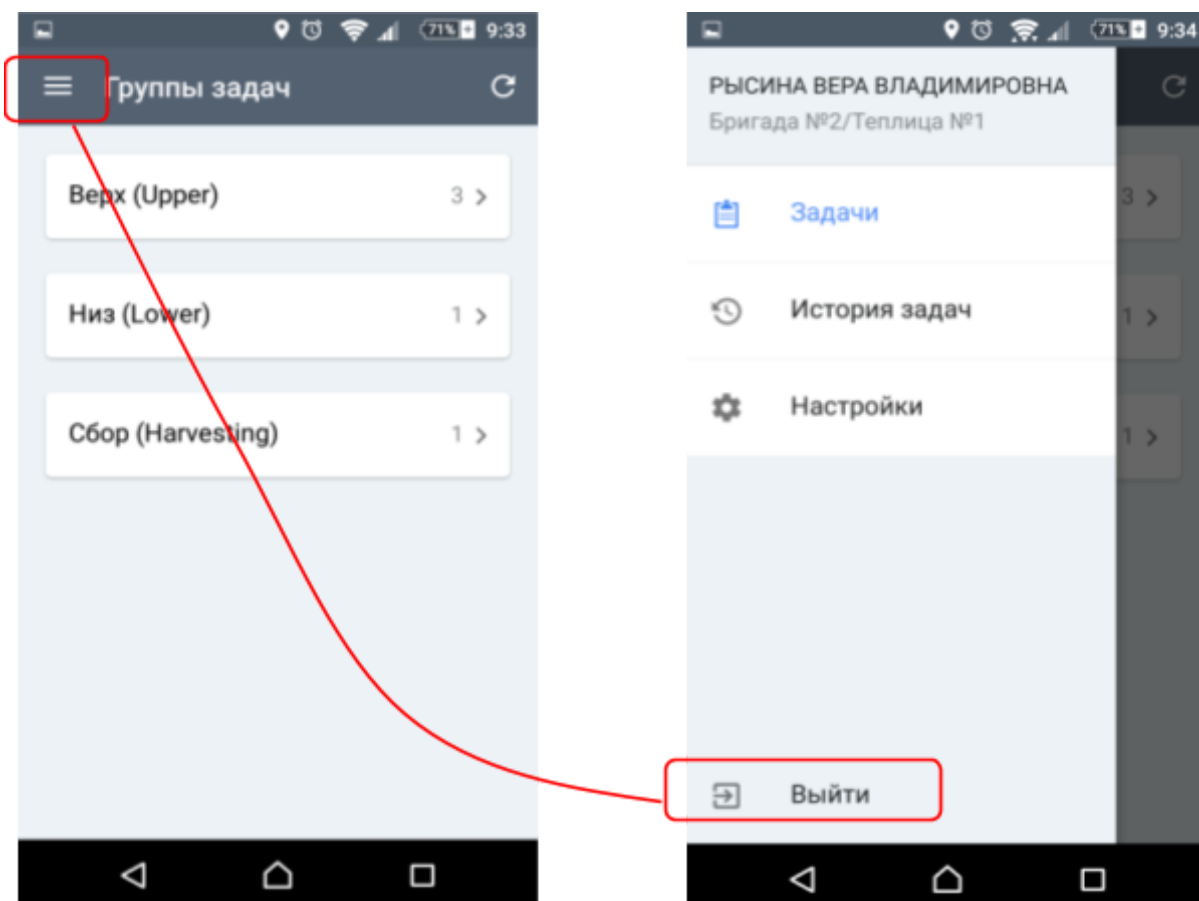
4. История задач

Все сделанные за день отметки можно посмотреть в меню «История задач» В данном пункте меню отображаются даты начала и завершения, а также параметры выполнения (например, ряд и тележка)



5. Выход из приложения

Для полного выхода из приложения, войдите в меню, нажав на кнопку , и выберите пункт «Выйти»



14. Приложение 6: Действия при выявлении ошибок

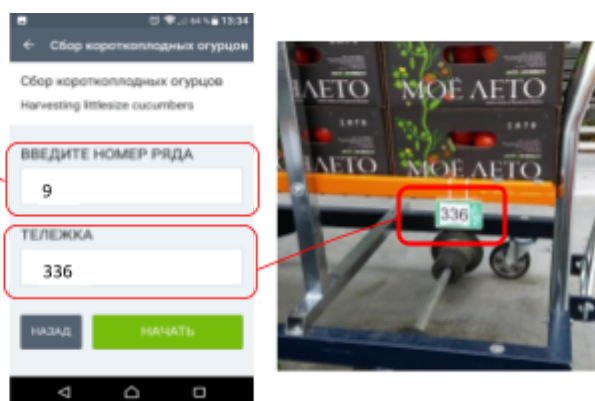
1. Не хватает отметок по выполненным работам:
 - a. Проверьте, что последняя работа на мобильном устройстве завершена. Если нет - завершите;
 - b. Если отметки осуществлялись и об этом есть запись на телефоне (в Истории задач) - переподключитесь к Wi-Fi;
 - c. Если отметки в мобильном приложении не осуществлялись по какой-либо причине, то выполните недостающие задачи на своём мобильном устройстве с указанием рядов и тележек (если это требуется для окончания работы);
2. В Задаче неправильно указан ряд/тележка:
 - a. Если текущая работа с некорректным рядом на мобильном устройстве ещё не завершена - отмените эту работу;
 - b. Если задача с некорректными данными уже завершена - обратитесь к мастеру бригады;
3. Лишняя отметка о выполненной Задаче:
 - a. Если текущая работа на мобильном устройстве ещё не завершена – отмените эту задачу;
 - b. Если ненужная задача уже завершена - обратитесь к мастеру бригады;

- с. Внимание! В **теплицах №3 и №4** для ввода номера ряда и номера тележки необходимо сканировать телефоном (центральным выступом задней крышки) специальные карточки, расположенные на лотках и на боковых частях тележек.

Карточка номера ряда



Карточка номера тележки



15. Приложение 7: Инструкция овощеводу по уходу за мобильными устройствами

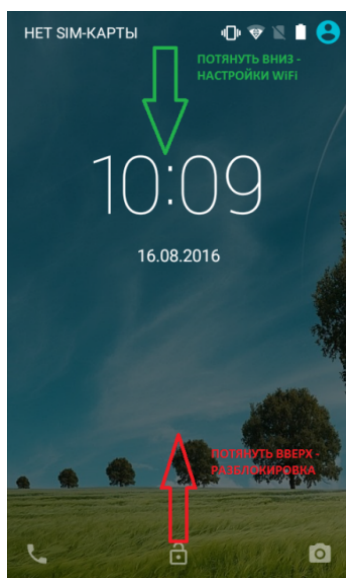
Индивидуальное мобильное устройство (телефон) выдается каждому овощеводу под роспись. Каждый телефон имеет индивидуальный номер, расположенный на задней крышке устройства.

1. При получении телефона необходимо проверить его исправность и отсутствие внешних повреждений (царапины, сколы, вмятины). В случае выявления каких-либо повреждений, необходимо обратиться к мастеру бригады и потребовать замены телефона.
2. Перед началом работы необходимо включить телефон. Проверить есть ли соединение с Wi-Fi и правильно ли установлена Дата. См. Рисунок 1.
3. Во время работы необходимо бережно относиться к телефону:
 - а. предохранять телефон от падений и попадания влаги;

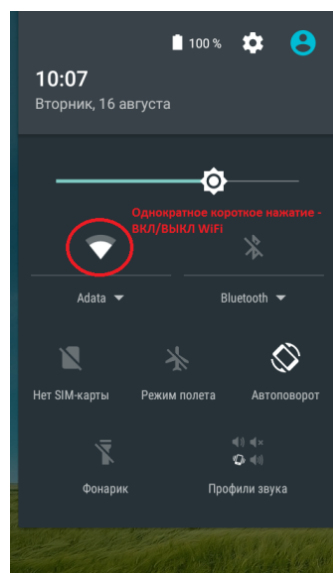
- b. использовать устройство чистыми, сухими руками;
 - c. производить чистку телефона каждый раз при окончании смены;
4. По окончании работы телефон необходимо поместить на зарядку в специальный шкаф (оборудован ячейками для отдельного хранения), предварительно очистив телефон от загрязнений влажной салфеткой для протирки. Телефон – не выключать!
 5. Место для хранения и зарядки телефона необходимо содержать в чистоте. Хранить посторонние предметы в гнездах (ячейках) для телефонов запрещается!
 6. В случае повреждения телефона во время работы, нужно немедленно обратиться к мастеру бригады.

Рисунок 1: «Перезапуск Wi-Fi»

Шаг 1: потянуть верхнюю часть экрана вниз, до появления меню настроек



Шаг 2: выключить Wi-Fi. Затем включить Wi-Fi



16. Приложение 8: Сценарий пользователя (для овощевода)

1. Я, как овощевод, могу авторизоваться в мобильном приложении по штрих-коду, отсканированному камерой смартфона, чтоб начать рабочий день.
2. Я могу получить список заказ-нарядов/заданий с указанием участков, чтоб начать работу над конкретным заказ-нарядом в конкретной локации.
3. Получив заказ-наряд, я следую в назначенный участок к NFC метке, которая расположена на участке.
4. Я сканирую метку смартфоном и приложение позволяет мне начать работу по заданию в этой локации.
5. В процессе работы я вижу в приложении индикацию состояния подключения к серверу.
6. Я оказываюсь вне зоны сети и приложение переходит в offline-режим, отображая этот статус на экране.
7. Обнаружив место задания, я могу сделать фото работ и комментариев, которые будут сохранены в приложении.
8. Вернувшись в зону приема сети, приложение подключается к серверу и автоматически отправляет сделанные фото и комментарий на сервер. Так же отправляется статус выполнения задачи и время сделанной фотографии.
9. Я перехожу к выполнению следующей задачи (переход на шаг 3)
10. По итогам дня я завершаю рабочий сеанс и приложение фиксирует время завершения рабочего дня. Если в данный момент сервер не доступен, данные сохраняются и отправляются при следующем подключении к сети.