

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки/специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка мобильной версии информационной системы управления
заявками в отделе АСУ»

Обучающийся

Т. Е. Братченко

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.п.н., доцент Е.А. Ерофеева

(ученая степень, звание, И.О. Фамилия)

Тольятти 2024

Аннотация

Выпускная квалификационная работа на тему: «Разработка мобильной версии информационной системы управления заявками в отделе АСУ».

Выпускная квалификационная работа состоит из введения, трех глав, списка литературы. Во введении обоснована актуальность выбранной темы, поставлена цель и задачи исследования. В первой главе «Анализ объекта исследования» представлены: описание организации, цели и задачи организации, организационная структура, функциональная модель, лучшие практики в предметной области, анализ существующих разработок для решения обозначенной задачи, функциональные требования к проекту, бизнес-цели. Во второй главе «Архитектура проекта и особенности реализации» представлены: описание информационной модели, концептуальная схема взаимодействия информационных систем, описание алгоритмов обработки данных, описание разработки интерфейса, описание базы данных и запросы, инструкции для всех категорий пользователей, примеры реализации проекта, тестирование ПО, результаты работы приложения. В третьей главе представлено экономическое обоснование проекта по расчетам чистой приведенной стоимости и рентабельности инвестиций. В заключении приведены основные выводы, полученные в результате исследования и разработки мобильного программного обеспечения.

Работа содержит 63 страницы текста, 5 таблиц, 41 рисунок.

Оглавление

Введение.....	5
Глава 1 Анализ объекта исследования.....	7
1.1 Описание организации.....	7
1.2 Структуры бизнес-процессов подразделения.....	10
1.3 Лучшие практики в предметной области.....	13
1.3.1 1С ИТILIUM.....	13
1.3.2 ITSM 365.....	15
1.4 Функциональные требования проекта.....	16
1.5 Бизнес-цели и требования ИТ-проекта.....	17
1.6 Описание проекта.....	18
Глава 2 Архитектура проекта и особенности реализации.....	19
2.1 Архитектура.....	19
2.2 MVVM.....	21
2.3 Model.....	22
2.4 Описание бизнес логики приложения.....	23
2.4.1 Авторизация.....	23
2.4.2 Запрос заявок.....	26
2.4.3 Работа с заявками.....	30
2.4.4 Поиск заявок в локальной базе.....	32
2.4.5 База данных.....	34
2.4.6 Уведомления.....	35
2.4.7 Интерфейс.....	38
2.4.8 1С НТТР-сервис.....	41
2.5 Тестирование ПО.....	42
2.5.1 Авторизация.....	42
2.5.2 Назначить ответственного.....	43
2.5.3 Закрывать заявку.....	45
2.5.4 Принять заявку.....	46

2.5.5 Гибкость интерфейса.....	47
2.6 Руководство пользователя.....	48
2.6.1 Введение	48
2.6.2 Назначение.....	48
2.6.3 Системные требования.....	49
2.7 Описание процессов	49
2.7.1 Регистрация пользователя.....	49
2.7.2 Установка мобильного ПО	50
2.8 Описание операций.....	51
2.8.1 Просмотр заявок.....	51
2.8.2 Управление заявками	51
2.8.3 Переслать заявку	52
2.8.4 Поиск заявок.....	53
2.8.5 Выход из аккаунта	53
2.8.6 Аварийные ситуации	54
Глава 3 Экономическое обоснование проекта	55
3.1 Расчет чистой приведенной стоимости – NPV	56
3.2 ROI (Return on Investment).....	57
Заключение	59
Список используемой литературы	60

Введение

В современном мире большое количество организаций и предприятий осознают необходимость внедрения эффективных систем управления заявками для обеспечения более оперативного и качественного обслуживания своих клиентов [1], [2]. Отдел автоматизированной системы управления (АСУ) играет ключевую роль в этом процессе, поскольку он отвечает за разработку, внедрение и поддержку таких систем. Однако, с развитием мобильных технологий и увеличением числа пользователей мобильных устройств, возникает необходимость разработки мобильной версии информационной системы управления заявками. Это позволит пользователям получать доступ к системе независимо от места и времени, что значительно повысит уровень сервиса и удовлетворенность клиентов.

Выбор данной темы обусловлен ее актуальностью и значимостью для современных организаций. На данный момент мобильные приложения стали очень популярны среди пользователей и предоставляют им возможности взаимодействия с разными информационными системами, в том числе и системой управления заявками через мобильное устройство. Развитие мобильного направления является необходимым шагом для современных организаций. Также разработка мобильной версии информационной системы управления заявками для отдела АСУ позволит повысить продуктивность работы отдела, ускорить выполнение, обработку заявок и сократить время реагирования на них. Приложение поможет снизить нагрузку на сотрудников и повысит их производительность. Исследование по разработке мобильной версии информационной системы управления заявками имеет научную и практическую значимость. Научная значимость исследования заключается в возможности изучения и анализа методов и технологий разработки мобильных приложений, а также их применения в приложении по управлению заявками. Разработка мобильной версии информационной системы управления заявками требует понимания правил проектирования пользовательского интерфейса,

безопасности данных, интеграции с существующими системами управления [3], [12]. Исследование этой темы помогает расширить знания в области информационных технологий. Практическая значимость заключается в результатах, которые могут быть применены в реальной деятельности организаций. Разработка мобильной версии информационной системы управления заявками поможет усовершенствовать обслуживание клиентов, сократить время реагирования на заявки, повысить эффективность работы отдела АСУ и снизить нагрузку на сотрудников, что будет являться улучшением конкурентоспособности компании и повышением ее репутации на рынке. Исследование по разработке мобильной версии информационной системы управления заявками для отдела АСУ имеет как научную, так и практическую значимость, помогает расширить знания в области информационных технологий, а также может быть применено для улучшения работы организаций и повышения конкурентоспособности организации.

Данная тема является актуальной для современных предприятий, так как помогает улучшить качество обслуживания клиентов и эффективность работы отдела АСУ. Разработка мобильной версии информационной системы является перспективным направлением развития информационных технологий организации и помогает развитию цифровой экономики.

Глава 1 Анализ объекта исследования

1.1 Описание организации

Государственное бюджетное учреждение здравоохранения «Санкт-Петербургский клинический научно-практический центр специализированных видов медицинской помощи (онкологический)» [5].

Учреждение было создано в результате выделения из Санкт-Петербургского «Городского клинического онкологического диспансера» и находится в ведении Комитета по здравоохранению, осуществляющего координацию его деятельности. Онкологический центр работает с августа 2012 года. Амбулаторно-консультативное отделение принимает, порядка 350 пациентов в день. На стационарных отделениях ежемесячно получают специализированную медицинскую помощь более 1500 пациентов. В центре работают 13 операционных, с установленной техникой европейского уровня.

Ведущие сотрудники учреждения являются специалистами и экспертами городского, российского и международного уровня. В настоящий момент 51 сотрудник преподает на разных кафедрах высших медицинских учебных заведений Санкт-Петербурга. Большинство врачей, мед. сестер, врачей и других специалистов постоянно проходят повышение квалификации в ведущих медицинских центрах России и всего мира.

ГБУЗ «СПб КНпЦСВМП(о)» решает следующие основные задачи:

- оказание специализированной, медицинской помощи больным с онкологическими и предопухолевыми заболеваниями;
- разработка и организационно-методическое обеспечение мероприятий, направленных на совершенствование профилактики и раннего выявления онкологических и предопухолевых заболеваний, осуществляемых на территории Санкт-Петербурга;
- анализ состояния онкологической помощи населению, эффективности и качества профилактических мероприятий, диагностики, лечения и

наблюдения за онкологическими больными на территории Санкт-Петербурга;

- изучение, исследование и внедрение в практику современных методов выявления и лечения онкологии;
- участие в организации и проведении научно-практических мероприятий по проблемам онкологии;
- организация работы по повышению онкологической настороженности врачей и средних медицинских работников лечебно-профилактических учреждений;
- методическое руководство и консультативная помощь врачам специалистам Санкт-петербургских государственных учреждений здравоохранения по вопросам проведения профилактических осмотров, диспансеризации больных с предопухолевыми и хроническими заболеваниями;
- методическое руководство и консультативная помощь по вопросам санитарно-гигиенического просвещения среди населения по профилактике онкологических заболеваний, соблюдению здорового образа жизни;
- ведение учетной и отчетной документации, представление отчетов о деятельности в установленном порядке, сбор данных для регистров, ведение которых предусмотрено законодательством.

В центре работает отдел автоматизированных систем управления, который является самостоятельным структурным подразделением предприятия.

С точки зрения распределения полномочий, производственных обязанностей организации подразделение можно классифицировать по таким типам управления: с линейной организацией управления; с функциональной организацией управления; с линейным и функциональным управлением; с матричной системой управления; с использованием комитетов (комиссий).

Отдел автоматизированных систем управления (АСУ) - формальная, централизованная структура с плоской системой управления. Он состоит из руководства и секторов.

В организации выполняет следующие функции:

- Разработка, установка и обслуживание информационных систем: отдел АСУ занимается разработкой, установкой и обслуживанием программного обеспечения, которое помогает автоматизировать бизнес-процессы и улучшить эффективность работы организации.
- Поддержка информационных систем: отдел АСУ следит за тем, чтобы все системы работали без сбоев и были доступны сотрудникам организации в любое время.
- Осуществляет контроль за работоспособностью систем: отдел АСУ следит за работоспособностью информационных систем, выявляет и устраняет возможные проблемы, связанные с их функционированием.
- Защита информации: отдел АСУ обеспечивает безопасность данных организации, предотвращает утечки информации, защищает системы от вирусов и угроз.
- Повышение производительности: отдел АСУ занимается оптимизацией процессов работы организации, автоматизацией рутинных задач, сокращением времени на выполнение задач.
- Внедрением новых технологий: отдел АСУ следит за развитием ИТ, исследует новые технологии, которые могут быть полезны для работы организации.

Организационная схема центра представлена на рисунке 1.

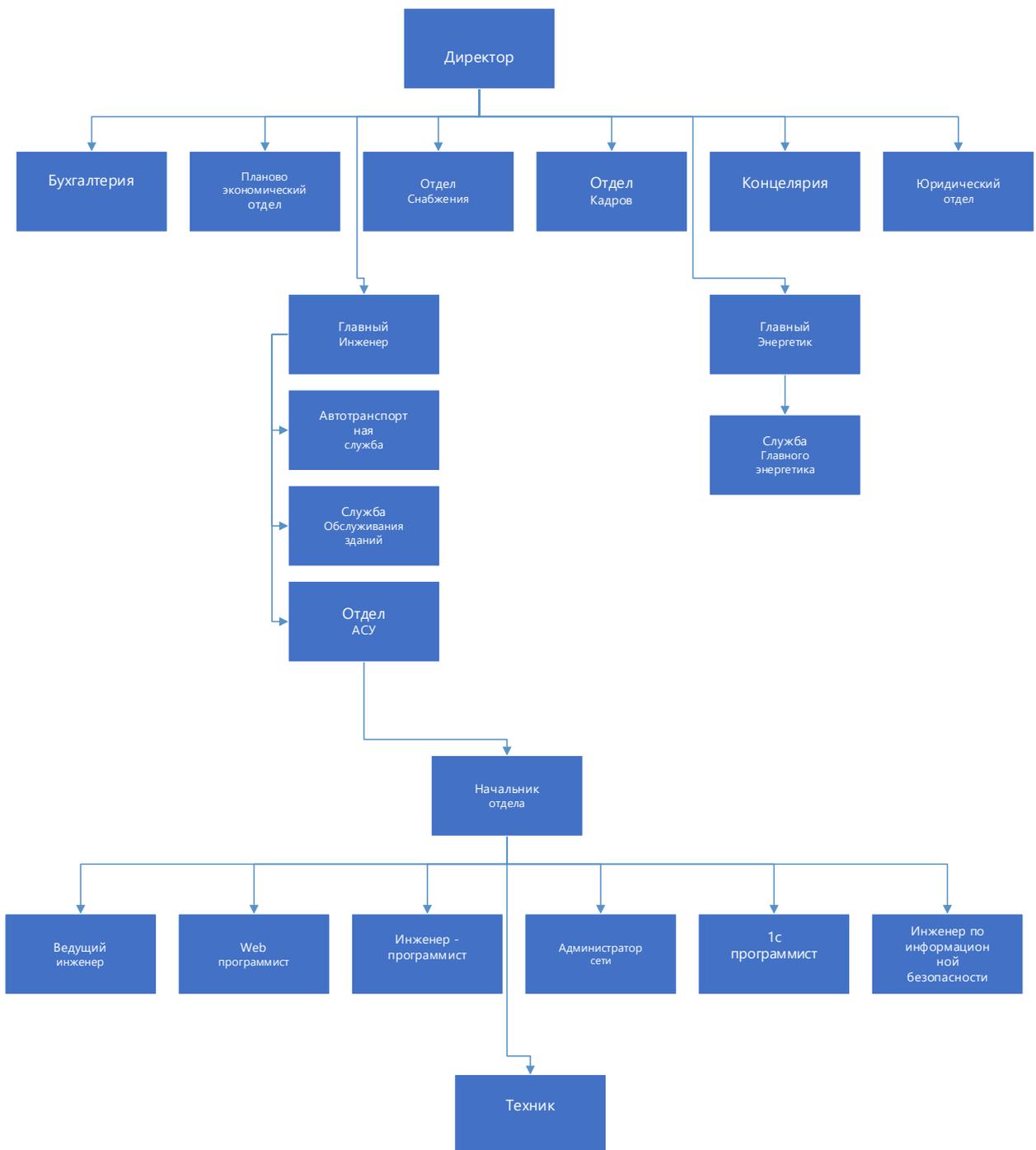


Рис. 1 - Организационная диаграмма центра

1.2 Структуры бизнес-процессов подразделения

Для определения требований и указания функций для последующей разработки системы, отвечающей поставленным требованиям и реализующей выделенные функции, используются диаграммы IDEF0.

Диаграмма IDEF0 AS- IS отдела АСУ представлена на рисунке 1 и 2.

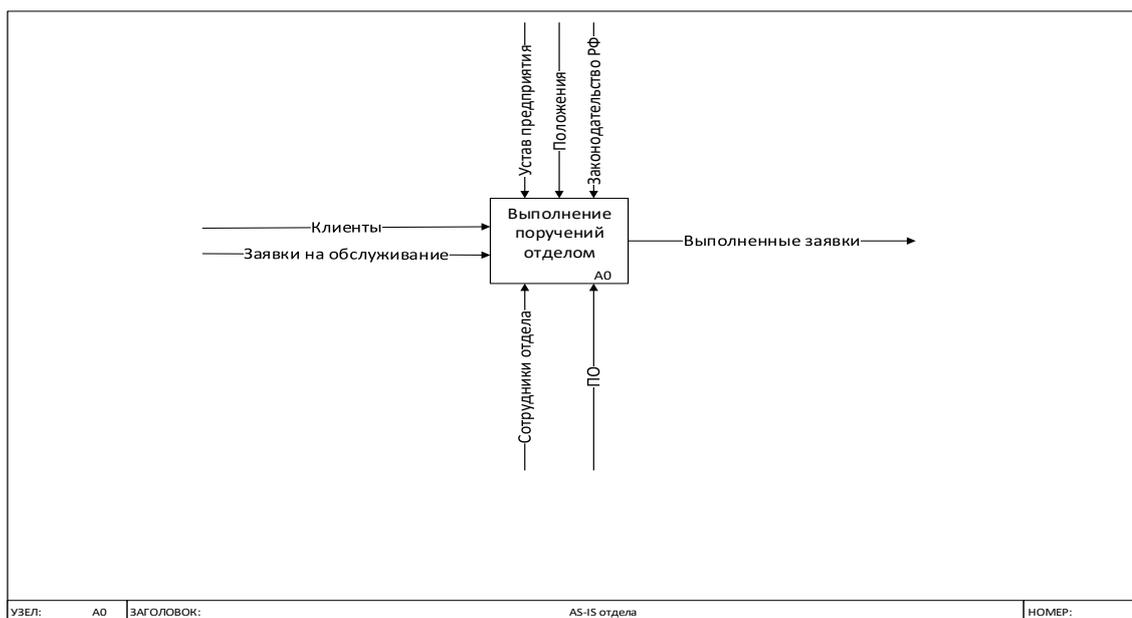


Рис. 2 - Представление контекстной диаграммы (AS-IS)

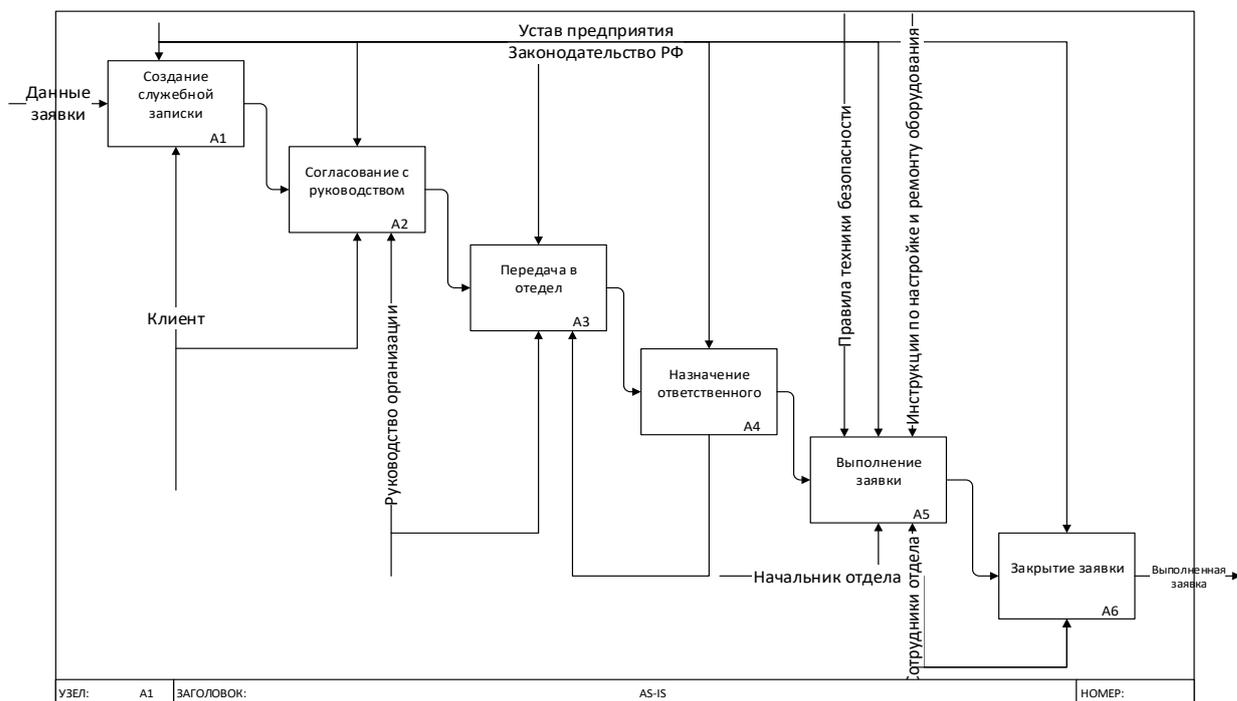


Рис. 3 - Представление диаграммы первого уровня

На данной диаграмме представлены все процессы от входных данных и создания заявки до уже выполненной заявки. Клиент, руководствуясь уставом

предприятия и законодательством РФ, создает служебную записку вручную, вносит необходимые данные (данные о ремонте и настройке оборудования) и относит на согласование руководству организации. Затем начальник отдела забирает согласованную служебную записку и назначает ответственного за выполнение данной заявки, руководствуясь уставом предприятия и законодательством РФ. Ответственные за выполнение заявки сотрудники выполняют поручение руководствуясь правилами техники безопасности и инструкциями по настройке, ремонту, установки, программированию и т.д. оборудования. После доклада о выполненной работе, начальник отдела закрывает заявку. Заявка приобретает статус выполнения.

Сделав анализ можно увидеть, что процессы создания, согласования, передачи и назначения ответственного можно сократить, ускорить и облегчить с помощью интеграции информационной системы, таким образом сэкономив траты рабочего времени на данные процессы.

Для определения требований и указания функций так же используем диаграмму IDEF0. Диаграмма ТО-ВЕ представлена на рисунке 3 и 4.

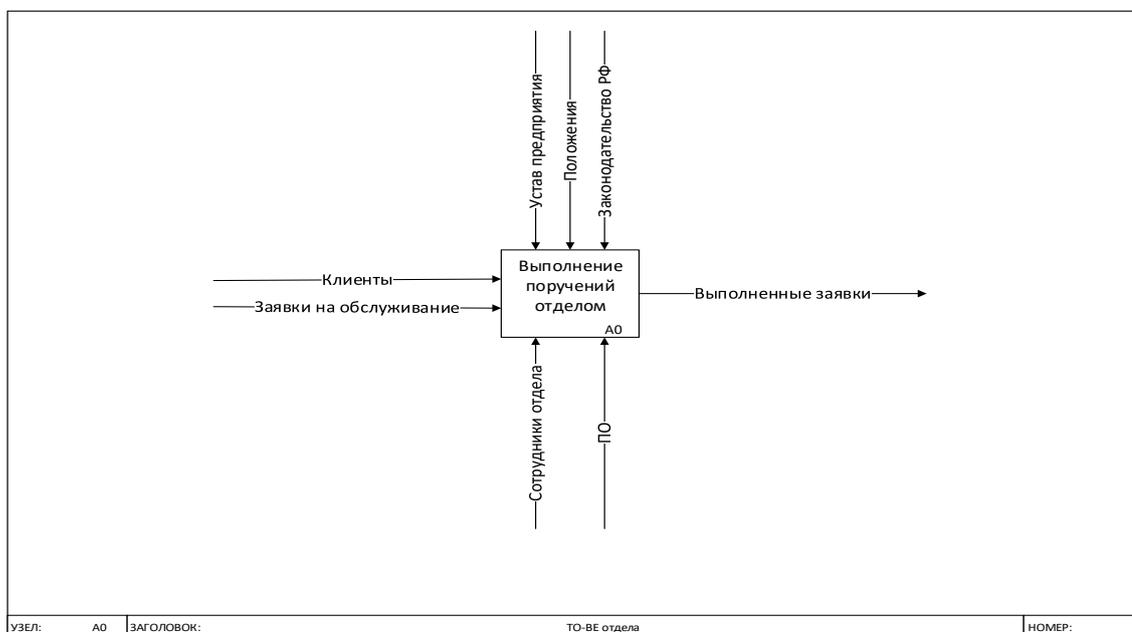


Рис. 3 - Представление контекстной диаграммы (ТО-ВЕ)

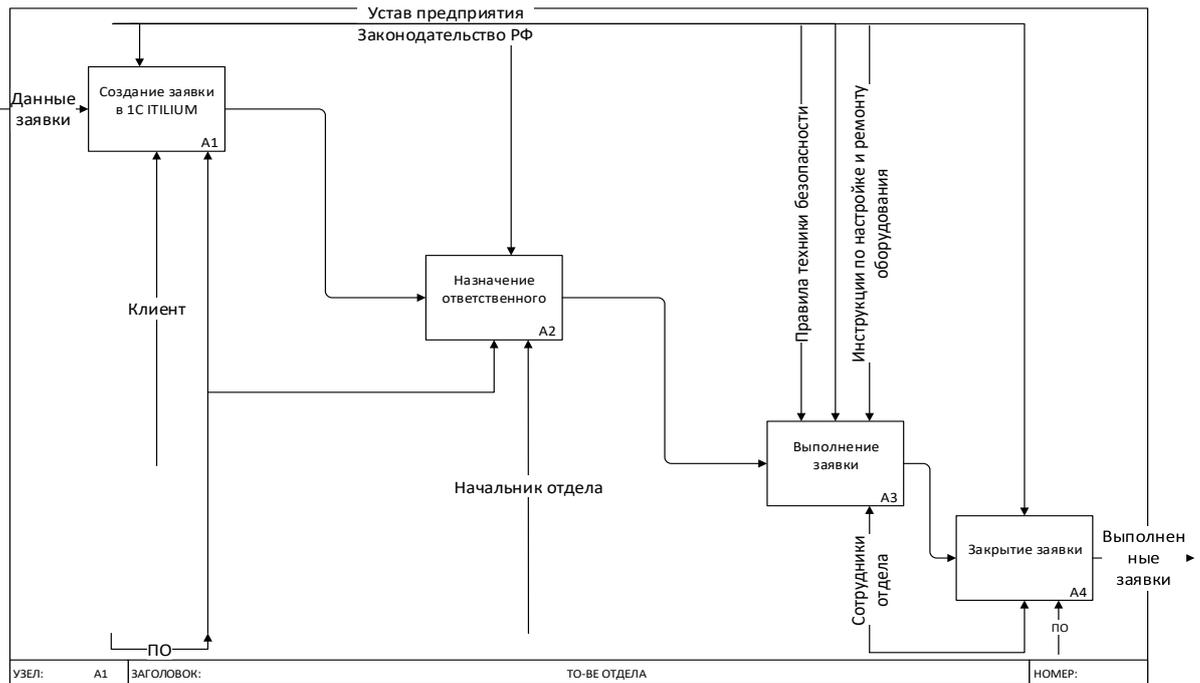


Рис. 4 - Представление контекстной диаграммы (ТО-ВЕ)

По данной диаграмме видно, что общая схема сократилась и множество ручной работы заменено программным обеспечением. Для решение этих задач было выбрано программное обеспечение 1С ITILUM, а для большей оптимизации работы данных процессов необходимо создать мобильное приложение. Таким образом сотрудники смогут получать и закрывать заявки не только в отделе где стоит компьютер с программой 1С ITILUM, но и в любом другом месте где имеется интернет соединение.

1.3 Лучшие практики в предметной области

1.1.1 1С ITILUM

Для работы с обращениями, изменениям, событиями и т.д. в организации используется конфигурация 1С ITILUM [10], [11].

1С ITIL IUM предназначена для автоматизации управления услугами как в области информационных технологий, так и в других функциональных областях.

Решение помогает управлять подразделением, организовать работу с обращениями клиентов и сотрудников, вести учет различного оборудования и ПО, нужного для оказания услуг.

Основные преимущества решения:

- повышение прозрачности и эффективности процессов управления услугами;
- повышение качества оказываемых услуг и удовлетворенности потребителей услуг;
- обеспечение эффективной совместной работы сотрудников сервисных подразделений;
- возможность создания системы самообслуживания (Self-Service) – пользователи могут решить свой вопрос, пользуясь инструкциями и подсказками, не обращаясь в службу поддержки;
- сокращение простоев сервисных активов и других конфигурационных единиц.

Функциональные возможности продукта:

- Управление обращениями: регистрация обращений (инцидентов и запросов) и их учет, обработка обращений в рамках SLA, организация поступления обращений через единую точку контакта, распределение загрузки и контроль деятельности сотрудников сервисных служб, инструментарий для автоматической регистрации и оперативного управления обращениями.
- Управление работами: учет нарядов на выполнение работ, планирование ресурсов и управление ими в разрезе услуг и клиентов.
- Управление событиями: контроль и анализ событий, влияющих на предоставляемые услуги, предупреждение инцидентов, оперативное

информирование о случившихся инцидентах.

1.1.2 ITSM 365

Заменить 1С итилиум возможно используя help desk сервисы. ITSM 365 – SaaS-сервис для среднего и малого бизнеса, который нужен для управления поступающими заявками от пользователей. Управление облачными приложениями клиентов реализовано через единый портал. В портале хранится вся история взаимодействия с клиентом: общая информация, тарифы, лицензии, счета и т.д. Чтобы расширить возможности контроля за функционированием приложений и обеспечить высокий уровень доступности сервисов, была проведена интеграция портала ITSM 365 с интеллектуальной системой мониторинга на базе платформы Naumen DAP.

Основная задача инструмента — регистрация и дальнейшее отслеживание исполнения заявок пользователей, поступивших по всем каналам связи, с учетом трудозатрат исполнителей. Кроме того, сервис позволяет вести базу знаний и каталог услуг, определяя, в том числе, и уровень предоставления каждой из них (SLA), т.е. сроки реакции по различным типам заявок. Система позволяет контролировать сроки, оставшиеся по SLA, и уведомляет исполнителей и других заинтересованных лиц при приближении к окончанию регламентного срока. А с точки зрения бизнеса этот уровень контролируется посредством встроенной системы отчетности, позволяющей выгружать данные во внешние системы для последующего поиска «узких» мест в обслуживании.

Конфигурация 1С Итилиум и help desk сервисы в сравнении имеют очень похожий функционал [4], [10], [11]. Но выбор пал на конфигурацию 1С итилиум, потому что эта конфигурация более дешевая по стоимости лицензии, чем стоимость help desk сервисов, а также 1С конфигурацию можно обслуживать и дополнять самостоятельно, что сократит время решения какой-либо проблемы возникшей в ходе работы программы или ее дополнения новыми функциями.

1.4 Функциональные требования проекта

Функциональные требования – это тип требований, который определяет функции, возможности и функции, которыми должна обладать система или продукт для удовлетворения потребностей своих пользователей. Эти требования описывают, что должна делать система, а не как она должна это делать. Обычно они выражаются в терминах конкретных вариантов использования, описывающих поведение системы в различных сценариях.

Функциональные требования часто используются в проектах разработки программного обеспечения, где они предоставляют подробную спецификацию поведения программного обеспечения. Однако они также могут использоваться в других типах проектов, таких как разработка продукта или инициативы по улучшению процессов.

Функциональные требования разрабатываемого мобильного приложения:

- Управление заявками: просмотр информации по заявкам; отслеживание статуса заявки; назначение ответственного/принятие заявки/закрытие заявки
- Авторизация
- Поиск заявок по локальной базе
- Получение уведомлений: о новых заявках для пользователя и администратора; о взятии в работу заявки для администратора; закрытии заявки для администратора
- Распределение ролей.
- Функция «Поделиться» заявкой в сторонней программе (мессенджеры, соц. сети и т.д.)
- Совершение звонка по номеру телефона в приложении

1.5 Бизнес-цели и требования ИТ-проекта

По функциональным требованиям можно определить бизнес цели разрабатываемого мобильного ПО, а именно:

- Более быстрое выполнение задач по заявкам за счет того, что сотрудник может отреагировать на заявку в месте, где выполняется последняя задача.
- При авторизации с помощью алгоритма шифрования SHA-1 обеспечивается безопасность и конфиденциальность данных пользователя.
- Функции «Поделиться» помогает быстрее привлекать нужного сотрудника к выполнению задачи.
- Обратная связь с клиентом через функцию звонка по клику на номер телефона в заявке.

По бизнес целям можно определить тех. требования разрабатываемого мобильного ПО, а именно:

- Запросы в базу данных 1С через сервис 1С для получения информации о заявках, обновлении данных по заявкам.
- Авторизация должна быть безопасной для данных пользователей, защита данных при работе через сеть и хранение данных.
- Просмотр, редактирование, назначение ответственного, закрытие заявок с функцией привязки комментария после выполнения задачи.
- Уведомления о новых заявках, изменении данных по заявкам и т.д. с помощью push-уведомлений.
- Поиск заявок по любым реквизитам заявки в локальной базе данных.
- Возможность проводить различные операции с заявками в приложении без доступа к интернету.

1.6 Описание проекта

Мобильное ПО разработано под операционную систему Android и работает с опубликованным 1С сервисом в локальной сети организации, который необходим для обмена данными с базой данных конфигурации Итилиум. Цель проекта сократить время на выполнение задачи сотрудником отдела АСУ, сделать взаимодействие сотрудника с клиентом более удобным, обеспечить доступ к заявкам из любого места, где есть соединение с интернетом. Приложение предоставляет пользователям возможность управлять заявками, через сервис 1С конфигурации с любых мобильных устройств. Пользователи могут просматривать информацию по заявкам, отслеживать статус заявки, назначать ответственных, принимать в работу и закрывать выполненные заявки. Передача данных из приложения на сервис и обратно, осуществляется с помощью http-запросов на опубликованный http-сервис 1С конфигурации. Концептуальная схема взаимодействия мобильного приложения и 1С сервиса представлена на рисунке 1.

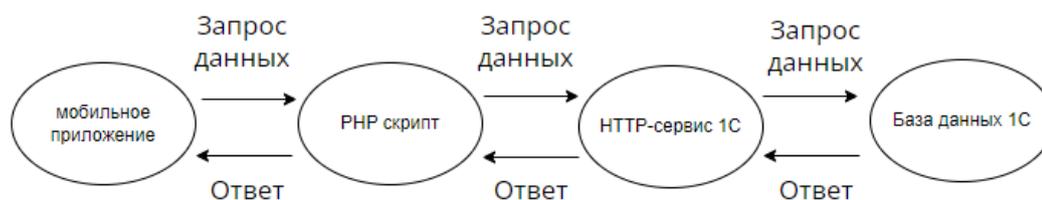


Рисунок 1 – Концептуальная схема взаимодействия

База данных 1С находится в локальной сети организации, также сервис публикуется в локальной сети, поэтому было принято решения добавить PHP скрипт на сервер с фиксированным ip адресом, на котором находится сайт организации, для перенаправления данных на HTTP-сервис 1С чтобы реализовать возможность делать запросы не только в локальной, но и в глобальной сети.

Глава 2 Архитектура проекта и особенности реализации

2.1 Архитектура

Приложение, разработанное под Android, содержит такие компоненты, как: действия; фрагменты; службы; поставщики контента; приемники широковещательных сообщений. Многие из этих компонентов прописываются в конфигурационном файле приложения.

Операционная система использует этот файл для правильной интеграции этих компонентов, в том числе и в интерфейс. Пользователи могут использовать в системе множество приложений, постоянно перемещаясь между ними, поэтому это необходимо учесть при разработке т.е. сделать приложение более гибким чтобы предотвратить ошибки во время многозадачного использования системы.

Некоторые компоненты регулируются системой самостоятельно в зависимости от оставшейся памяти или наличия интернет-соединения и т.д., то есть система сама принимает решение об остановке и удалении из оперативной памяти, к примеру фоновых задач, и т.д. Поэтому рекомендуется все необходимые данные сохранять в постоянной памяти устройства или в локальной базе данных.

Нужно правильно определить архитектуру, которая позволит приложению масштабироваться, повысит его надежность и упростит его тестирование [12].

Для правильной работы мобильного ПО, приложение должно иметь как два уровня:

- Уровень UI, отображающий данные на экране.
- Уровень данных, который содержит бизнес-логику приложения и предоставляет данные приложения.

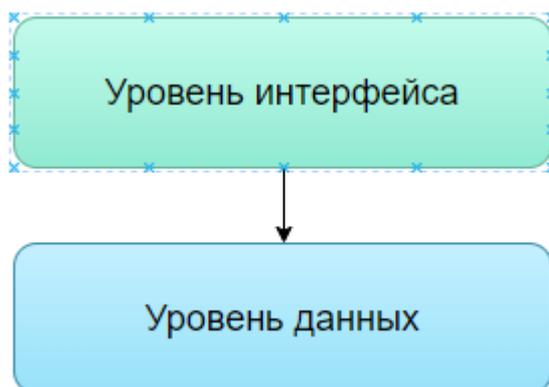


Рисунок 2 – Общая схема архитектуры

Интерфейс необходим для облегчения взаимодействия пользователя с приложением, а также должен быть гибким и быстро и правильно реагировать на определенные события, происходящие в приложении, будь то получение данных из локальной базы или взаимодействие с внешними сервисами.

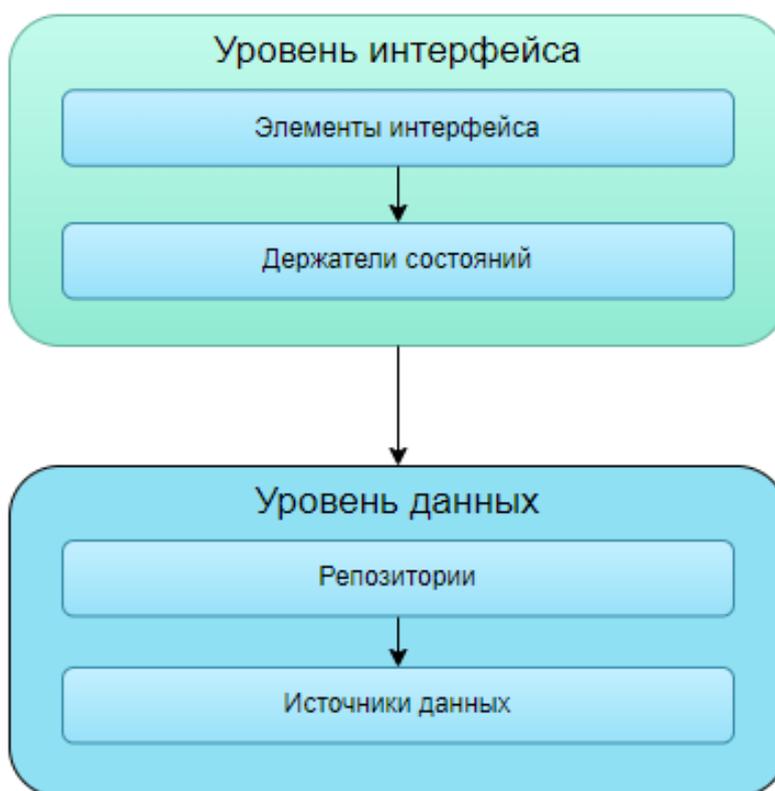


Рисунок 3 – Общая схема уровней архитектуры.

Интерфейс состоит из:

- Множества различных элементов для взаимодействия с пользователем.
- Держателей состояний (ViewModel), которые содержат в себе данные, передают их интерфейсу и обрабатывают какую либо логику приложения.

Уровень данных приложения содержит бизнес-логику и состоит из репозиториев, в которых может быть много источников данных.

2.2 MVVM

MVVM — это архитектура мобильного приложения, которая делит приложения на три необходимых уровня [3], [12].

При использовании MVVM в мобильном приложении нужно начинать с разработки уровня бизнес-логики (Model) перед уровнем интерфейса (View и ViewModel).

- Бизнес-логика включает в себя обработку данных, взаимодействие с сервером, хранение и обработку информации. Определение бизнес-логики позволяет понять, какие данные и функции будут реализованы в мобильном приложении.
- ViewModel будет представлять данные и логику, необходимую для их отображения и обработки в пользовательском интерфейсе.
- После того как созданы Model и ViewModel, разрабатывается пользовательский интерфейс. Интерфейс отображает данные, полученные из ViewModel, и обрабатывает пользовательский ввод.

Начало разработки с бизнес-логики позволяет разработчикам четко определить функциональность приложения, структуру данных и логику работы приложения до того, как приступить к созданию пользовательского интерфейса. Это помогает более эффективно интегрировать бизнес-логику с интерфейсом. Поэтому именно эта архитектура была выбрана для создания проекта.

Таким образом, общая схема взаимодействия компонентов будет выглядеть следующим образом (рисунок 4).

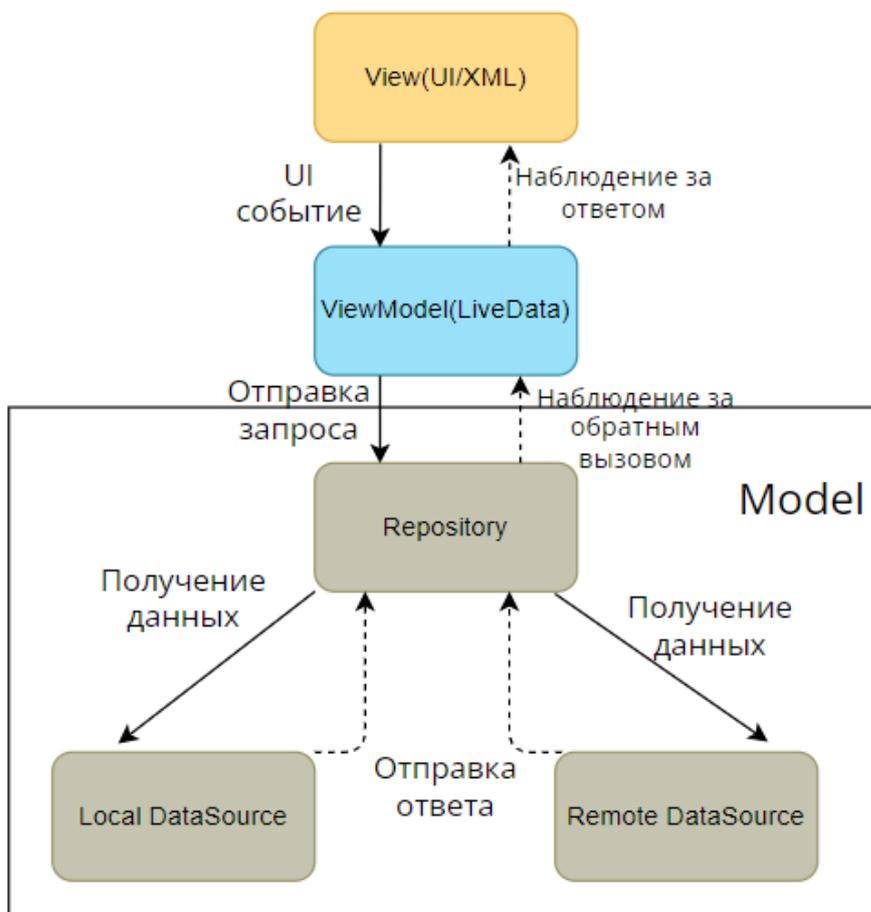


Рисунок 4 – Общая схема взаимодействия компонентов

2.3 Model

В модели андроид приложения, репозиторий (repository) обычно представляет собой слой абстракции между источниками данных (например, базой данных, сетью) и остальными компонентами приложения. Репозиторий отвечает за доступ к данным, их обработку и предоставление другим частям приложения.

Репозиторий позволяет изолировать остальные части приложения от конкретных источников данных, что упрощает тестирование и поддержку

приложения. Он также может содержать логику для кэширования данных, обработки ошибок и синхронизации данных между различными источниками.

Использование репозитория в модели андроид приложения помогает создать более чистую и модульную архитектуру приложения, улучшая его расширяемость и поддерживаемость.

В проекте реализована локальное хранилище как SQLite реляционная база данных, работа с которой происходит с помощью библиотеки ROOM. Библиотека ROOM предоставляет уровень абстракции поверх SQLite, обеспечивающий свободный доступ к базе данных, используя при этом всю мощь SQLite.

2.4 Описание бизнес логики приложения

2.4.1 Авторизация

В мобильном приложении предусмотрена авторизация. Первый раз запуская приложение, пользователь видит splash screen (Рисунок 7) на время проверки данных наличия данных авторизации и обработки, а также выполняется проверка на наличие Интернет-соединения. После всех проверок открывается экран авторизации в том случае, если данные авторизации не найдены. Пользователь выбирает свое ФИО и вводит пароль. После нажатия кнопки «Войти», происходит проверка подключения к интернету, если есть доступ к интернету, то пароль хэшируется по алгоритму SHA1, ФИО и хэш-сумма пароля записываются в json строку, иначе появляется всплывающее сообщение об отсутствии интернета (Рисунок 5).

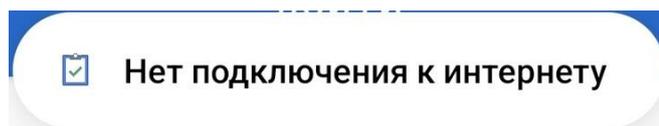


Рисунок 5 – Всплывающее сообщение

Далее формируется запрос, в тело которого помещается созданная json строка, на PHP скрипт на сервере, где запрос перенаправляется на HTTP-сервис 1С.

HTTP-сервис принимает запрос, проверяет правильность введенных данных и возвращает ответ. Если ответ равен «true», то мобильное приложение записывает данные авторизации и делает HTTP-запрос на получение заявок и выводит заявки на экран, иначе появляется всплывающее сообщение «Неверные данные авторизации» (Рисунок 8). Блок - схема алгоритма авторизации представлена на Рисунке 6. Интерфейс экрана авторизации представлен на Рисунке 7.

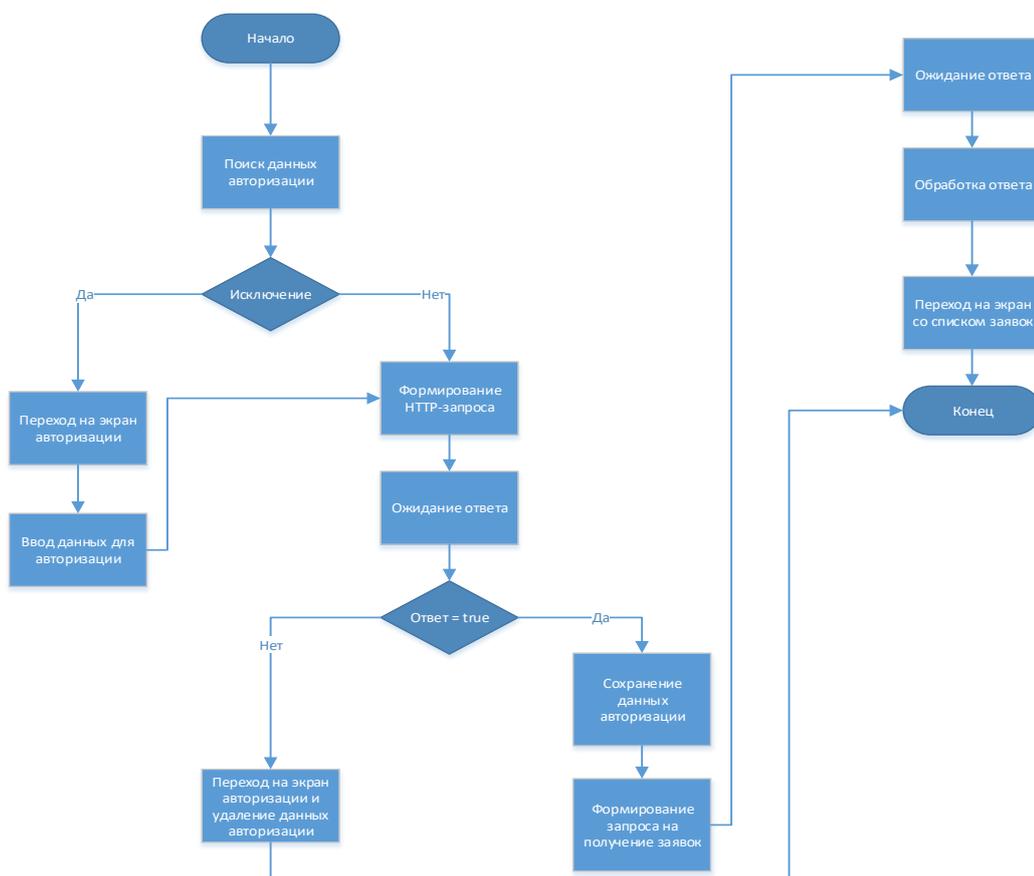


Рисунок 6 – Алгоритм авторизации в мобильном приложении.

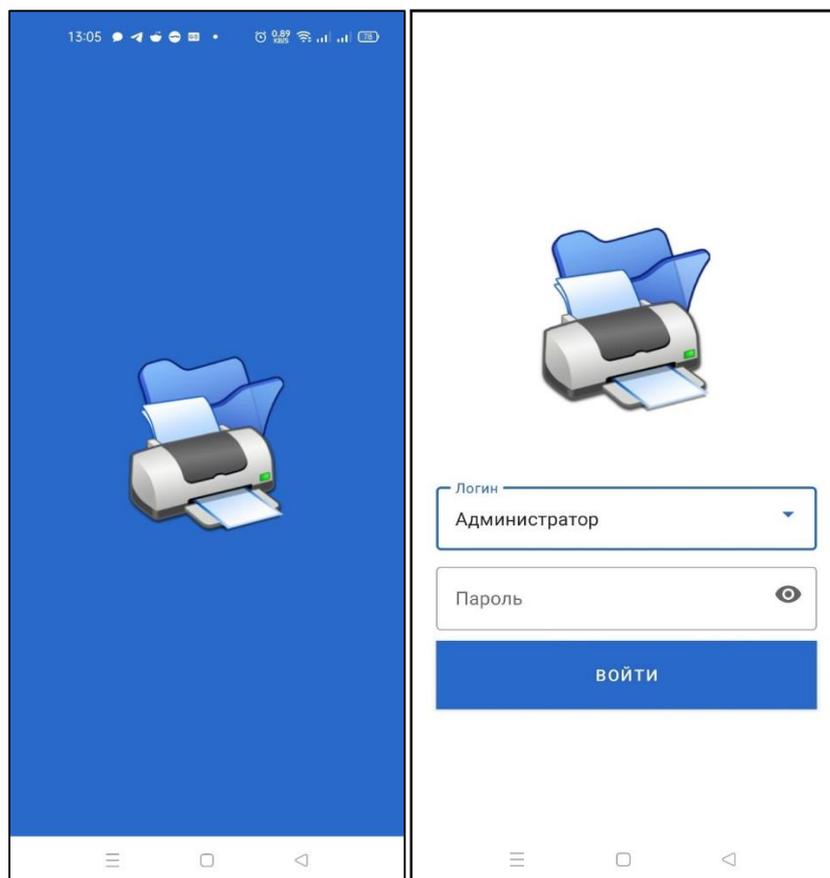


Рисунок 7 – Splash screen и экран авторизации



Рисунок 8 – Всплывающее сообщение

Также предусмотрена функция разлогирования пользователя. По нажатию кнопки «Выход» (Рисунок 9) на верхней панели приложения, удаляются данные авторизации, данные по заявкам и общие настройки приложения, выполняется запрос на удаление токена из базы 1С, далее осуществляется переход на экран авторизации.



Рисунок 9 – Кнопка «Выход»

2.4.2 Запрос заявок

Заявки регистрируются начальниками, старшими сестрами и заведующими отделениями в программе «1С Итилиум» для подразделения «АСУ» [10]. Начальник подразделения «АСУ» получает уведомление о регистрации новой заявки, выбирает ответственного за выполнение и сохраняет документ(заявку) в «1С Итилиум». После назначения ответственного, по нажатию кнопки «Записать и закрыть» заявка сохраняется в базе 1С.

← → ☆ Обращение №0000000578, зарегистрировано в 05.08.1 10:16:57, *

Записать и закрыть Записать

Номер:	0000000578	Дата:	05.08.2022 10:16:57	Состояние:	В работе
Клиент:	ККМТ	Ответственный:	Александр Павлович	Запрос обслуживания:	<input checked="" type="checkbox"/>
Инициатор:	Екатерина Сергеевна	Услуга:	Обслуживание и ремонт		
Место возникновения проблемы:	Б700				
Телефон для контакта:	89533671662				

★ Общие сведения 📄 Решение 📄 Общение [1]

Тема: Произвольная заявка

Описание:

Срочно!!! компьютеры вынесли. нужно установить в кабинете у заведующего!!! на отделении ремонт!!!
и пожалуйста сделайте мне доступ нормальный. постоянно глючит. пишет у пользователя нет права пользоваться этой программой!!!

Рисунок 10 – Форма заявки в 1С Итилиум

При наличии доступа в интернет, мобильное приложение выполняет запрос с телом в виде json строки с параметрами отбора заявок в базе данных

на PHP скрипт, расположенный на локальном web-сервере с фиксированным ip-адресом для возможности обращения к 1С HTTP-сервису из глобальной сети [8]. Блок – схема алгоритма отправки запроса в 1С из мобильного приложения представлена на Рисунке 11.



Рисунок 11 – Алгоритм отправки запроса в 1С.

Скрипт «index.php» получает json строку из тела запроса, перенаправляет запрос на http-сервис 1С, где выполняется запрос в базу

данных 1С для нахождения заявок по заданным параметрам. Далее выполняется проверка на количество заявок и если заявки не найдены, то в ответ на запрос отправляется сообщение об отсутствии заявок в базе данных, иначе циклично записывается строка json и отправляется ответом на запрос.

Мобильное приложение принимает ответ, обрабатывает его и выводит соответствующую иконку при отсутствующих заявках, либо выводит список заявок на экран (Рисунок 13, 14). Блок – схема алгоритма обработки ответа в мобильном приложении представлена на Рисунке 12.

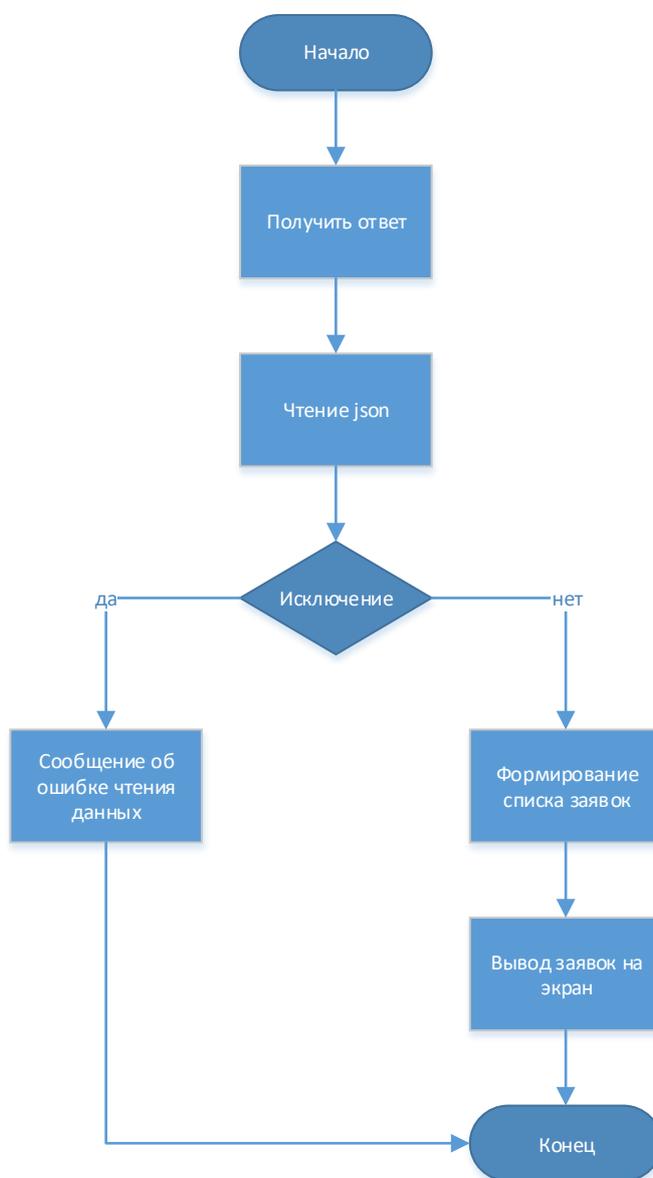


Рисунок 12 – Алгоритм обработки ответа в мобильном приложении.

При отсутствии Интернет-соединения выполняется поиск заявок в базе данных, что позволяет работать в офлайн режиме.

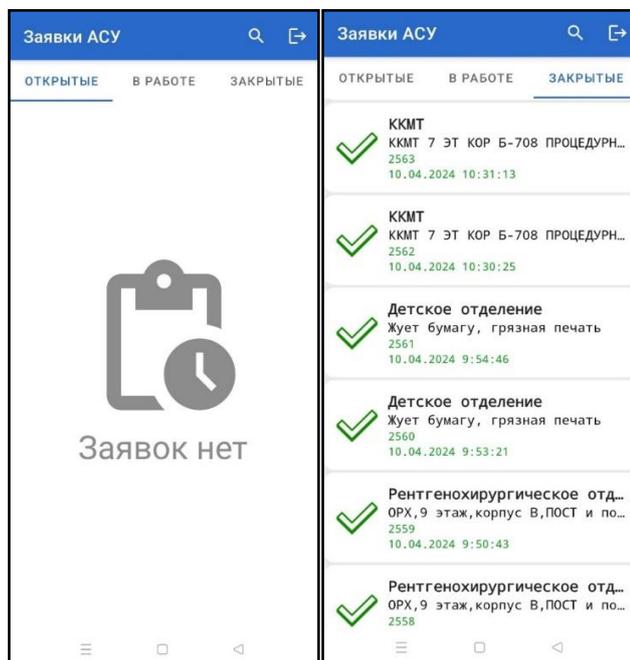


Рисунок 13 – Список заявок

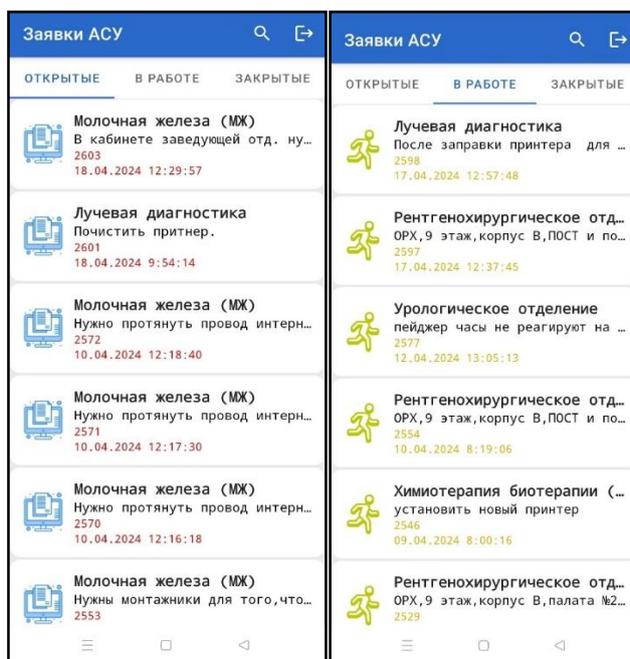


Рисунок 14 – Список заявок

2.4.3 Работа с заявками

При нажатии на заявку из списка заявок, открывается новая Activity с полным описанием заявки. В зависимости от роли авторизованного сотрудника и статуса выбранной заявки, на новой Activity появляются кнопки: «Назначить ответственного» с полем выбора, «Принять заявку», «Заккрыть заявку» с полем «Решение». (Рисунок 15)

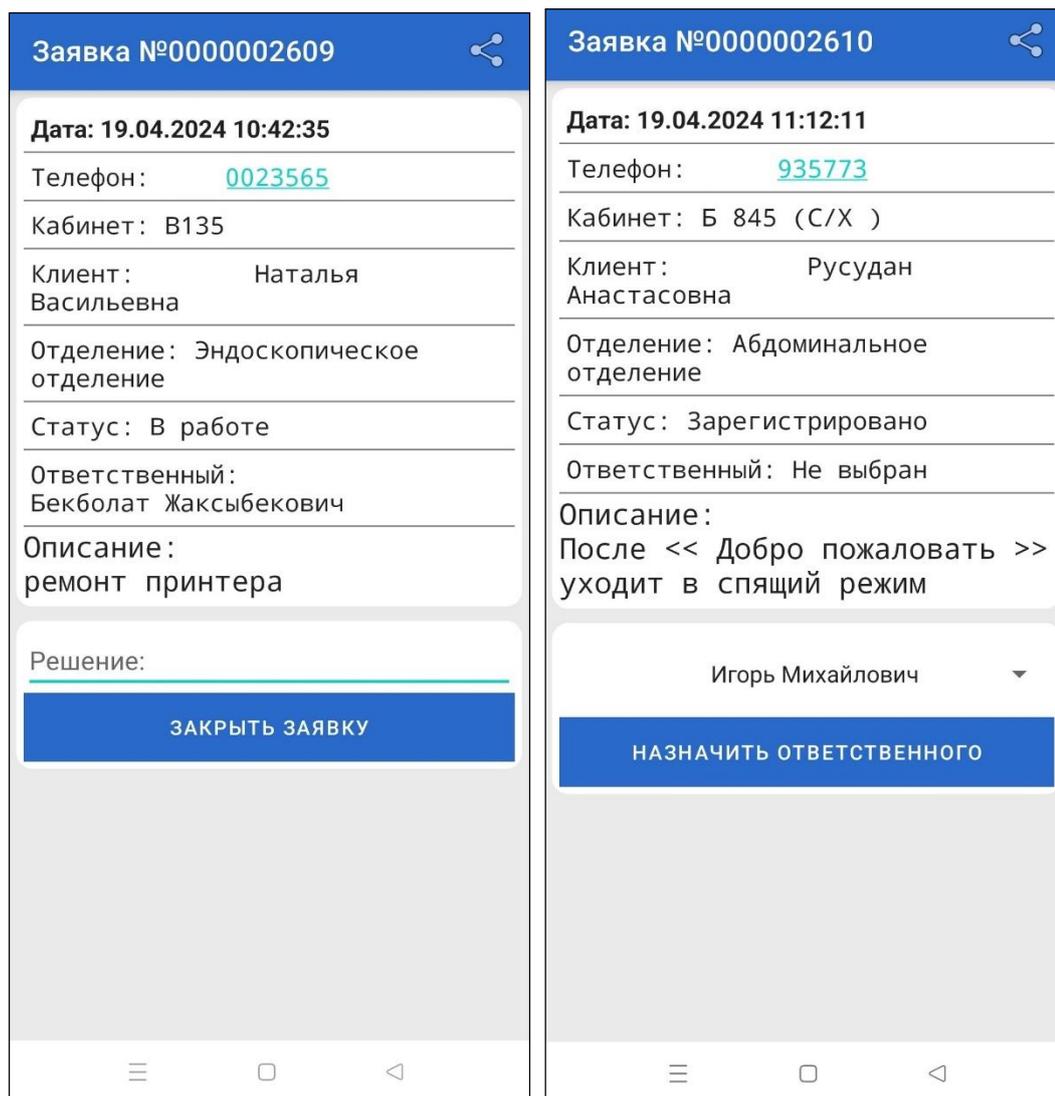


Рисунок 15 – Заявка

При нажатии кнопки «Назначить ответственного» или «Принять заявку», формируется json строка с данными ответственного, номером заявки,

статусом. Далее отправляется запрос из мобильного приложения на php скрипт, откуда запрос перенаправляется в HTTP-сервис. Функция сервиса делает запрос на изменение данных по заявке и меняет данные в базе 1С. После чего возвращает ответ о назначении ответственного или принятии заявки в работу или закрытии заявки, данные по заявке обновляются в локальной базе и экран с заявкой закрывается. При этом списки заявок по категориям обновляются при изменении данных в локальной базе приложения с помощью наблюдаемого класса держателя данных LiveData. В отличие от обычного наблюдаемого объекта, LiveData учитывает жизненный цикл, то есть учитывает жизненный цикл других компонентов приложения, таких как действия, фрагменты или службы. Эта осведомленность гарантирует, что LiveData обновляет только наблюдателей компонентов приложения, которые находятся в активном состоянии жизненного цикла.

Если при закрытии заявки поле «Решение» пустое, то появится сообщение о необходимости заполнить это поле (Рисунок 16).

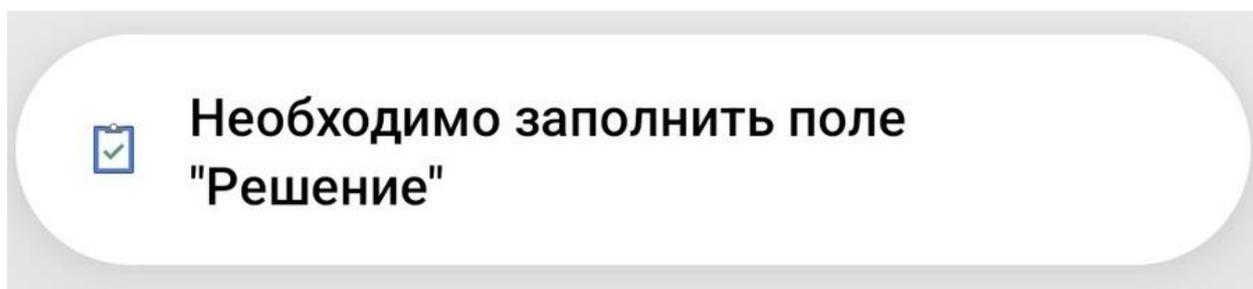


Рисунок 16 – Заявка

Если заявки нет в базе, то в ответе отправляется строка об отсутствии заявки. Если ответственный уже назначен, то возвращается в ответе соответствующее сообщение, которое выводится на экран. Также выводится на экран соответствующее сообщение при отсутствии Интернет-соединения. Блок – схема алгоритма принятия заявки представлен на Рисунке 17.

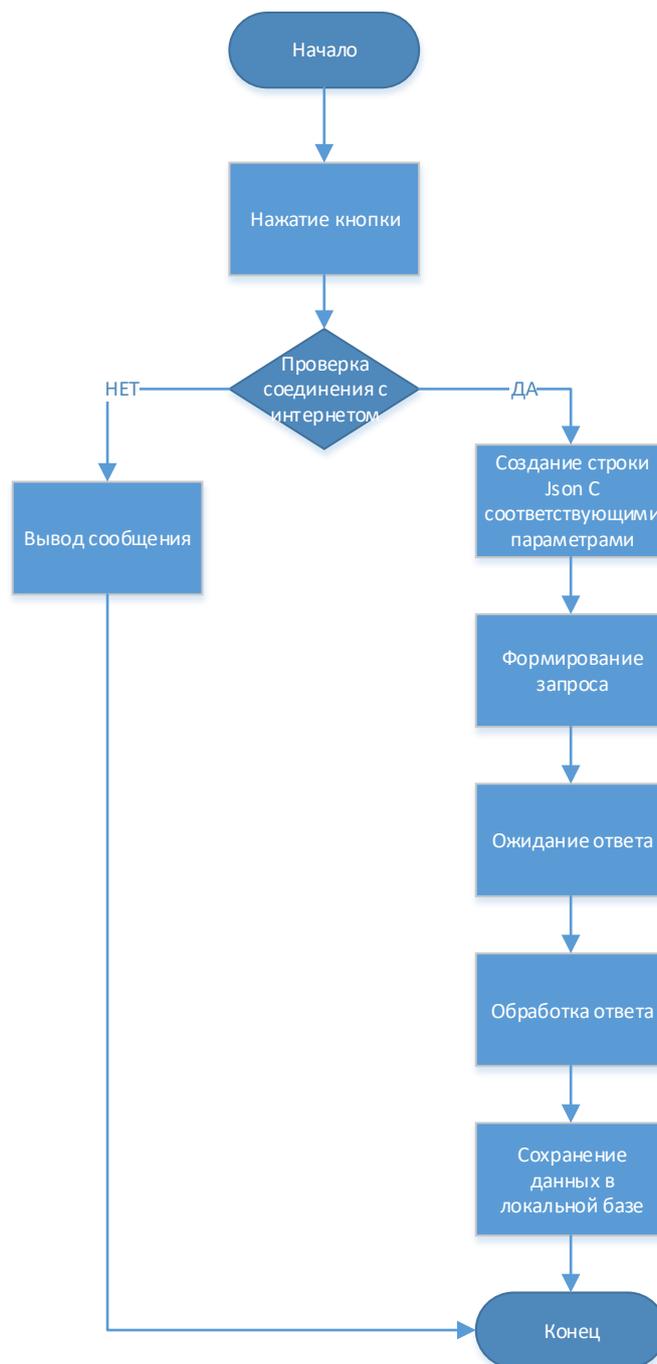


Рисунок 17 – Алгоритм принятия/назначения ответственного/закрытия заявки в мобильном приложении.

2.4.4 Поиск заявок в локальной базе

По нажатию кнопки поиск (Рисунок 18), которая находится в верхней панели приложения, открывается новый экран с виджетом SearchView в верхней панели приложения.

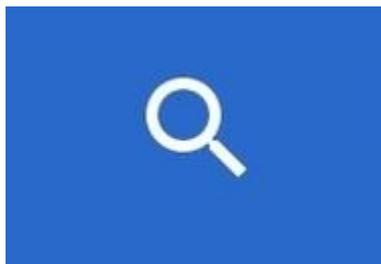


Рисунок 18 – Кнопка поиска

SearchView — это виджет, предоставляемый Android-фреймворком, который позволяет пользователям искать определённые данные в приложении. При вводе символа, каждый раз выполняется функция `OnQueryTextListener`, с помощью которой выполняется SQL запрос в базу данных приложения. Поиск происходит по любым данным из заявки. Список найденных обновляется на ходу с помощью `LiveData` класса (Рисунок 19).

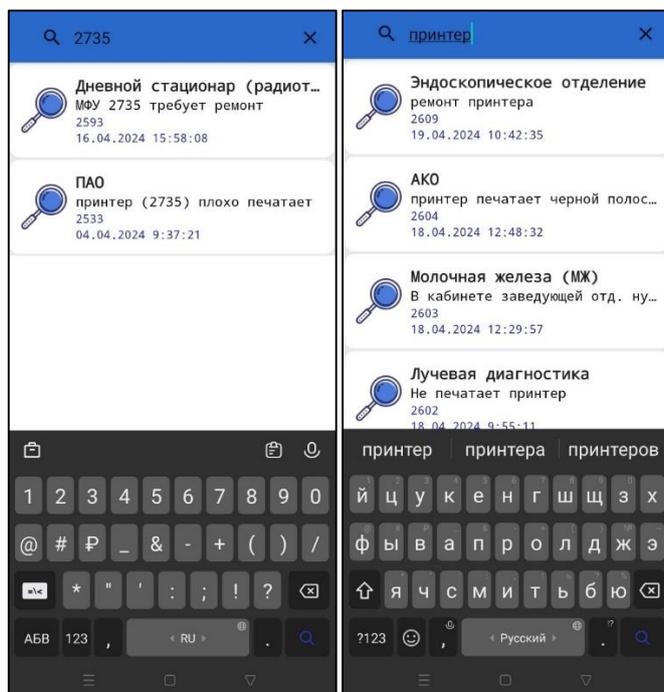


Рисунок 19 – Экран поиска заявок

2.4.5 База данных

Приложения, которые обрабатывают большие объемы данных, могут получать преимущества от хранения этих данных на устройстве. Один из более распространенных способов - кэширование некоторых частей данных, чтобы пользователь мог использовать их даже без доступа к интернету.

Room persistence library обеспечивает уровень абстракции над SQLite, что позволяет удобно работать с базой данных, используя все возможности SQLite. Основные преимущества Room включают в себя проверку SQL-запросов на этапе компиляции, использование удобных аннотаций для уменьшения повторяющегося кода и упрощенные способы миграции базы данных.

Room включает в себя объекты данных, представляющие таблицы в базе данных приложения, объекты доступа к данным (DAO), которые предоставляют методы для запросов, обновлений, вставок и удалений данных, а также класс Database, который предоставляет экземпляры DAO, связанные с базой данных. Приложение может использовать DAO для извлечения данных из базы данных в виде объектов данных и для обновления или вставки новых строк в таблицы.

В приложении реализована база данных состоящая всего из двух несвязанных таблиц.

Таблица «TableLogin» хранит в себе данные авторизации пользователя и роль. Таблица состоит из 4 полей: «id» (ключевое поле, генерируется автоматически, тип данных числовой), «login» (текстовый тип данных), «password» (текстовый тип данных), «permission» (текстовый тип данных).

Таблица «TableTickets» хранит в себе данные по заявкам и состоит из 11 полей: «id» (ключевое поле, генерируется автоматически, тип данных числовой), «ticket_id» (числовой тип данных), «phone» (текстовый тип данных), «room» (текстовый тип данных), «flo» (текстовый тип данных), «department» (текстовый тип данных), «message» (текстовый тип данных),

«status» (текстовый тип данных), «responsible» (текстовый тип данных), «date» (текстовый тип данных), «solution» (текстовый тип данных).

Таблицы формируются из классов данных с определенными аннотациями.

Запросы в базу данных осуществляются по схеме, представленной на рисунке 4. Часть запросов в базу данных выглядит следующим образом.

2.4.6 Уведомления

Уведомления в приложении реализованы с помощью сервиса FCM.

Firestore Cloud Messaging (FCM) — это кроссплатформенная служба, которая обрабатывает отправку, маршрутизацию и постановку сообщений в очередь между серверными приложениями и мобильными клиентскими приложениями. FCM является преемником Google Cloud Messaging (GCM) и основан на службах Google Play.

Реализация FCM включает в себя два основных компонента для отправки и получения:

- Доверенная среда, такая как облачные функции для Firebase или сервер приложений, на которой можно создавать, нацеливать и отправлять сообщения.
- Клиентское приложение Apple, Android или веб-приложение (JavaScript), которое получает сообщения через соответствующую транспортную службу для конкретной платформы.

Firestore Cloud Messaging (FCM) для Android предоставляет возможность отправлять уведомления на устройства пользователей через сервер Firebase.

Шаги реализации:

- Инициализация: сначала необходимо настроить приложение Android для работы с FCM. Это включает добавление зависимостей в файл `build.gradle`, настройку файла `google-services.json` и регистрацию приложения в консоли Firebase.

- Регистрация устройства: при запуске приложения, выполняется запрос на разрешение пользователя на получение уведомлений и затем устройство регистрируется в службе FCM. Каждое устройство имеет свой токен регистрации, который используется для отправки уведомлений.
- Отправка уведомлений: чтобы отправить уведомление на устройство, сервер должен отправить запрос к службе FCM с токеном устройства и данными уведомления (например, заголовок, текст, изображение и другие параметры). FCM затем доставляет уведомление на устройство.
- Обработка уведомлений на устройстве: когда уведомление доставлено на устройство, система Android обрабатывает его и отображает пользователю. Также есть возможность добавить обработчики событий для обработки нажатий на уведомления, например, открытие определенной активности при нажатии на уведомление.
- Дополнительные функции: FCM также предоставляет дополнительные функции, такие как отправка уведомлений в группы пользователей, расширенные параметры уведомлений, планирование отправки уведомлений и другие возможности.

В целом, Firebase Cloud Messaging обеспечивает простой и эффективный способ отправки уведомлений на Android-устройства пользователей и обеспечивает надежную доставку сообщений.

В разрабатываемом мобильном приложении отправка/получение уведомлений работает по следующему алгоритму:

- При первом открытии приложения, функция `onNewToken()` класса `MyFirebaseInstanceIdService` генерирует токен доступа для устройства, сохраняет его в общих настройках приложения и отправляет токен с логином авторизованного пользователя в базу данных 1С при успешной авторизации. Токен сохраняет в справочнике 1С с логином пользователя.
- Необходимо подключить FCM в Google API Console. Google API Console — это облачный инструмент, который позволяет разработчикам

API управлять проектами, использующими Google API. После подключения необходимо запросить токен доступа OAuth 2.0. Токен доступа имеет ограничение час, поэтому его необходимо обновлять.

- Далее необходимо запросить токен обновления для дальнейших запросов токена доступа.
- Все токены сохраняется в базе данных 1С. Обновление токена доступа происходит путем запроса на google сервер каждый час с помощью регламентного задания в 1С.
- Токены используются при запросах на fcm для отправки уведомлений на устройства пользователей. Уведомления можно отправлять как одному конкретному пользователю, так и многим.
- Пример запроса на отправку уведомления

Таким образом при появлении новой заявки/назначении ответственного/закрытия заявки из 1С выполняется запрос и пользователям приходят соответствующие уведомления. (Рисунок 20)

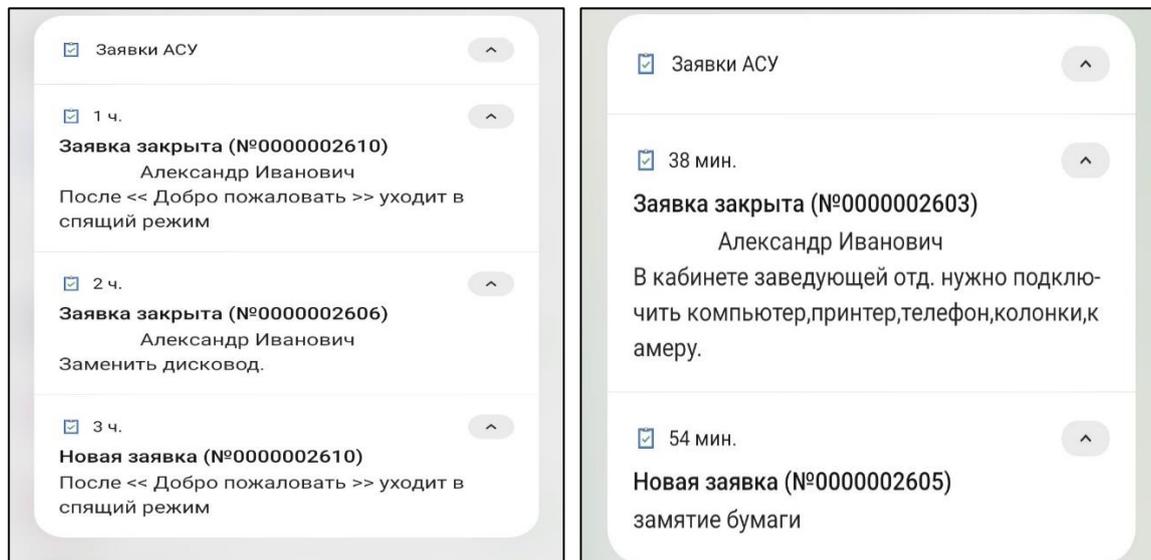


Рисунок 20 – Уведомления

При нажатии на уведомление формируется обратный стек и открывается экран с заявкой. Обратный стек необходим для возвращения на главный экран приложения при выходе с экрана с заявкой, открытой через уведомление.

2.4.7 Интерфейс

Пользовательский интерфейс предназначен для отображения данных приложения на экране, а также служит инструментом для взаимодействия с пользователем. Каждый раз, когда данные изменяются или из-за взаимодействия с пользователем (например, нажатия кнопки) или из-за работы с сервисом (например, ответа сети), пользовательский интерфейс должен обновляться, чтобы отразить эти изменения. То есть, пользовательский интерфейс — это визуальное представление состояния приложения, полученное из уровня данных.

Android Studio предоставляет удобные инструменты для создания и редактирования пользовательского интерфейса с помощью визуального редактора макетов, предварительного просмотра на разных устройствах с помощью эмулятора и возможность быстрого просмотра изменений в реальном времени.

При создании проекта использовались следующие инструменты:

- Создание макета: макет представляет структуру и расположение элементов интерфейса на экране устройства, который создается в виде XML файла в проекте. (Рисунок 21)
- Добавление виджетов: есть возможность добавлять виджеты в макет, такие как кнопки, текстовые поля, изображения, списки и другие элементы, используя соответствующие теги XML. (Рисунок 21)



Рисунок 21 – Макет экрана авторизации и экрана описания заявки

- Настройка виджетов: для каждого виджета можно задать различные свойства, такие как размер, цвет, текст и другие параметры, используя атрибуты XML.
- Для взаимодействия с интерфейсом при каких то событиях, например, выбор заявки из списка пользователем, нужно использовать привязку компонента через функции `setOnClickListener()`.
- При создании интерфейса в студии использовались необходимые ресурсы: строки, изображения, массивы, темы, чтобы сделать интерфейс более гибким, приятным для пользователя.
- Фрагменты позволяют создавать отдельные части интерфейса, которые можно использовать на одном экране сколько угодно раз, например, при

обновлении данных, при этом не будет перезагружать целый экран, а только его часть, которая и является фрагментом.

- Динамические списки для вывода данных о заявках на экране устройства. Такие списки позволяют отображать наборы данных и обеспечивают гибкость элементов и их взаимодействие с пользователем с помощью определенных функций. (Рисунок 13, 14)
- Темы позволяют задавать общий вид интерфейса приложения с помощью определенных параметров для элементов, таких как цвета, шрифты, отступы и другие. Это необходимо чтобы обеспечить внешний вид приложения. В приложении реализована темная тема, которая активируется в зависимости от общих настроек системы, а именно при включенном режиме затемнения (Рисунок 22).

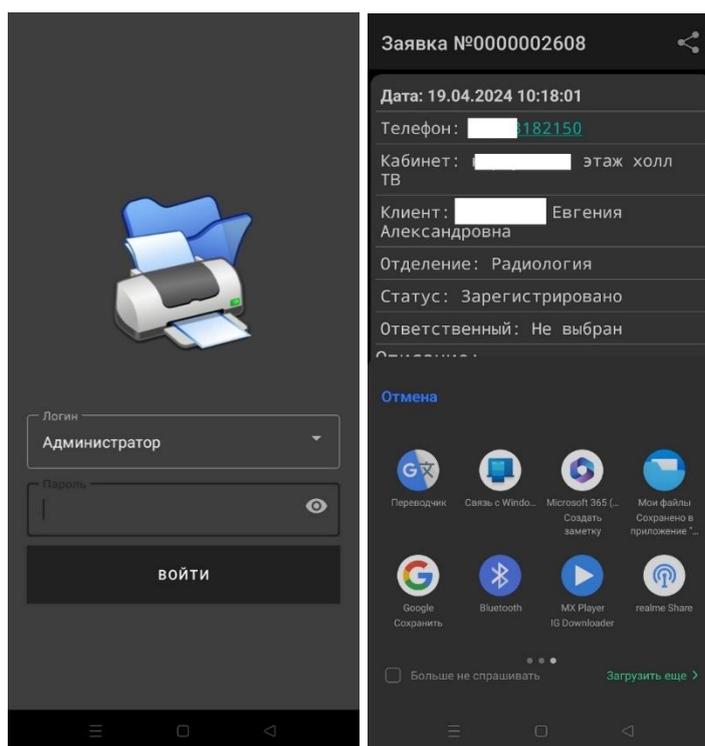


Рисунок 22 – Темная тема приложения

Интерфейс необходимо протестировать на различных устройствах и разрешениях экранов, чтобы убедиться, что все элементы приложения

отображаются правильно. Для тестирования интерфейса в студии разработки можно использовать эмулятор различных устройств, что позволяет быстро протестировать приложение на различных устройствах и уровнях API не имея их физических аналогов. (Рисунок 23)

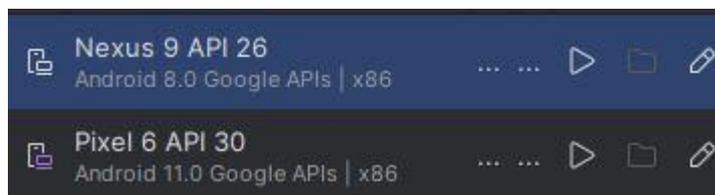


Рисунок 23 – Используемые эмуляторы для тестирования

2.4.8 1С HTTP-сервис

Чтобы получить доступ к базе данных 1С, которая находится в локальной сети организации, необходимо создать в конфигураторе сервис, который будет принимать запросы из сторонних систем и приложений. Функции, которые описаны в сервисе будут вытаскивать нужные данные из базы и отправлять их ответом на запрос по протоколу HTTP.

Опубликовать сервис на сервере 1С. После создания и публикации HTTP-сервиса в 1С, сторонние ПО смогут отправлять HTTP-запросы к сервису и получать ответы с данными.

Таким образом, HTTP-сервис в 1С позволяет обращаться к базе данных 1С из других сервисов и приложений, обеспечивая обмен данными через интернет по протоколу HTTP.

Сервис, разработанный для мобильного приложения, имеет 4 функций (Рисунок 24):

- AuthorizationPostAuthorization(Запрос) – функция принимает запрос с данными авторизации в виде json-строки, сверяет данные авторизации с данными из базы 1С и отправляет в ответ сообщение в зависимости от успешности проверки.

- PostGetTicket(Запрос) – функция принимает запрос с данными по заявке в виде json-строки и меняет статус заявки в базе. В ответ на запрос передается сообщение об успешности выполнения.
- PostGetTicketsForRoom(Запрос) – функция принимает запрос на получение заявок из базы данных 1С, в ответ на запрос передается json-строка с данными по заявкам.
- PostWriteToken(Запрос) – функция принимает запрос с данными пользователя (ФИО) и токеном устройства, записывает данные в базу и в ответе передает сообщение об успешности выполнения.

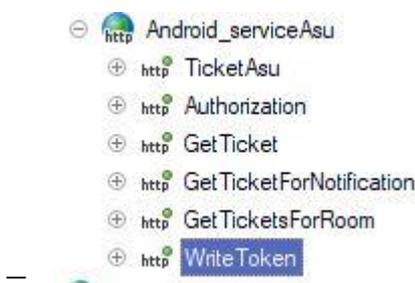


Рисунок 24 – 1С сервис

2.5 Тестирование ПО

2.5.1 Авторизация

Для тестирования авторизации определим несколько различных сценариев:

- Выбрать из списка ФИО и ввести правильный пароль.
- Не выбирать ФИО и не вводить пароль
- Выбрать ФИО из списка и ввести не правильный пароль.
- Сделать запрос на авторизацию при выключенном интернет-соединении.

В первом случае осуществляется запрос на сервис 1С, передаются данные авторизации и приходит ответ (Рисунок 27), после чего открывается экран со списками заявок.

Во втором случае получим ответ сервиса и всплывающее сообщение на экране. (Рисунок 28)

В третьем случае получим ответ сервиса и всплывающее сообщение на экране. (Рисунок 25)

В четвертом случае получим всплывающее сообщение. (Рисунок 26)

```
2024-04-23 15:25:01.569 7064-7064 Ответ серв...вторизации com.onco.tabticketasu I Неверный логин или пароль
```

Рисунок 25 – Ответ сервиса



Рисунок 26 – Всплывающее сообщение

```
2024-04-23 15:35:18.015 7484-7484 Данные авторизации com.onco.tabticketasu I {"data":{"Login":"Администратор","password":"1a123310dd34d97c0d2330e88c494bd40dbec111","permission":"Администратор"}}
2024-04-23 15:35:18.184 7484-7484 Ответ серв...вторизации com.onco.tabticketasu I true
2024-04-23 15:35:18.287 7484-7484 Permission com.onco.tabticketasu I Администратор
```

Рисунок 27 – Ответ сервиса

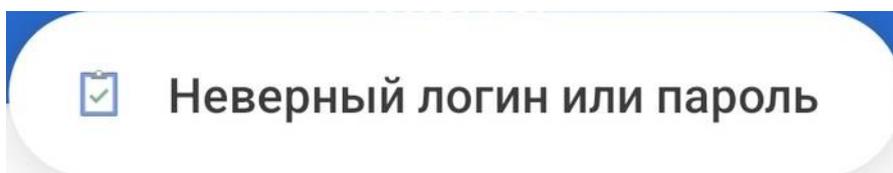


Рисунок 28 – Ответ сервиса

2.5.2 Назначить ответственного

Для тестирования функции «Назначить ответственного» определим несколько различных сценариев:

- Выбрать из списка ФИО и нажать кнопку «Назначить ответственного» при выключенном интернет-соединении.
- Выбрать из списка ФИО и нажать кнопку «Назначить ответственного», если ответственный уже назначен.
- Выбрать из списка ФИО и нажать кнопку «Назначить ответственного» при существующем интернет-соединении и, если не выбран ответственный.

В первом случае получим ответ сервиса (Рисунок 26) и всплывающее сообщение на экране.

Для тестирования второго и третьего сценария необходимо создать тестовую заявку в 1С. (Рисунок 29)

Рисунок 29 – Тестовая заявка

Во втором случае получим ответ сервиса (Рисунок 30) и всплывающее сообщение на экране.

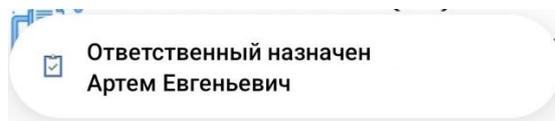


Рисунок 30 – Всплывающее сообщение

В третьем случае получим ответ сервиса (Рисунок 30) и всплывающее сообщение на экране. Также назначенному ответственному придет уведомление (Рисунок 31).

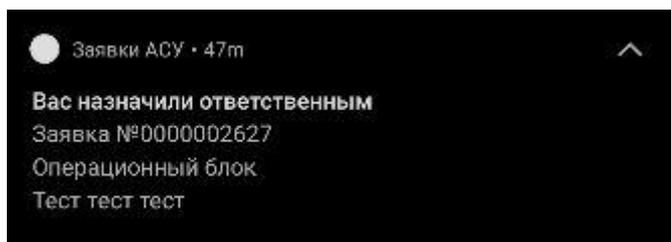


Рисунок 31 – Уведомление

2.5.3 Закрывать заявку

Для тестирования функции «Закрывать заявку» определим несколько различных сценариев:

- Нажать кнопку «Закрывать заявку» при выключенном интернет-соединении.
- Нажать кнопку «Закрывать заявку» при включенном Интернет-соединении и без заполненного поля «Решение».
- Нажать кнопку «Закрывать заявку» при включенном Интернет-соединении и с заполненным полем «Решение».

В первом случае получим ответ сервиса (Рисунок 25) и всплывающее сообщение на экране.

Во втором случае получим ответ сервиса (Рисунок 32) и всплывающее сообщение на экране.

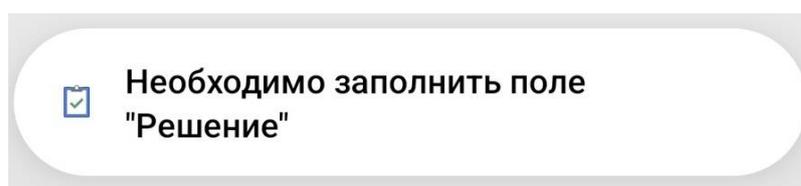


Рисунок 32 – Всплывающее сообщение

В третьем случае получим ответ сервиса (Рисунок 33) и всплывающее сообщение на экране. Также администратору придет уведомление (Рисунок 34).



Рисунок 33 – Всплывающее сообщение

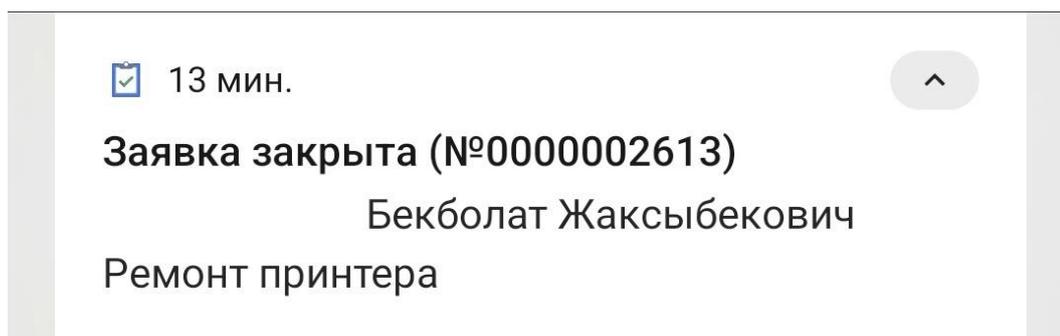


Рисунок 34 – Уведомление

2.5.4 Принять заявку

Для тестирования функции «Принять заявку» определим несколько различных сценариев:

- Нажать кнопку «Принять заявку» при выключенном Интернет-соединении.
- Нажать кнопку «Принять заявку» при включенном Интернет-соединении и, если ответственный уже назначен.
- Нажать кнопку «Принять заявку» при включенном Интернет-соединении и с не выбранным ответственным.

В первом случае получим ответ сервиса (Рисунок 26) и всплывающее сообщение на экране.

Во втором случае получим ответ сервиса и всплывающее сообщение на экране. (Рисунок 30) В третьем случае получим ответ сервиса (Рисунок 35) и всплывающее сообщение на экране. Также администратору придет уведомление (Рисунок 36).

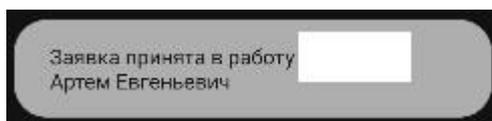


Рисунок 35 – Всплывающее сообщение

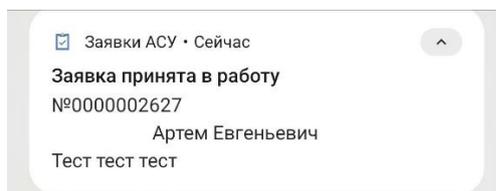


Рисунок 36 – Уведомление

2.5.5 Гибкость интерфейса

Гибкость интерфейса можно проверить на разных эмуляторах (Рисунок 37-38).

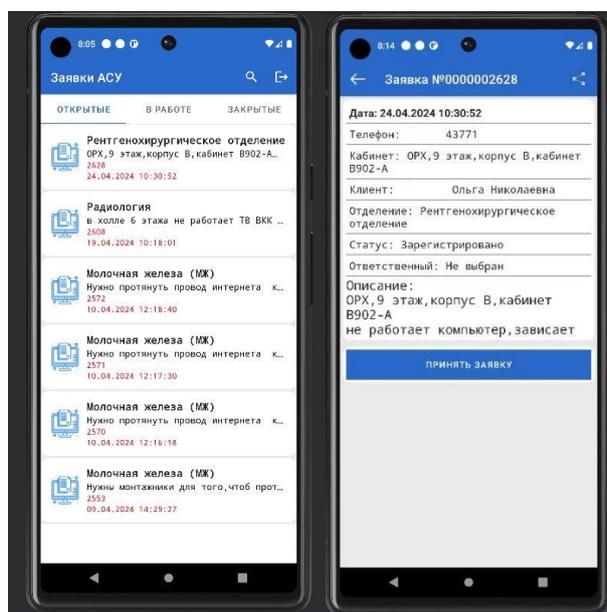


Рисунок 37 – Эмулятор «Pixel 6» API 30 1080x2400

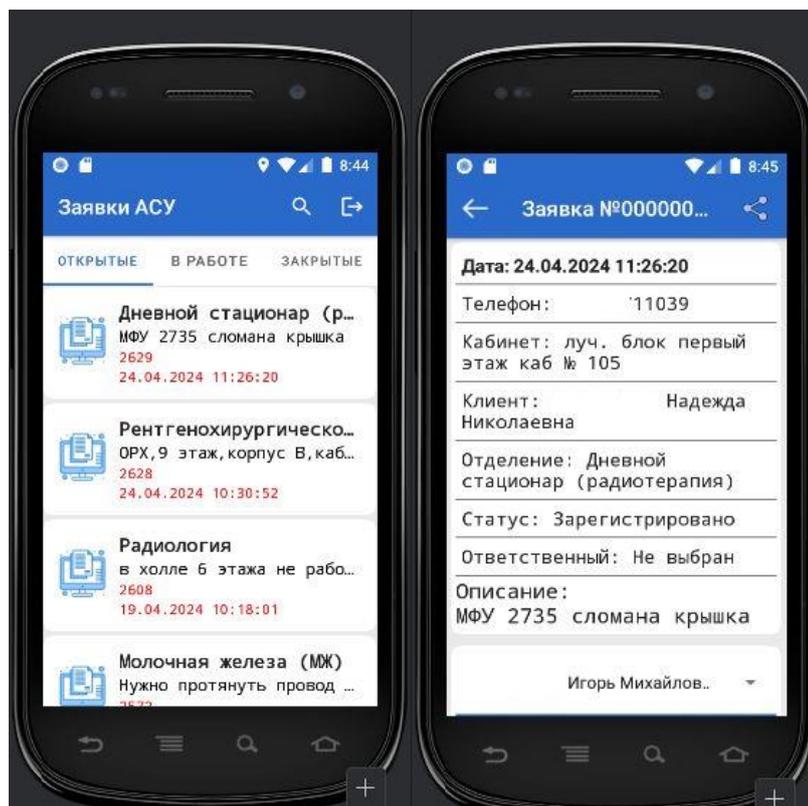


Рисунок 38 – Эмулятор «Nexus S» API 26 480x800

2.6 Руководство пользователя

2.6.1 Введение

Данное руководство пользователя предназначено для пользователей мобильного приложения на платформе Android, предназначенного для интеграции с 1С конфигурацией. Это приложение предоставляет удобный и эффективный способ управления заявками, связанными с работой в 1С системе.

2.6.2 Назначение

Приложение предлагает ряд функциональных возможностей, таких как управление заявками, авторизация, поиск по базе данных, прием уведомлений о новых заявках и их статусах, распределение ролей, возможность обмена

информацией о заявках через сторонние программы и совершение звонков прямо из приложения.

В данном руководстве мы подробно рассмотрим каждую из перечисленных функций, предоставим инструкции по использованию приложения, а также дадим рекомендации по оптимальному взаимодействию с 1С конфигурацией через мобильное приложение.

2.6.3 Системные требования

Таблица 1 – Системные требования мобильного устройства

№	Мобильное устройство	
1	Операционная система	Android 8.0 или выше
2	ОЗУ	4 ГБ или больше
3	Размер встроенной памяти	32 ГБ или больше
4	Поддержка Wi-Fi	Любой версии
5	Поддержка мобильной сети	3G или выше

2.7 Описание процессов

2.7.1 Регистрация пользователя

Таблица 2 – Порядок действий регистрации пользователя

№	Действие	Ответственный
1	Подача заявки на создание нового пользователя (в отдел АСУ). Шаблоны заявлений находятся на файл сервере в папке «документация».	Руководитель подразделения, запрашивающий доступ
2	Завести нового пользователя в 1С конфигурации, предоставить новому пользователю данные авторизации.	Ведущий 1С программист
3	Подача служебной записки руководителю подразделения АСУ для предоставления доступа к внутренним ресурсам предприятия	Руководитель подразделения, запрашивающий доступ

2.7.2 Установка мобильного ПО

1. Скачать на мобильное устройство установочный файл, который лежит на файле сервере предприятия в папке «Программное обеспечение» под названием «TicketsASU.apk».
2. Открыть скачанный файл на мобильном устройстве, дать разрешение на установку приложений из неизвестных источников, установить приложение.
3. Открыть приложение, на экране авторизации выбрать свое ФИО из выпадающего списка, ввести полученный пароль, выданный при регистрации и нажать кнопку «Войти». (Рисунок 39)

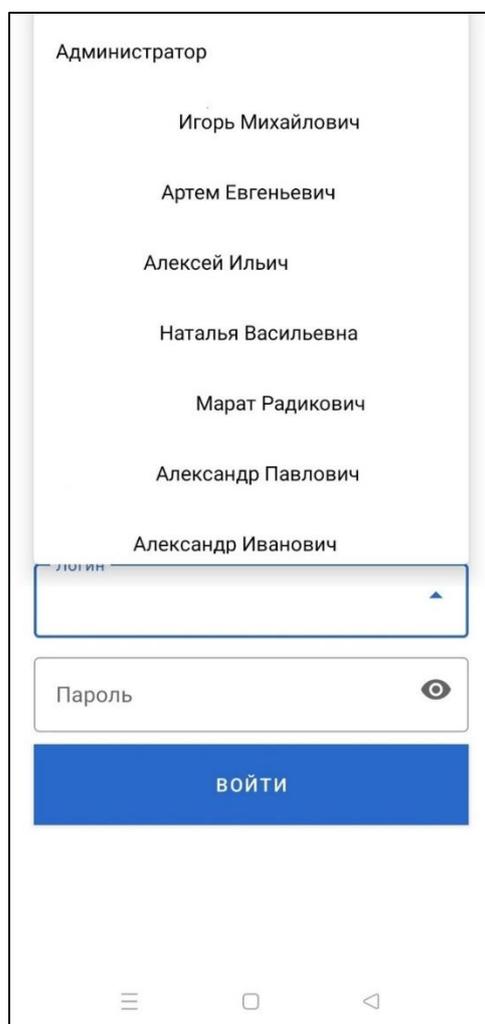


Рисунок 39 – Авторизация

2.8 Описание операций

2.8.1 Просмотр заявок

Основная экранная форма приложения «Заявки АСУ» предоставляет доступ ко всем спискам заявок. Содержание экранной формы и доступные функции зависят от роли пользователя в системе. Общими для любой роли являются следующие элементы экранной формы:

- Список заявок
- Поиск заявок в базе
- Выход из аккаунта

Всего при регистрации пользователя в системе возможна выдача ему одной из 2 ролей: пользователь; администратор.

Для пользователя с ролью "пользователь" динамические списки заявок на вкладке «Открытые» содержат все новые заявки в системе, для вкладки «В работе» содержат заявки, выполнением которых занят авторизованный пользователь и для вкладки «Закрытые» содержат все закрытые заявки в системе.

Для пользователя с ролью "администратор" динамические списки заявок на вкладке «Открытые» содержат все новые заявки в системе, для вкладки «В работе» содержат все заявки, выполнением которых заняты сотрудники отдела и для вкладки «Закрытые» содержат все закрытые заявки в системе.

2.8.2 Управление заявками

По нажатию на заявку во вкладке из списка открывается форма заявки. Содержание экранной формы и доступные функции зависят от роли пользователя в системе. Общими для любой роли являются следующие элементы экранной формы:

- Полное описание заявки
- Кнопка «Поделиться»
- Кликабельный номер телефона

Для пользователя с ролью "пользователь" в зависимости от выбранной категории заявки, форма содержит кнопку «Принять заявку» или «Закрывать заявку».

Для пользователя с ролью "администратор" в зависимости от выбранной категории заявки, форма содержит кнопку «Назначить ответственного» со списком выбора сотрудника отдела или кнопку «Закрывать заявку».

2.8.3 Переслать заявку

Пользователи могут поделиться информацией о заявке с другими пользователями через сторонние программы, такие как мессенджеры или социальные сети. Кнопка «Поделиться» находится в верхней части формы с полным описанием заявки. (Рисунок 40).

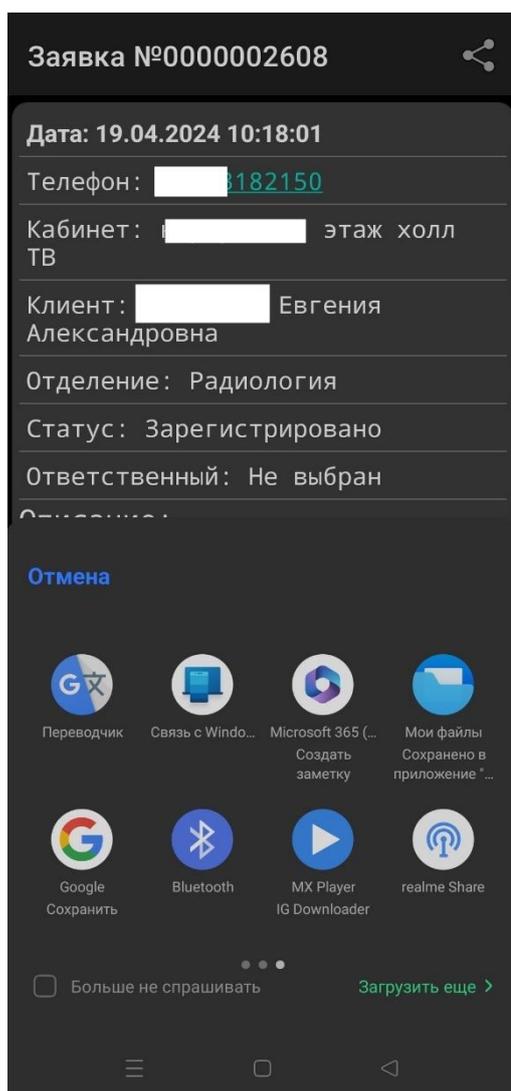


Рисунок 40 – Функция «Поделиться»

2.8.4 Поиск заявок

Пользователи могут найти заявки в локальной базе через поиск заявок. Кнопка поиска находится в верхней части приложения на экране со списками заявок. Заявки можно искать по любому реквизиту заявки. (Рисунок 41).

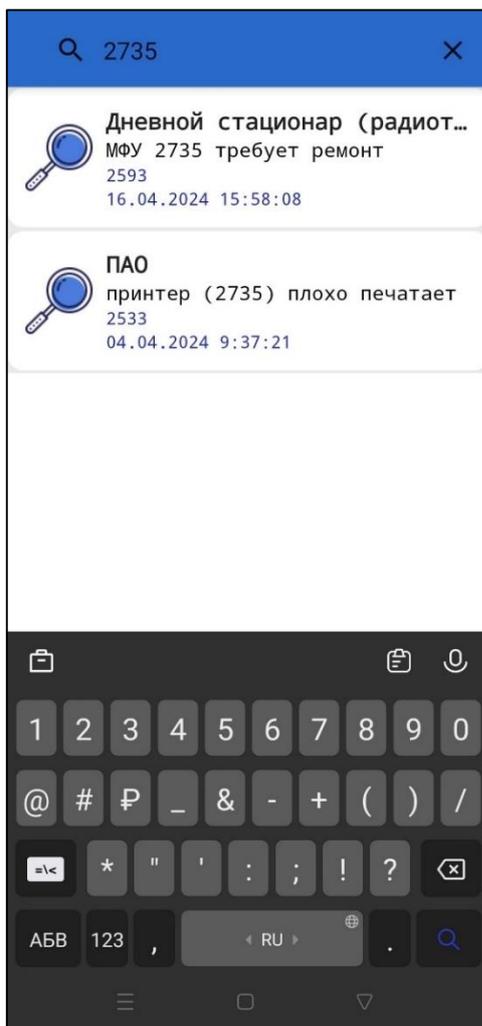


Рисунок 41 – Поиск заявок

2.8.5 Выход из аккаунта

Любой авторизованный пользователь может выйти из аккаунта. Кнопка выхода находится в верхней части приложения на экране со списками заявок. При выходе из аккаунта, все данные в приложении удаляются и уведомления больше не приходят.

2.8.6 Аварийные ситуации

В процессе эксплуатации мобильного приложения аварийные ситуации, выражающиеся в отсутствии отклика сервера, могут возникать по следующим причинам:

- физическая потеря связи с сервером системы;
- изменение сетевого адреса сервера или сервиса 1С;
- проведение регламентных работ на сервере, связанных с отключением служб и сервисов.

При отсутствии отклика сервера следует обращаться к администратору системы либо к системному администратору ЛВС. Специалисты проведут диагностику и исправление неполадки.

Глава 3 Экономическое обоснование проекта

Эффективность ИТ проекта – это степень его соответствия своему назначению. Принято различать функциональную и экономическую эффективность.

Можно выделить три основные группы методов, позволяющих определить эффект от внедрения любого ИТ проекта:

- финансовые
- качественные
- вероятностные

Чаще всего применяются следующие основные финансовые методы:

- чистый приведенный доход (Net Present Value, NPV);
- расчет рентабельности инвестиций ROI (Return on Investment);

Для составления таблицы расходов на разработку мобильного приложения, нужно учитывать различные категории расходов [3], [17]. Ниже приведена таблица, в которой указаны основные категории расходов при разработке и их стоимость:

Таблица 3 – Категории расходов на разработку

Категория расходов	Стоимость
Заработная плата разработчику	210 000р.
Тестирование	20 000р.
Обучение и повышение квалификации команды	7 000р.
Обслуживание и поддержка	70 000р.
Компьютер для разработки	60 000р.
Итого:	367 000р.

Ожидаемые доходы от использования приложения:

- Сокращение времени на обработку заявок на 20%: средний доход от увеличения производительности в месяц составляет 80 000р.
- Повышение эффективности работы персонала: сокращение издержек на персонал на 15% в месяц, что эквивалентно 60 000р.

3.1 Расчет чистой приведенной стоимости – NPV

Метод расчета чистой приведенной стоимости проекта позволяет оценить его дисконтированную стоимость, определяемую как разность между дисконтированными (т.е. приведенными к настоящему моменту) ожидаемыми поступлениями от реализации проекта и дисконтированными затратами на его осуществление, включая величину первоначальных инвестиций [17].

Метод чистой текущей стоимости (NPV) состоит в следующем: 1. Определяется текущая стоимость первоначальных инвестиций (I_0), т.е. решается вопрос, сколько инвестиций нужно зарезервировать для проекта. 2. Рассчитывается текущая стоимость будущих денежных поступлений от проекта, для чего доходы за каждый год CF (денежный поток) приводятся к текущей дате. Результаты расчетов показывают, сколько средств нужно было бы вложить сейчас для получения запланированных доходов, если бы ставка доходов была равна барьерной ставке (для инвестора ставке альтернативной доходности, для предприятия цене совокупного капитала или через риски). Подытожив текущую стоимость доходов за все годы, получим общую текущую стоимость доходов от проекта PV (present value):

$$PV = \sum_{t=0}^n \frac{CF_t}{(1+r)^t}$$

где PV – общая текущая стоимость доходов проекта;

n – число периодов;

CF_t – приток денежных средств в период t ;

r – ставка дисконтирования.

Текущая стоимость инвестиционных затрат (I_0) сравнивается с текущей стоимостью доходов (PV). Разность между ними составляет чистую текущую стоимость доходов (NPV):

$$NPV = PV - I_0$$

I_0 – величина исходных инвестиций в нулевой период.

Учитывая показатели издержек и доходов, сначала определим чистый денежный поток (Net Cash Flow) на каждый год.

1. Расходы: 367 000р. (Таблица 3)

2. Доходы:

– Доход от увеличения производительности за год: 960 000р.

– Сокращение расходов за год: 720 000р.

Денежный поток:

Год 1: 960 000р. + 720 000р. – 367 000р. = 1 313 000р.

Год 2: 960 000р. + 720 000р. = 1 680 000р.

Год 3: 960 000р. + 720 000р. = 1 680 000р.

NPV при ставке дисконтирования в 10%:

$$NPV = (1\,313\,000\text{р.} / 1.10^1) + (1\,680\,000\text{р.} / 1.10^2) + (1\,680\,000\text{р.} / 1.10^3) + (-367\,000\text{р.})$$

$$NPV = 1\,193\,636\text{р.} + 1\,388\,429\text{р.} + 1\,262\,228\text{р.} - 367\,000$$

$$NPV = 3\,477\,293\text{р.}$$

NPV для проекта составляет 3 477 293р. Проект приносит необходимую прибыль для окупаемости инвестиций.

1.1 ROI (Return on Investment)

Коэффициент рентабельности инвестиций — показатель, который отражает окупаемость всех вложений в продукт, рекламные кампании или проект [15], [16]. Вложениями являются расходы: зарплата коллективу, производство продукта, упаковка, вложение в компанию-стартап, аренда

склада и помещений, покупка сервисов и оборудования для работы, маркетинг и многое другое.

Показатель ROI используется:

- Для аналитики рекламных кампаний, чтобы узнать, какая сумма денежных средств возвращается в организацию.
- Инвестирование денег в компанию или проект: новая программа для ПК, приложение, строительство недвижимости, развитие бизнеса или спортивной команды.

При вложении средств в стартап, ценные бумаги или акции. Возврат инвестиций рассчитывают для того, чтобы не допустить ошибок, которые приведут к пустой трате денежных средств. То есть принять важное управленческое решение: продолжать ли покупать сырье для производства продукта, оставлять или убирать канал продвижения, закрыть проект или прекратить участие в нем.

ROI относится к показателям эффективности, а значит — указывает, что на полученный коэффициент стоит обратить внимание.

Для расчета ROI можно воспользоваться формулой:

$$ROI = ((\text{Доходы} - \text{Расходы}) / \text{Расходы}) * 100\%$$

1. Доходы:

a. Доход за год: 960 000р.

b. Уменьшение расходов за год: 720 000р.

c. Общие доходы: 1 680 000р.

2. Расходы: 367 000р. (Таблица 3)

3. ROI:

$$ROI = ((1\,680\,000\text{р.} - 367\,000\text{р.}) / 367\,000\text{р.}) * 100\%$$

$$ROI = (-1\,313\,000\text{р.} / 367\,000\text{р.}) * 100\%$$

$$ROI = 357.22\%$$

Положительное значение ROI означает, что инвестиции окупились и принесли прибыль.

Заключение

В итоге данной квалификационной работы было разработано мобильное приложение под операционную систему Android, предназначенное для управления заявками в отделе АСУ организации и интегрировано с информационной системой 1С. Приложение обеспечивает удобный доступ к заявкам из любого места с интернет-соединением, позволяет просматривать информацию по заявкам, отслеживать статус, назначать ответственных, принимать и закрывать заявки.

Основная цель проекта заключалась в сокращении времени на выполнение задачи сотрудниками отдела АСУ, ускорении реагирования и взаимодействия с клиентами и обеспечении удобного доступа к заявкам. Разработанное приложение позволяет пользователям управлять заявками через сервис 1С конфигурации с мобильных устройств.

Функциональные требования приложения реализованы, включая управление заявками, авторизацию, поиск по базе данных, получение уведомлений, распределение ролей, функцию "Поделиться" заявкой и возможность совершения звонка прямо из приложения.

В результате было написано приложение, которое поможет улучшить и повышению эффективности работы отдела АСУ, а также обслуживанию клиентов. Разработанное мобильное приложение представляет собой весомый шаг в развитии используемых информационных технологий организации и управления заявками, который принесет организации большие выгоды и конкурентные преимущества.

Список используемой литературы

1. Анализ и оптимизация бизнес-процессов : лабораторный практикум / составители М. Г. Романенко. — Ставрополь : Северо-Кавказский федеральный университет, 2015. — 79 с. Текст : электронный. URL: <https://www.iprbookshop.ru/62919.html>
2. Андрианова А.А. Алгоритмизация и программирование. Практикум А.А. Андрианова, Л.Н. Исмагилов, Т.М. Мухтарова. – СПб.: Изд-во Лань, 2019. – 240 с.
3. Бабанов А.М. Технология разработки программного обеспечения: структурный подход. – Томск: ТГУ, 2016. – 157 с.
4. Букунов, С. В. Автоматизация процессов бизнес-планирования с помощью системы управления проектами MS Project : учебное пособие / С. В. Букунов, О. В. Букунова. — Санкт-Петербург : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 72 с. — ISBN 978-5-9227-0746-6. Текст : электронный. URL: <https://www.iprbookshop.ru/74321.html>
5. Гагарина, Л. Г. Технология разработки программного обеспечения : учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0707-8. - Текст : электронный. URL: <https://znanium.com/catalog/product/1011120>
6. ГБУЗ «Санкт-Петербургский клинический научно-практический центр специализированных видов медицинской помощи (онкологический)»: официальный сайт. – Санкт-Петербург, 2010. – URL <https://oncocentre.ru>
7. Гвоздева В.А. Информатика, автоматизированные информационные технологии и системы. - М.: ИНФРА-М, 2013. - 544 с.
8. Грекул В.И. Проектирование информационных систем / В.И. Грекул, Н.Л. Коровкина, Г.А. Левочкина. – М.: Издательство Юрайт, 2017. – 385 с.

9. Долженко, А. И. Технологии командной разработки программного обеспечения информационных систем : курс лекций / А. И. Долженко. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 300 с. — ISBN 978-5-4486-0525-3. — Текст : электронный. URL: <http://www.iprbookshop.ru/79723.html>
10. Книга составлена по материалам электронного аналога издания 1С Программирование для начинающих. Детям и родителям, менеджерам и руководителям: М.Г. Радченко. Разработка в системе 1С:Предприятие 8.3 (ISBN 978-5-9677-2628-6, М.: ООО 1С-Паблишинг 2017) опубликованного в информационной системе ИТС ПРОФ (ISBN 978-5-9677-2631-6)
11. Коцюба И.Ю. Основы проектирования информационных систем / И.Ю. Коцюба, А.В. Чунаев, А.Н. Шиков. – СПб.: Университет ИТМО, 2015. – 206 с.
12. Кудрина Е.В. Основы алгоритмизации и программирования на языке С# / Е.В. Кудрина, М.В. Огнева. – М.: Изд-во Юрайт, 2019. – 322 с.
13. Кумагина Е.А. Модели жизненного цикла и технологии проектирования программного обеспечения / Е.А. Кумагина, Е.А. Неймарк. – Нижний Новгород: Изд-во ННГУ, 2016. – 41 с.
14. Мезенцев, К.Н. Автоматизированные информационные системы. - М.: ИЦ Академия, 2013. - 176 с.
15. Михеева Е.В. Информатика / Е.В. Михеева, О.И. Титова. – М.: Издательский центр «Академия», 2014. – 352 с.
16. Мокеев В.В. Бизнес-информатика / В.В. Мокеев, Е.В Бунова, О.С. Буслаева. – Челябинск: издательский центр ЮУрГУ, 2015. – 67 с.
17. Официальная документация по разработке ПО на Android: официальный сайт. – California, 2021. – URL: developer.android.com (дата обращения 15.03.2024).
18. Хабр (бывший Хабрахабр) – русскоязычный веб-сайт в формате системы тематических коллективных блогов (именуемых хабами) с элементами новостного сайта, созданный для публикации новостей,

аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и интернетом: официальный сайт. – Кипр, 2006. – URL <https://habr.com>.

19. Шустова Л.И. Базы данных / Л.И. Шустова, О.В. Тараканов. – М.: Инфра-М, 2016. – 304 с.

20. Ясенев В.Н. Информационные системы и технологии в экономике. - М.: ЮНИТИ, 2014. - 560 с.