

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

институт

Прикладная информатика, математика и естественнонаучные
дисциплины
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ Е.Н.Скуратенко
подпись инициалы, фамилия

« ____ » июня _____ 2016г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 «Прикладная информатика»

код – наименование направления

Организация мобильного доступа
для системы управления заявками VsDeskв ГАУ РХ «ЦИНТ Хакасии»

тема

Руководитель

подпись, дата

доцент, канд.пед.наук
должность, ученая степень

Т.В. Соловьева

инициалы, фамилия

Выпускник

подпись, дата

П.А. Бредюк

инициалы, фамилия

Нормоконтролер

подпись, дата

В.И. Кокова

инициалы, фамилия

Абакан 2016

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
"Сибирский федеральный университет"

институт

Прикладная информатика, математика и естественнонаучные
дисциплины
кафедра

УТВЕРЖДАЮ

Заведующий

кафедрой

Е.Н.Скуратенко

подпись инициалы, фамилия

« ____ » _____ 2016 г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы
бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту Бредюку Павлу Андреевичу

фамилия, имя, отчество

Группа 52-1 (ХБ 12-06) Направление (специальность) 09.03.03

номер

код

«Прикладная информатика»

наименование

Тема выпускной квалификационной работы Организация мобильного доступа для системы управления заявками VsDesk в ГАУ РХ «ЦИНТ Хакасии»

Утверждена приказом по институту № 170 от 01.03.2016
Руководитель ВКР Т.В.Соловьева, доцент, канд.пед.наук, ХТИ – филиал СФУ
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР – заказ ГАУ РХ «ЦИНТ ХАКАСИИ»

Перечень разделов ВКР

1. Организационно-экономическая характеристика государственного автономного учреждения Республики Хакасия «Центр информатизации и новых технологий Республики Хакасия».

2. Разработка приложения для мобильной платформы Android.

3. Расчет экономической эффективности проекта.

Перечень графического материала –

Руководитель ВКР

Т.В. Соловьева

подпись

инициалы и фамилия

Задание принял к исполнению _____ П.А. Бредюк
подпись, инициалы и фамилия студента

« 01 » марта 2016 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Организация мобильного доступа для системы управления заявками vsDesk в ГАУ РХ «ЦИНТ Хакасии» содержит 72 страницы текстового документа, 19 рисунков, 14 таблиц, 23 формулы, 10 использованных источников, 2 приложения.

СИСТЕМА УПРАВЛЕНИЯ ЗАЯВКАМИ VSDESK, МОБИЛЬНЫЙ ДОСТУП, КЛИЕНТ, ПРИЛОЖЕНИЕ, СЕТЕВЫЕ ТЕХНОЛОГИИ.

Цель работы: повышение эффективности работы сотрудников с системой контроля выполнения заявок vsDesk за счет организации мобильного доступа.

Задачи проекта: провести анализ работы «ЦИНТ Хакасии», построить модель информационных потоков организации, определить цель и задачи проектирования приложения для мобильной платформы Android, проанализировать и выбрать средства разработки, создать приложение на базе платформы Android, рассчитать экономический эффект от внедрения проекта.

Новые требования бизнеса – доступ к системе автоматизации обработки запросов vsDeskc мобильных устройств. Для повышения эффективности работы сотрудников с системой решено разработать мобильное приложение.

Изучена система vsDesk, выявлены ее архитектура и структура компонентов. Базовый интерфейс для разработанного приложения взят из системы vsDesk с изменениями, обусловленными переходом на другую платформу. В качестве базового функционала использованы две категории задач системы: обработка запросов и база знаний.

Созданное приложение разработано в IDE Android Studio для операционной системы Android с помощью языка объектно-ориентированного программирования Java.

Работа находится на этапе тестирования.

ABSTRACT

Graduation thesis on the theme of «Mobile access organization to vsDesk application management system in the state autonomous institution «Information and New Technologies Center of the Republic of Khakassia» contains 80 pages, 32 images, 12 tables, the list of 10 references and 5 applications.

VSDesk APPLICATION MANAGEMENT SYSTEM, MOBILE ACCESS, CLIENTS, APPLICATION, NETWORK TECHNOLOGIES.

The objective is to improve work efficiency of employees with vsDesk application management system by organizing mobile access.

The tasks are to analyze the work of «Information and New Technologies Center of the Republic of Khakassia», to create an information flow model of the institution, to define the aim and tasks of designing application for the Android OS Mobile Platform, to analyze and choose design tools, to estimate economic efficiency from the project implementation.

New business needs is the access to vsDesk application processing system using mobile devices. To improve work efficiency of employees with the system, a mobile application is developed.

vsDesk system is considered, its architecture and structure components are determined. Base interface for the developed application is taken from vsDesk system, including changes caused by another platform place. Two categories of system tasks are used as a base functional: application processing and database system.

Created application is developed in IDE Android Studio for Android OS using an object-oriented language Java programming.

The work is in the testing phase.

Normkontroller of foreign language

signature, date

Elokhova G.V.
(surname, name, patronymic)

СОДЕРЖАНИЕ

Введение.....	8
1 Организационно-экономическая характеристика государственного автономного учреждения Республики Хакасия «Центр информатизации и новых технологий Республики Хакасия»	9
1.1 Анализ деятельности ГАУ РХ «ЦИНТ Хакасии»	9
1.2 Структурная схема правительства республики Хакасия и предприятия «ЦИНТ Хакасии»	11
1.3 ИТ-инфраструктура организации «ЦИНТ Хакасии»	14
1.4 Постановка цели и задач проектирования.....	22
1.5 Анализ и выбор среды разработки	23
1.6 Выводы по разделу «Организационно-экономическая характеристика государственного автономного учреждения Республики Хакасия «Центр информатизации и новых технологий Республики Хакасия».....	25
2 Разработка приложения для мобильной платформы Android	26
2.1 Технологии разработки приложения на платформе Android	26
2.2 IDEAndroidStudio	27
2.2.1 Интерфейс AndroidStudio	27
2.2.2 Эмуляция приложения.....	30
2.3 Проектирование интерфейса приложения	33
2.3.1 Интерфейс главного модуля	33
2.3.2 Интерфейс модуля добавления заявки.....	35
2.4 Программная реализация приложения	36
2.4.1 Программная реализация главного модуля.....	36
2.4.2 Программная реализация модуля добавления заявок	40

2.5 Выводы по разделу «Разработка приложения для мобильной платформы Android».....	41
3 Расчет показателей экономической эффективности от внедрения мобильного приложения в организацию «ЦИНТ Хакасии»	43
3.1 Расчет трудозатрат на разработку мобильного приложения.....	43
3.2 Экономическая оценка разработки мобильного приложения.....	48
3.2.1 Постоянные издержки	49
3.3 Расчет эффективности внедрения для организации «ЦИНТ Хакасии» .	58
3.4 Выводы по разделу «Расчет показателей экономической эффективности от внедрения мобильного приложения в организацию «ЦИНТ Хакасии»..	62
Заключение	64
Список использованных источников	65
Приложение А	66
Приложение Б.....	70

ВВЕДЕНИЕ

В настоящее время в ГАУ РХ «ЦИНТ Хакасии» разработана и эксплуатируется автоматизированная система контроля выполнения заявок и пересылки сообщений vsDesk. Работа системы осуществляется по локальной сети, поэтому область действия системы не выходит за пределы организации. Одной из ее важнейших задач выступает техническое обслуживание компьютерного парка Правительства Республики Хакасия. Поэтому было решено создать мобильное приложение системы vsDesk с целью повышения эффективности работы информационной системы.

Цель: повышение эффективности работы сотрудников с системой контроля выполнения заявок vsDesk за счет организации мобильного доступа.

Задачи:

- Провести анализ деятельности ГАУ РХ «ЦИНТ Хакасии».
- Построить модель информационных потоков организации.
- Определить цель и задачи проектирования приложения для мобильной платформы Android.
- Проанализировать и выбрать средства разработки приложения.
- Разработать приложение на базе платформы Android.
- Рассчитать экономический эффект от внедрения проекта.

В аналитическом разделе детально изучена система предприятия vsDesk, проанализированы среды разработки на платформе Android, сделан выбор в пользу одной с обоснованием причин выбора. В проектном разделе описана выбранная интегрированная среда разработки (IDE) и этапы создания приложения, начиная с интерфейса, и заканчивая программным кодом.

1 Организационно-экономическая характеристика государственного автономного учреждения Республики Хакасия «Центр информатизации и новых технологий Республики Хакасия»

1.1 Анализ деятельности ГАУ РХ «ЦИНТ Хакасии»

Полное наименование организации – государственное автономное учреждение Республики Хакасия «Центр информатизации и новых технологий Республики Хакасия».

Адрес учреждения: 655001, Республика Хакасия, г. Абакан, ул. Щетинкина, 18.

ГАУ РХ «ЦИНТ Хакасии» основан с целью создания современной, высокотехнологичной, работоспособной и надежной системы в Республике, которая обеспечит максимально эффективное информационное взаимодействие между органами государственной власти Республики, органами местного самоуправления, а также общественными институтами и гражданами.

ГАУ РХ «ЦИНТ Хакасии» осуществляет техническую поддержку и сервисное обслуживание Аппарата Правительства Республики Хакасия, некоторых региональных органов исполнительной власти, подведомственных учреждений, а также коммерческих организаций. Суммарно это более 300 рабочих мест и свыше 500 единиц техники.

Вид деятельности государственного учреждения:

- Деятельность в области электросвязи.
- Оптовая торговля офисными машинами и оборудованием.
- Разработка программного обеспечения и консультирование в этой области.
- Деятельность по созданию и использованию баз данных и информационных ресурсов, в том числе ресурсов сети Интернет.

– Консультирование по аппаратным средствам вычислительной техники.

– Рекламная деятельность.

– Предоставление прочих услуг.

– Обработка данных.

– Прочая розничная торговля в специализированных магазинах.

– Техническое обслуживание и ремонт офисных машин и вычислительной техники.

– Аренда прочего автомобильного транспорта и оборудования.

– Предоставление услуг по монтажу, ремонту и техническому обслуживанию прочего электрооборудования, не включенного в другие группировки.

– Производство фильмов.

– Розничная торговля по заказам.

– Монтаж прочего инженерного оборудования.

– Научные исследования и разработки в области естественных и технических наук.

– Аренда легковых автомобилей.

– Деятельность в области оказания услуг межсистемной связи.

– Аренда офисных машин и оборудования, включая вычислительную технику.

– Прочая розничная торговля вне магазинов.

Основной целью предприятия «ЦИНТ Хакасии» является – централизация информатизации органов государственной власти в республики Хакасия[8].

Подцелями организации являются:

– Создание и развитие (модернизация) информационных систем и компонентов информационно-телекоммуникационной инфраструктуры.

– Техническое сопровождение и эксплуатация, вывод из эксплуатации информационных систем и компонентов информационно-телекоммуникационной инфраструктуры.

– Предоставление программного обеспечения, инженерной, вычислительной и информационно-телекоммуникационной инфраструктуры, в том числе на основе «облачных технологий».

– Создание и развитие (модернизация) информационных систем и компонентов информационно-телекоммуникационной инфраструктуры.

1.2 Структурная схема правительства республики Хакасия и предприятия «ЦИНТ Хакасии»

Предприятие «ЦИНТ Хакасии» является подведомственной организацией аппарата правительства республики Хакасия. Само предприятие общей численностью сотрудников составляет 15 человек. В правительстве Республики Хакасия штатная численность государственных гражданских служащих, иных работников администрации главы республики Хакасия – председателя правительства республики Хакасия, исполнительных органов государственной власти республики Хакасия составляет 1556 человек. Часть структуры правительства РХ и предприятие «ЦИНТ Хакасии» представлена на рисунке 1.

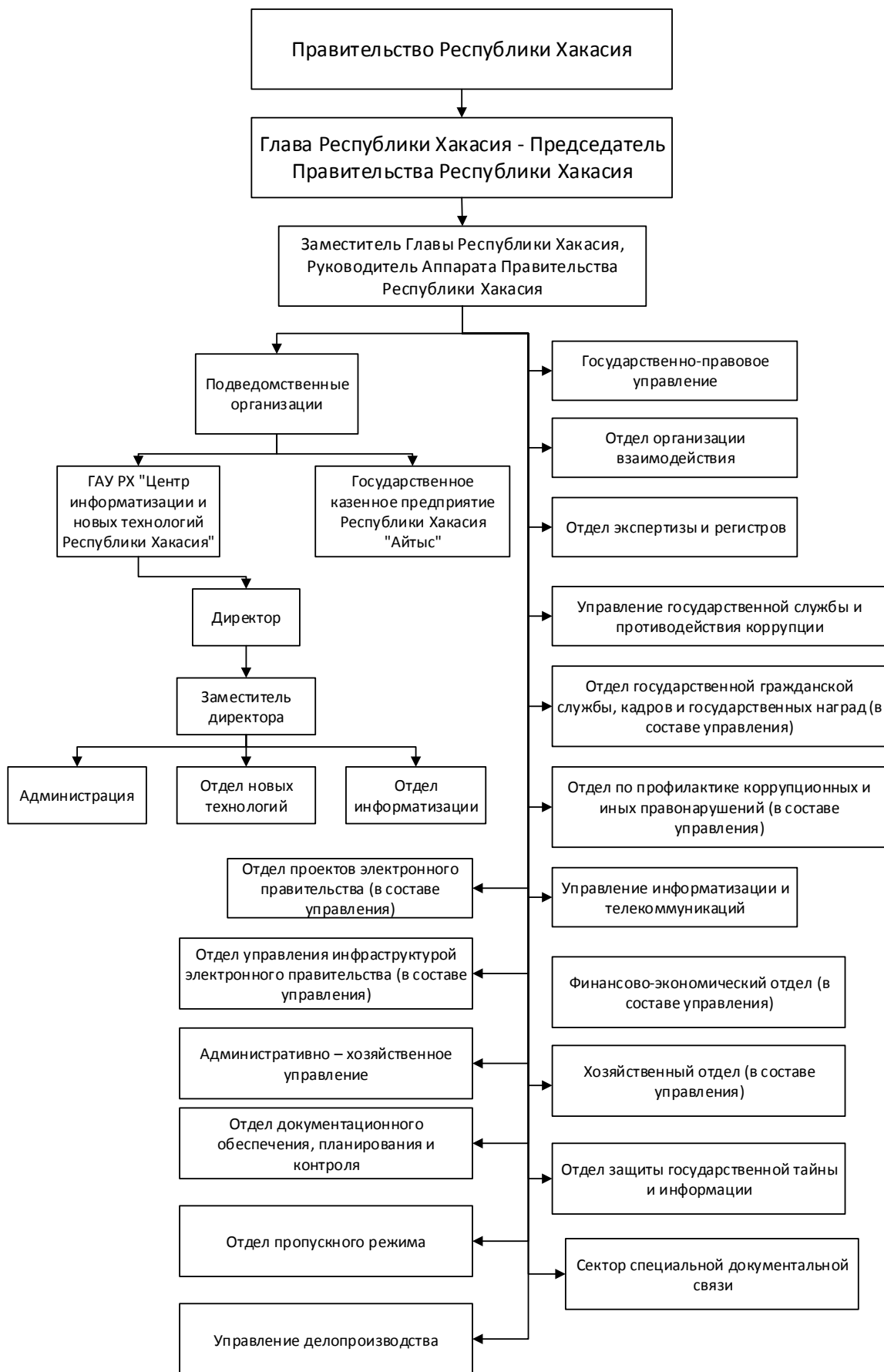


Рисунок 1 – Структурная схема правительства республики Хакасия и предприятия «ЦИНТ Хакасии»

Высшим должностным лицом Республики Хакасия является Глава Республики Хакасия – Председатель Правительства Республики Хакасия. Губернатор непосредственно координирует и контролирует деятельность:

– Заместителей Главы Республики Хакасия – Председателя Правительства Республики Хакасия.

– Администрации Главы Республики Хакасия – Председателя Правительства Республики Хакасия.

– Министерства промышленности и природных ресурсов Республики Хакасия.

– Министерства строительства и жилищно-коммунального хозяйства Республики Хакасия.

– Министерства сельского хозяйства и продовольствия Республики Хакасия.

– Министерства финансов Республики Хакасия.

– Государственного комитета по охране объектов животного мира и окружающей среды Республики Хакасия[10].

Администрация Главы Республики Хакасия является государственным органом Республики Хакасия, обеспечивающим деятельность Главы Республики Хакасия и организующим контроль за выполнением органами исполнительной власти Республики Хакасия постановлений, распоряжений и поручений Главы Республики Хакасия, а также осуществляющим содействие Главе Республики Хакасия в определении основных направлений внутренней политики республики.

Заместители Главы республики Хакасия обеспечивает координацию и контроль деятельности министерств и государственных комитетов.

Руководитель аппарата правительства Республики Хакасия организует работу по обеспечению деятельности правительства Республики Хакасия и курирует:

– Вопросы осуществления международного сотрудничества.

– Вопросы обеспечения в Республике Хакасия единого правового пространства и противодействия коррупции.

– Вопросы развития информатизации.

– Вопросы организации перехода на межведомственное взаимодействие при оказании государственных (муниципальных) услуг в Республике Хакасия.

– Вопросы обеспечения технической защиты информации в исполнительных органах государственной власти Республики Хакасия.

– Вопросы развития государственной гражданской службы Республики Хакасия и муниципальной службы в Республике Хакасия.

Также руководитель аппарата правительства является ответственным за внедрение универсальных электронных карт в Республике Хакасия и работу подведомственных организаций:

– Государственное автономное учреждение Республики Хакасия "Центр информатизации и новых технологий Республики Хакасия".

– Государственное казенное предприятие Республики Хакасия "Айтыс"[10].

Высшим должностным лицом центра информатизации и новых технологий является директор. В его подчинении находится заместитель директора, который курирует деятельность 3 отделов:

– Администрация.

– Отдел новых технологий.

– Отдел информатизации.

1.3 ИТ-инфраструктура организации «ЦИНТ Хакасии»

На предприятии «Центр информатизации и новых технологий Республики Хакасия» используются четыре информационные системы:

- Bitrix24 (Комплект инструментов для организации работы компании).

- vsDesk (Российская helpdesk система).

- 1С: Зарплата и кадры бюджетного учреждения 8.

- 1С: Бухгалтерия государственного учреждения.

Информационная система Bitrix24

Корпоративный портал – система управления внутренним информационным ресурсом компании для коллективной работы над задачами, проектами и документами, для эффективных внутренних коммуникаций[5].

Система Bitrix24 включает в себя следующие инструменты:

- Социальная сеть – единая интерактивная лента событий в корпоративном портале.

- Задачи и проекты помогают руководителям контролировать своевременное исполнение задач в подразделении, а подчиненным – не допускать нарушений.

- Документы онлайн – возможность совместного доступа к просмотру и редактированию документов.

- Чат и видеозвонки – позволяют сотрудникам оперативно обмениваться информацией и быстро принимать решения.

- Битрикс24.Диск – пространство для хранения документов и файлов компании, доступ к которому возможен с любого устройства и из любого места, где есть Интернет.

- Календари – помогают эффективно организовать рабочий день и наладить работу в команде.

- Битрикс24.Почта – готовый почтовый сервер для компании.

- CRM – повышение продаж.

- Звонки – IP-телефония.

– HR: Управление персоналом – автоматически решает тактические и стратегические задачи системы управления персоналом[5].

Информационная система vsDesk

HelpDesk (ServiceDesk) системы предназначены для автоматизации обработки запросов клиентов.

Основные особенности системы:

– Регистрация и обработка заявок и инцидентов пользователей.
– Автоматизированное изменение статусов заявок в соответствии с метриками SLA.

– Автоматизированный прием заявок по E-mail.

– Возможность прикрепления файлов и скриншотов к заявке.

– Возможность назначения групп исполнителей и наблюдателей.

– Конструктор дополнительных полей для заявок.

– E-mail и SMS уведомления о статусе заявки.

– Возможность авторизации на основе учетных записей ActiveDirectory.

– Отчеты по заявкам в разрезе заявителя и исполнителя за определенный период.

– Регистрация и классификация проблем в инфраструктуре.

– Связь проблем с инцидентами или с Конфигурационными единицами.

– Возможность автоматического сохранения записи в базе знаний по Обходному и Конечному решению проблемы.

– Возможность прикрепления файлов и скриншотов к проблеме.

– Отчеты о проблемах в разрезе Конфигурационной единицы или Сервиса.

– История жизненного цикла проблемы.

– Возможность экспорта данных отчетов в MicrosoftExcel.

- Каталог сервисов с привязкой метрик SLA и отчетом о доступности сервисов.
- База знаний с категориями, для хранения информации о решении проблем или документации.
- База Конфигурационных единиц с привязанными активами и привязкой возникших проблем.
- История изменений КЕ и привязанных активов.
- Простой конструктор типов активов с собственным набором полей.
- Экспорт в PDF и печать карточки Конфигурационной единицы и привязанных активов.
- Три роли пользователей системы "Пользователь", "Исполнитель", "Администратор" с разными интерфейсами.
- Highcharts на главной странице с показателями зарегистрированных проблем и просроченными заявками и инцидентами.
- Гибкая настройка интерфейса.
- Настройка ролей и прав пользователей, позволяющая адаптировать систему под большинство сценариев использования.
- Встроенный механизм резервного копирования.
- Настройка планировщика автоматических действий[9].

Информационная система «1С:Зарплата и кадры бюджетного учреждения 8»

Программа "1С:Зарплата и кадры бюджетного учреждения 8" - это мощный инструмент для комплексной автоматизации расчета заработной платы и ведения кадрового учета в государственных учреждениях, состоящих на самостоятельном балансе, финансируемых из федерального, регионального или местного бюджета, а также из бюджета государственного внебюджетного фонда, по следующим направлениям:

- Расчет заработной платы с поддержкой новых систем оплаты труда работников федеральных бюджетных учреждений.

- Расчет денежного содержания служащих на государственной гражданской службе.
- Исчисление регламентированных законодательством налогов и взносов с фонда оплаты труда.
- Отражение начисленной зарплаты и налогов в расходах учреждения.
- Управление денежными расчетами с работниками, включая депонирование.
- Учет кадров и анализ кадрового состава.
- Автоматизация кадрового делопроизводства.
- Набор кадров[3].

Информационная система 1С:Бухгалтерия государственного учреждения

Программный продукт «1С:Бухгалтерия государственного учреждения 8» включает технологическую платформу «1С:Предприятие 8.3» и прикладное решение (конфигурацию) «Бухгалтерия государственного учреждения».

«1С:Бухгалтерия государственного учреждения 8» обеспечивает автоматизацию бухгалтерского учета государственных учреждений, состоящих на самостоятельном балансе, финансируемых из федерального, регионального или местного бюджетов, а также из бюджета государственного внебюджетного фонда[4].

Сетевая архитектура

Сетевая архитектура представлена на рисунке 2.

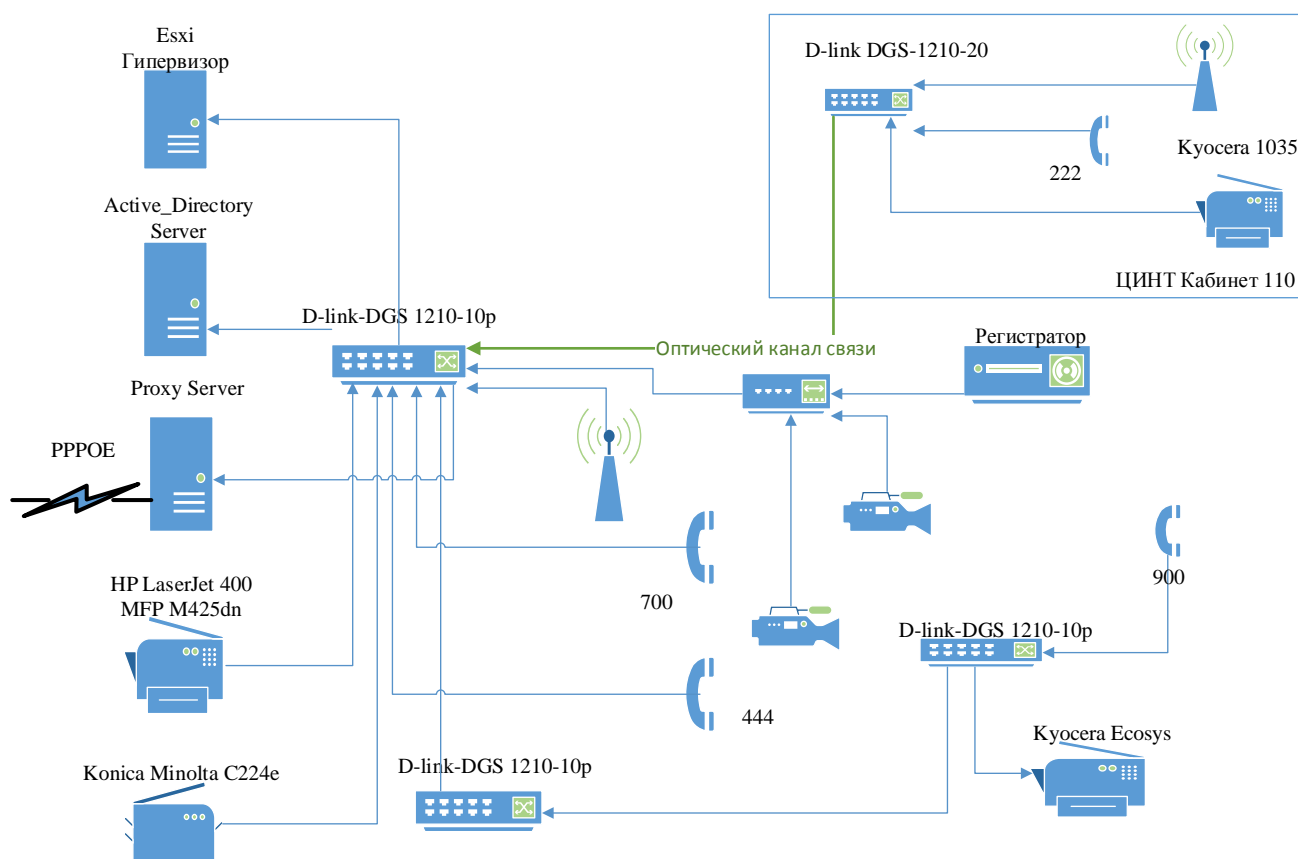


Рисунок 2 – Сетевая архитектура предприятия «ЦИНТ Хакасии»

Сетевая архитектура состоит из следующих компонентов:

1. vSphereESXiHypervisor – это аппаратный гипервизор, который устанавливается на физический сервер и разделяет его на несколько виртуальных машин. Все машины используют одни и те же физические ресурсы и могут работать одновременно. В отличие от других гипервизоров управление платформой vSphere осуществляется с помощью средств удаленного управления.

2. ActiveDirectoryServer – Доменные службы ActiveDirectory (AD DS) хранят данные каталога и управляют обменом данными между пользователями и доменами, включая процессы входа пользователей, проверку подлинности и поиск в каталоге.

3. ProxyServer – сервер в компьютерных сетях, позволяющий клиентам выполнять косвенные запросы к другим сетевым службам. Прокси-сервер

позволяет защищать компьютер клиента от некоторых сетевых атак и помогает сохранять анонимность клиента.

4. HPLaserJetPro 400 MFPM425dn – это многофункциональное устройство, которое может использоваться как производительный принтер, автоматический сканер, копир или факс.

5. KonicaMinolta C224e – многофункциональная система bizhub, включающая лазерный принтер и другие устройства «все-в-одном».

6. D-link-DGS 1210-10p – это web-smart коммутатор в который включены функции расширенного управления и безопасности, обеспечивающих лучшую производительность и масштабируемость.

7. Kyocera Ecosys – настольный принтер ECOSYS P2035d представляет собой компактное устройство с функцией двусторонней черно-белой печати в стандартной комплектации.

8. Kyocera 1035 – многофункциональное устройство формата А4.

9. 4 IP-телефона, 2 видекамеры и 2 Wi-Fi точки доступа Ubiquiti.

Взаимодействия между кабинетами происходит с помощью оптического канала связи, а для передачи данных с провайдером используется сетевой протокол канального уровня передачи кадров PPP через Ethernet.

Аппаратные средства

На предприятии «ЦИНТ Хакасии» используются следующие аппаратные средства в разном количестве:

1. Монитор Samsung LTA520HB13 (сенсорная панель d=52 дюйма).
2. Монитор TFT 17" Philips 170S6FS 450/1 12 ms.
3. Монитор 17" MONITOR Acer AL 1716S (LCD, 1280*1024).
4. Компьютер INTEL Celeron D326J с монитором LCD Samsung 710 N.
5. Принтер лазерный HP LaserJet 1018 12стр/мин 2 Мб USB2.0.
6. Коммутатор Switch/HUB D-Link 8-ports.
7. Коммутатор DGS-1008P/C1A 8x10/100/1000 Mbps Ethernet ports.
8. МФУ цветное Konica Minolta bizhub C224e.

9. Многофункциональное устройство Kyocera 1035.
10. Принтер + сканер+копир Kyocera Ecosys M2035DN.
11. Системный блок Pentium III-800-VIA.
12. Системный блок Dual - Core E6300.
13. Системный блок Int/Celeron 2.66/512/80/128/DVD+/-RW/клав.мышь.
14. Телевизор UltraHD LED 65' (165 см) LG 65UF670V (UHD, 3840x2160, DVB-T2/C/S2).
15. Ноутбук (АРМ оператора ЭВМ портативный ПК ASUS X550LB-XX181H).
16. Акустическая система AC L-AUDIO NG-15A.

Программная архитектура и прикладные программные средства

В таблице 1 представлены программные средства с описанием выполняемых функций.

Таблица 1 – Программные средства

Программные средства		Описание
1	Microsoft Windows XP Service pack 3 Pro	Полный набор заплаток по безопасности, важнейших патчей и обновлений для операционной системы Windows XP.
2	Microsoft Windows: 7, 7 Pro, 7 Enterprise, 8.1 Pro, 8.1 Enterprise, 10	Операционная система.
3	Microsoft Windows server 2012	Серверная операционная система от компании Microsoft.
4	Skype for business	Бесплатное программное обеспечение с закрытым кодом, обеспечивающее текстовую, голосовую и видеосвязь через Интернет между компьютерами.
5	Linux ubuntu	Операционная система.
6	Kaspersky Endpoint Security	Антивирусная программа, надежно защищающая файловые серверы.
7	Proxy server firewall kerio	Программный межсетевой экран, разработанный компаниями Kerio Technologies и TinySoftware.
8	VMware vSphere Hypervisor 5.0	Программный продукт для виртуализации уровня предприятия
9	VipNet	Комплексный подход к обеспечению ИБ

Окончание таблицы 1

10	Система Дело	Система электронного документооборота
11	Mozillathunderbird	Бесплатная кроссплатформенная свободно распространяемая программа для работы с электронной почтой и группами новостей.
12	MozillaFireFox	Свободный браузер, разработкой и распространением которого занимается MozillaCorporation
13	ABBYY FineReader	Программа для распознавания текста, которая позволяет быстро и точно переводить изображения документов и PDF-файлы в электронные редактируемые форматы без необходимости перепечатывания.
14	Advanced IP-scanner	Быстрый, надежный и удобный сканер IP-адресов для Windows.
15	7Zip	Свободный файловый архиватор с высокой степенью сжатия данных
16	iTunes	Медиаплеер для организации и воспроизведения музыки и фильмов, разработанный компанией Apple и бесплатно распространяемый для платформ OS X и Windows
17	TeamViewer11	Интеллектуальное решение, которое тесно интегрируется с бизнес-средой, обеспечивая унифицированный и эффективный пользовательский интерфейс.

1.4 Постановка цели и задач проектирования

Для достижения повышения эффективности работы сотрудников с системой vsDesk в области обработки запросов необходимо разработать приложение для мобильной платформы Android.

Целью проектирования:повышение эффективности работы сотрудников с системой контроля выполнения заявок vsDesk за счет организации мобильного доступа.

Задачи:

- Провести анализ работы «ЦИНТ Хакасии».
- Построить модель информационных потоковорганизации.

- Определить цель и задачи проектирования приложения для мобильной платформы Android.
- Проанализировать и выбрать средства разработки.
- Создать приложение на базе платформы Android.
- Рассчитать экономическую эффективность от внедрения проекта.

1.5 Анализ и выбор среды разработки

AndroidStudio

Официальной средой программирования под Android является AndroidStudio. Основные достоинства AndroidStudio:

- Расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана.
- Сборка приложений, основанная на Gradle.
- Различные виды сборок и генерация нескольких .apk файлов.
- Рефакторинг кода.
- Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.
- Встроенный ProGuard и утилита для подписки приложений.
- Шаблоны основных макетов и компонентов Android.
- Поддержка разработки приложений для AndroidWear и Android TV.
- Встроенная поддержка GoogleCloudPlatform, которая включает в себя интеграцию с сервисами GoogleCloudMessaging и AppEngine.
- AndroidStudio 2.1 поддерживает Android N Preview SDK, а это значит, что разработчики смогут начать работу по созданию приложения для новой программной платформы.
- Новая версия AndroidStudio 2.1 способна работать с обновленным компилятором Jack, а также получила улучшенную поддержку Java 8 и усовершенствованную функцию InstantRun.

Eclipse

Eclipse – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation. Первоначально среда была разработана для языка Java, но в настоящее время существуют многочисленные расширения для поддержки и других языков:

- C/C++ – CMD.
- Fortran – Photran.
- Perl – EPIC.
- PHP – PDT.
- JavaScript – JSEclipse.
- Python – PyDev.
- Ruby – RDT.

Embarcadero RAD Studio

Embarcadero RAD Studio – среда быстрой разработки приложений для Microsoft Windows фирмы Embarcadero Technologies. Объединяет Delphi и C++ Builder в единую интегрированную среду разработки. Начиная с версии RAD Studio XE6 поддерживает разработку приложений для Android. Для начала разработки необходимо создать новый проект Multi-Device Application и в Project Manager активизировать платформу Android.

PhoneGap

PhoneGap – бесплатный open-source, созданный Nitobi Software. Позволяет создать приложения для мобильных устройств используя JavaScript, HTML5 и CSS3, под все мобильные операционные системы. Готовое приложение компилируется в виде установочных пакетов.

NetBeans

NetBeans – свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++ и ряда других. Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведётся независимым

сообществом разработчиков-энтузиастов (NetBeansCommunity) и компанией NetBeansOrg. Для разработки программ должен быть предварительно установлен JDK подходящей версии.

В качестве среды разработки выбрана AndroidStudio, т.к. ее функционал достаточен для достижения поставленных задач, а официальная поддержка обеспечит сопровождение продукта в будущем.

1.6 Выводы по разделу «Организационно-экономическая характеристика государственного автономного учреждения Республики Хакасия «Центр информатизации и новых технологий Республики Хакасия»

В настоящий момент система автоматизации обработки запросов vsDesk не полностью удовлетворяет потребностям организации «ЦИНТ ХАКАСИИ». Для повышения эффективности работы сотрудников с системой решено разработать приложение на платформе Android.

В качестве среды разработки выбрана AndroidStudio, т.к. ее функционал достаточен для достижения поставленных задач, а официальная поддержка среды со стороны разработчиков обеспечит сопровождение продукта в будущем.

Была досконально изучена система vsDesk, выявлены ее архитектура и структура компонентов. Базовый интерфейс для разработанного приложения взят из системы vsDesk с изменениями, обусловленными переходом на другую платформу. В качестве базового функционала использованы две категории задач системы: обработка запросов и база знаний.

2 Разработка приложения для мобильной платформы Android

2.1 Технологии разработки приложения на платформе Android

Одним из способов разработки является использование IDE Android Studio с основным языком программирования Java. Этот способ позволяет создать приложение с использованием всех возможностей высокоуровневого языка программирования, в том числе использование объектной модели при разработке интерфейса. Недостаток данного способа – отличная от системы vsDesk технология проектирования, в результате чего становится невозможным использование существующего кода напрямую. Для реализации приложения использовались методы со схожим функционалом из библиотеки JDK – Java Development Kit.

Другим способом создания мобильного приложения является использование платформы Cordova. Достоинством данной системы является проектирование на языке HTML5 и PHP. Cordova – это платформа, позволяющая разрабатывать мобильные приложения на разные платформы, путем встраивания браузера в мобильное приложение. Так как система vsDesk написана с использованием web-технологий, возможно использование исходного кода для создания приложения.

Из двух вариантов была выбрана интегрированная среда разработки Android Studio для платформы Android. Преимуществом данного варианта является то, что приложение, написанное на родной платформе, обладает наибольшей стабильностью в работе. Имеет широкий функционал благодаря библиотеке JDK.

2.2 IDEAndroidStudio

2.2.1 Интерфейс AndroidStudio

Интерфейс IDE Android Studio состоит из нескольких частей (Рисунок 3). Ключевые панели обозначены на рисунке цифрами. Рассмотрим каждую из них.

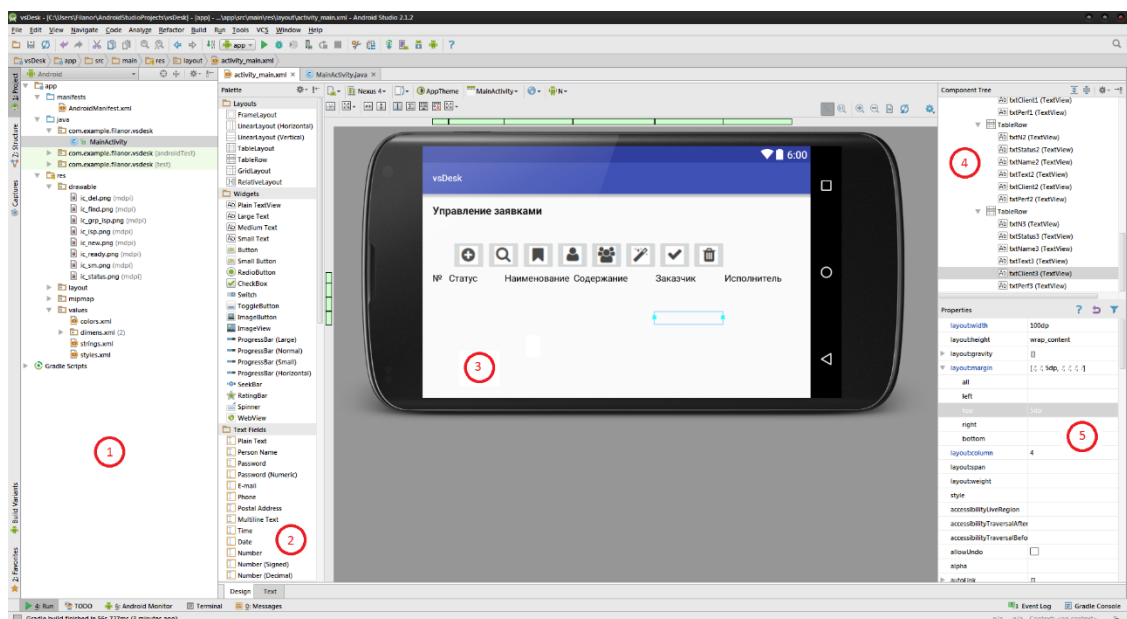


Рисунок 3 – Интерфейс IDE Android Studio

На левой стороне экрана, под цифрой 1, расположена панель структуры приложения. Она состоит из нескольких каталогов: manifests, java и res. В каталоге manifests хранится файл конфигурации приложения. В папке java хранится код нашего приложения. В ней расположены пакеты приложения, в которых хранятся все классы, объявленные в проекте. В каталоге res хранятся ресурсы, которые используются в нашем приложении.

Рассмотрим ресурсы подробнее. В папке drawable хранятся изображения. В папке layout расположены макеты экранов. В папке mipmap хранятся изображения значков приложения. Каталог values содержит описание стилей, строковых констант и размеров экрана. Ниже в директории

расположена папка GradleScripts. Она содержит файлы инструкции для автоматической системы сборки. Более подробно можно рассмотреть панель структуры на рисунке 4.

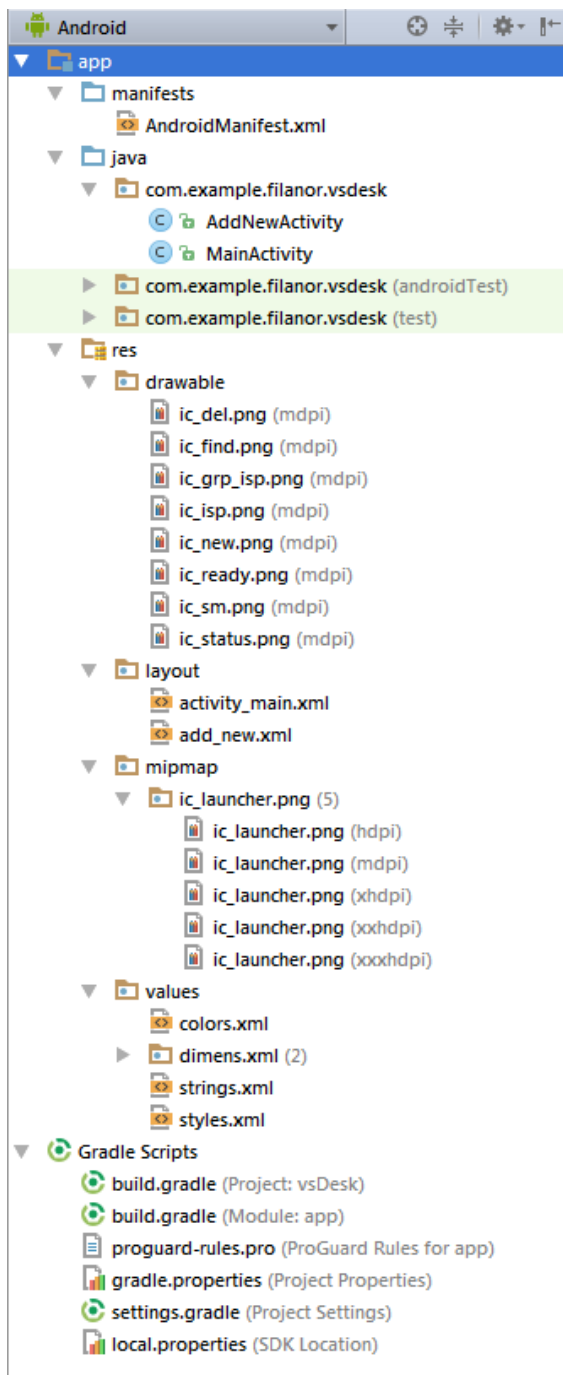


Рисунок 4 – Панель структуры приложения

Рассмотрим панель под номером 2. Она содержит палитру доступных при проектировании элементов. Они разделены на группы. Самыми первыми

идут Layouts. Это различные слои, на которых размещаются все последующие элементы. Комбинируя их, можно добиться различных видов интерфейса. За ними следуют Widgets – виджеты. В них сосредоточены элементы по выводу информации различного вида (текстовой, визуальной, прогрессивной), по взаимодействию с пользователем посредством кнопок, переключателей и флажков. Следом находятся TextFields – текстовые поля. Их роль состоит в том, чтобы дать пользователям возможность вводить информацию. Все элементы данной группы одного класса – EditText, но с разными заданными свойствами. Следующая группа – Containers. Представляет собой контейнеры – элементы, позволяющие группировать в себе другие элементы. Часть из них обладает весьма незаменимым функционалом, в частности контейнеры с возможностью прокрутки по горизонтали или вертикали. Date&Time – элементы следующей группы по взаимодействию с датой и временем. Последняя группа – Expert. В ней находятся специализированные элементы по взаимодействию с анимацией, аудио и видео.

Панель под номером 3 – это представление экрана устройства. На этой панели происходит проектирование интерфейса приложения. В зависимости от выбранных слоев проектируется расположение всех элементов на экране.

Следующая панель – панель выбранных элементов и их иерархия относительно выбранных слоев. Панель можно использовать не только в качестве структуры элементов экрана, но и проектировать элементы, располагая их по слоям.

Последняя панель – панель свойств элементов. Благодаря этой панели можно настроить любой элемент под свои нужды. В начале идут свойства, связывающие элемент со слоем: масштабирование, отступы, привязка и т.д. Затем следуют свойства конкретно элемента. Самым важным свойством является id, позволяющее идентифицировать элемент. Остальные свойства варьируются в зависимости от класса элемента.

2.2.2 Эмуляция приложения

После определенного этапа разработки возникает необходимость в тестировании приложения. В AndroidStudio представлены инструменты по эмуляции мобильного устройства, позволяющие проверить работоспособность приложения без инсталляции на конкретную машину.

Эмуляция запускается командой главного меню Run ->Run 'app' либо сочетанием клавиш «Shift + F10». Откроется окно выбора эмулятора (Рисунке 5).

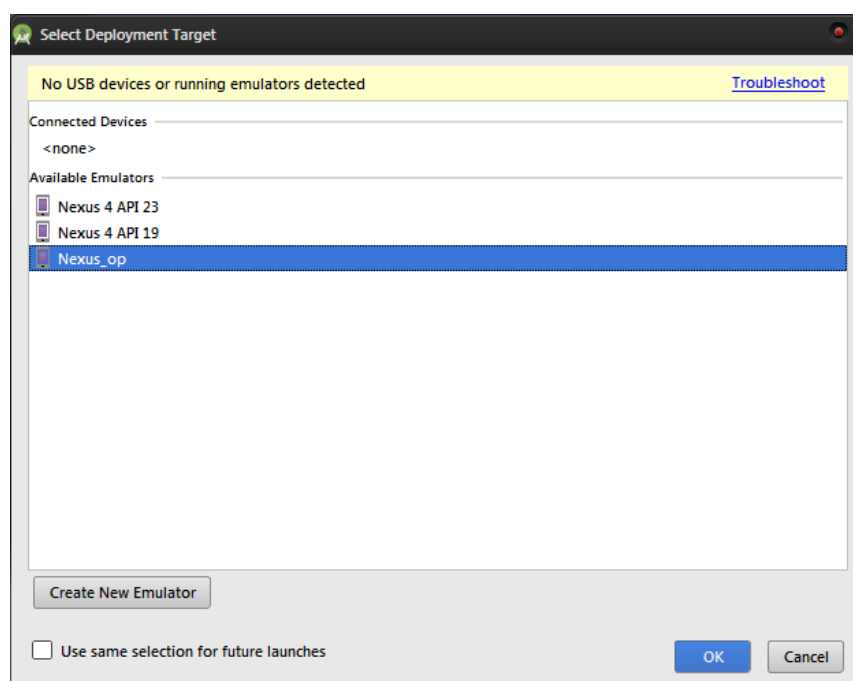


Рисунок 5 – Окно выбора эмулятора

Если есть возможность подсоединения реального мобильного устройства к компьютеру, AndroidStudio предложит эмуляцию приложения на конкретном устройстве. Если такой возможности нет, необходимо создать эмуляцию либо выбрать ее из уже существующих с заданными настройками. При разработке приложения нами создавалась новая эмуляция. Для этого выбиралась опция «CreateNewEmulator». Далее откроется окно выбора устройства конфигурации (Рисунок 6).

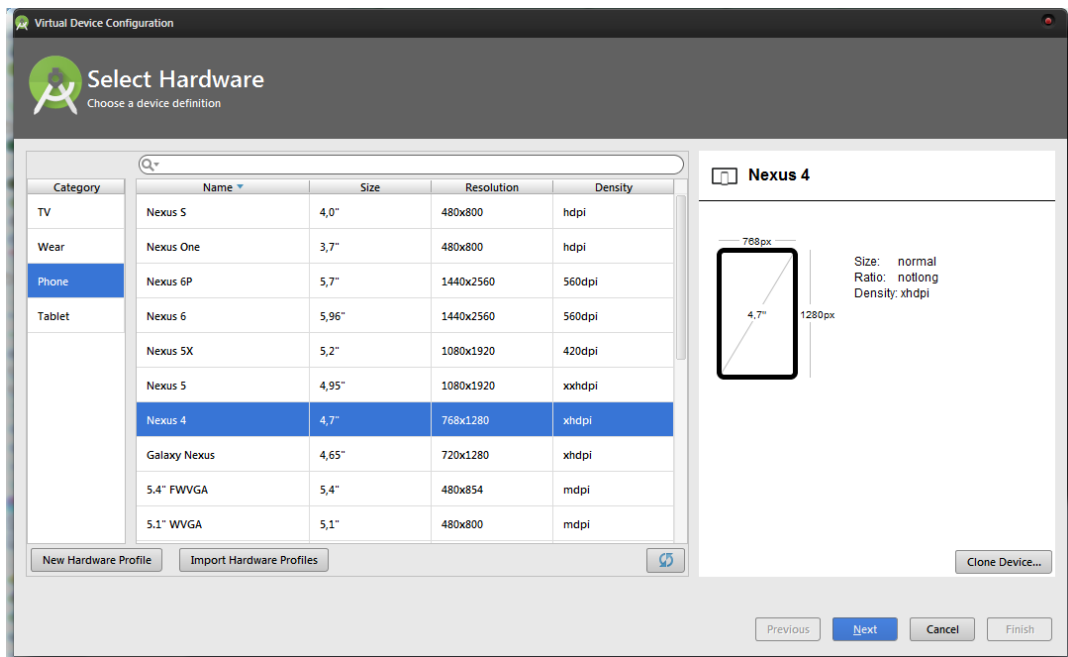


Рисунок 6 – Окно выбора устройства конфигурации

Необходимое устройство выбиралось в соответствии с требованиями к приложению. В таблице можно посмотреть параметры устройства – разрешение и диагональ экрана, а справа представлено визуально, как устройство выглядит. В нашем случае это Nexus 4. Далее откроется окно выбора версии операционной системы Android (Рисунок 7).

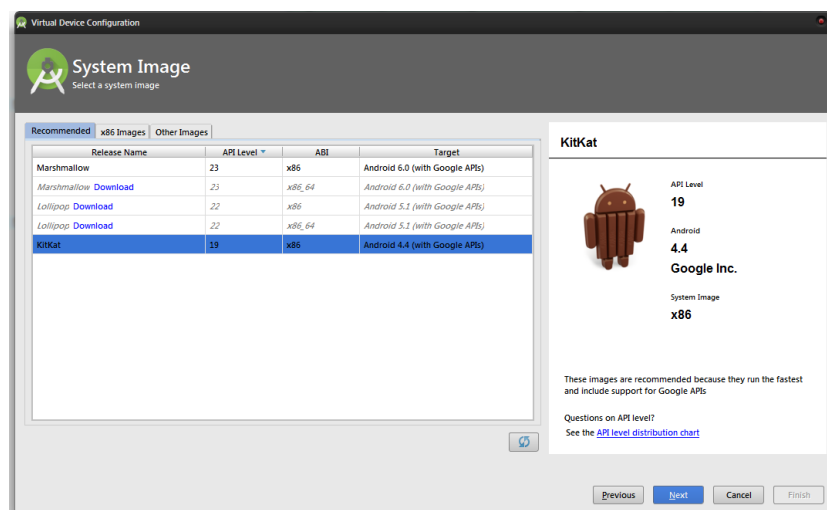


Рисунок 7 – Окно выбора версии системы Android

Из предложенных вариантов была выбрана подходящая версия. Всю необходимую информацию по выбранной системе можно посмотреть на правой панели. В нашем случае – это версия 4.4. После ее выбора откроется финальное окно настройки (Рисунок 8).

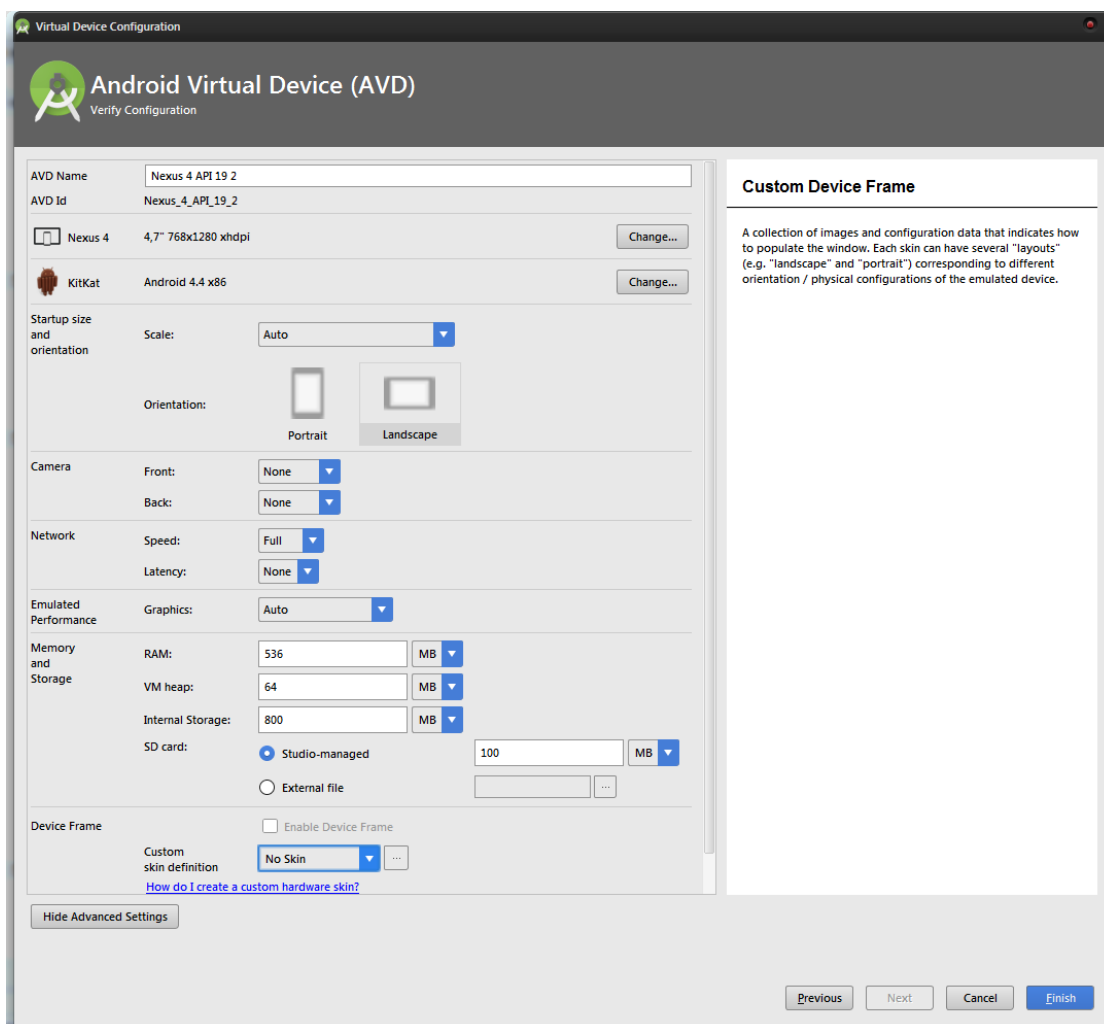


Рисунок 8 – Окно финальной настройки эмулятора

Выбираем ориентацию приложения – за это отвечает опция Orientation. В нашем случае – это Landscape. В опции выбора темы (скина) приложения Customskindefinition было выбрано NoSkin для более быстрой скорости работы. Корректируем выделение памяти под эмуляцию в зависимости от требований системы в группе опций MemoryandStorage. Также назначаем имя для эмуляции, но можно было оставить по умолчанию. IDE создала эмулятор и поместила его в список доступных в окне выбора эмулятора.

Осталось выбрать созданного нами эмулятора и запустить, нажав на кнопку «OK».

2.3 Проектирование интерфейса приложения

2.3.1 Интерфейс главного модуля

Одной из важнейших частей построения программы является проектирование интерфейса. Программа состоит из нескольких модулей, каждый из которых обладает своим интерфейсом. Рассмотрим главный модуль программы (Рисунок 9).

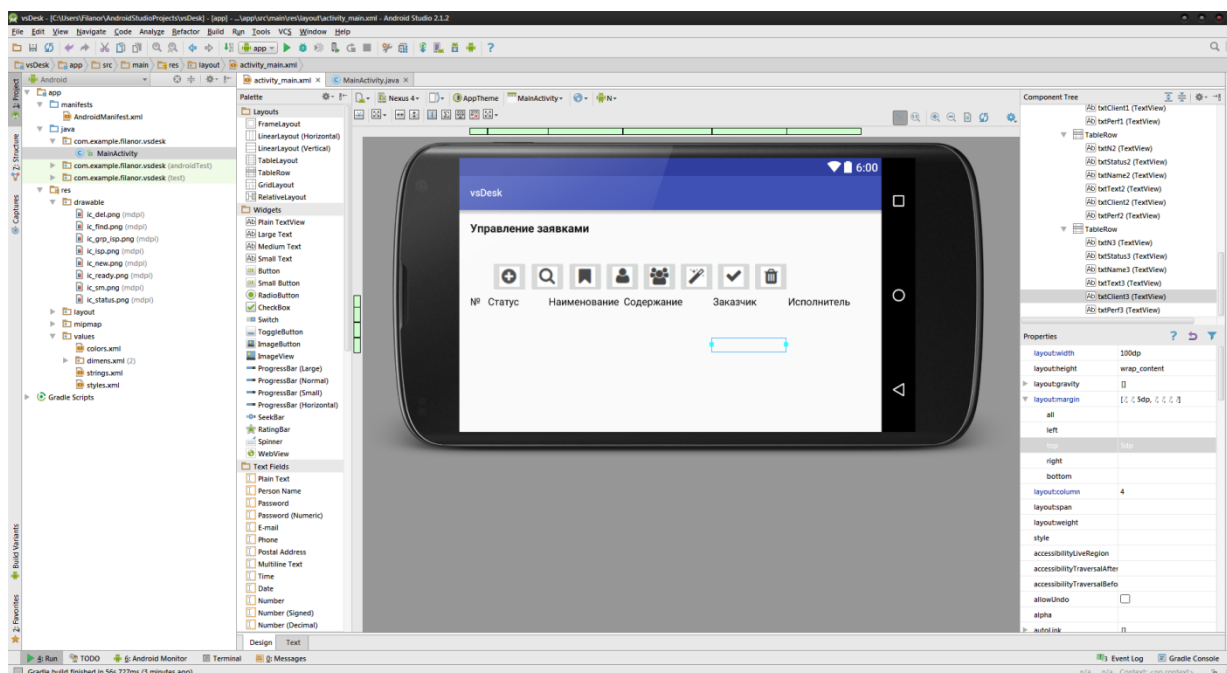


Рисунок 9 – Главный модуль программы

Рассмотрим поочередно элементы, из которых состоит интерфейс.

Вверху экрана находится надпись «Управление заявками». Элементы данного типа имеют класс `TextView`, позволяющий располагать текст без возможности его редактирования для пользователя.

Следующими элементами выступают кнопки типа ImageButton. В отличие от стандартных кнопок Button на канве ImageButton вместо текста размещается изображение. Для того чтобы загрузить изображение на канву, необходимо добавить его в ресурсы проекта, а именно в папку drawable. После этого в окне свойств компонента было выбрано – src и прописан адрес изображения: @drawable/«имя изображения».

Вся дальнейшая часть занята таблицей, в которой содержатся все зарегистрированные заявки. Т.к. в палитре компонентов Android Studio отсутствует компонент, инкапсулирующий свойства таблицы, создадим дополнительный слой класса TableLayout. Он позволяет располагать элементы в виде таблицы. В нашем случае у нас 6 столбцов: «№», «Статус», «Наименование», «Содержание», «Заказчик» и «Исполнитель». Создадим три строки. В каждой ячейке таблицы разместим компонент TextView. Он будет отвечать за вывод информации. Чтобы текст не сливался друг с другом, зададим отступ от верхней и правой ячейки в 5dp. За отступы отвечает свойство layout:margin. У него четыре атрибута для каждой из сторон: top, bottom, left, right и all. Изменим значение полей right и top. Помимо этого зададим фиксированную ширину каждого поля, чтобы таблица сохраняла фиксированный размер, не раздуваясь от слишком длинного текста. За фиксацию ширины компонента отвечает свойство layout:width. По умолчанию оно задано значением «wrap_content», что означает фиксацию по содержанию. Т.е. если длина текста превышает ширину ячейки, она растягивается, чтобы уместить в себя весь текст. При фиксации он переносится на новую строку, т.к. значение свойства layout:height задано «wrap_content» по умолчанию. Еще одним возможным значением для этого свойства является «match_parent». При этом компонент занимает всю поверхность экрана в зависимости от указанного направления.

2.3.2 Интерфейс модуля добавления заявки

Рассмотрим модуль добавления заявки (Рисунок 10).

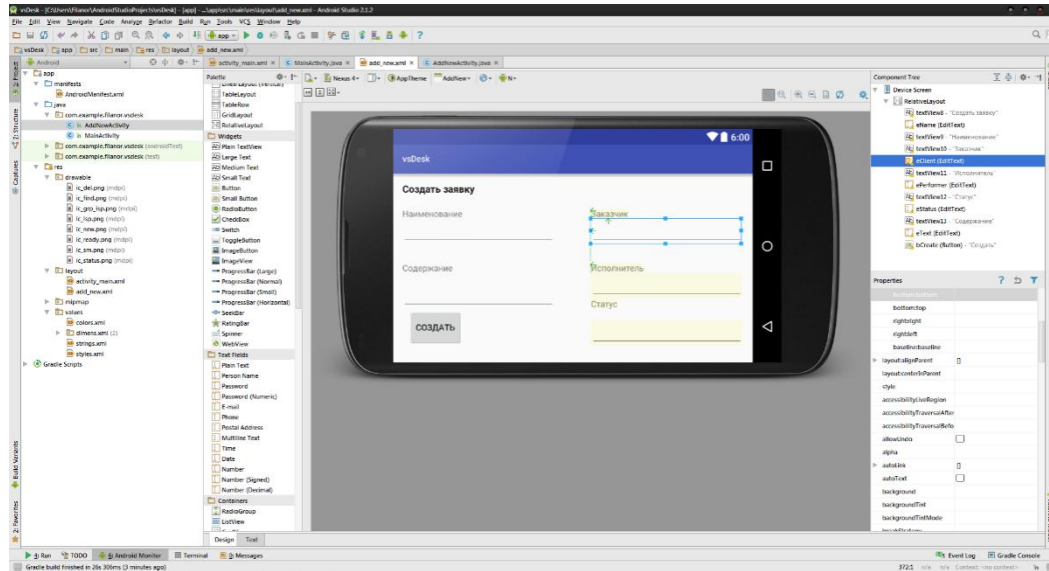


Рисунок 10 – Модуль добавления заявки

Кроме уже известных компонентов класса `TextView`, служащих для вывода текстовой информации, на экране представлены компоненты класса `EditText`. Они позволяют пользователю вносить информацию в приложение. В данном случае, это информация о новой заявке. Основным свойством класса `EditText` является `text`, в которое будет сохраняться написанный текст. Все компоненты класса `EditText` имеют фиксированную ширину, заданную свойством, описанным выше. Для компонента, сохраняющего информацию о содержании, необходимо сделать возможным перенос текста по строкам. Задается это полем `textMultiLine` свойства `inputType`. Необходимо это по той причине, что информация в данном поле может быть большого объема, и одной строки может не хватить.

2.4 Программная реализация приложения

2.4.1 Программная реализация главного модуля

В перспективе планируется подключение к локальной клиентской базе данных, с тем, чтобы брать из нее всю необходимую информацию. Но в данный момент это не представляется возможным. Поэтому, чтобы показать таблицу с заявками в работающем приложении, программно зададим несколько заявок. Для их хранения объявим в главном классе программы несколько массивов, для каждого из полей (Рисунок 11).

```
public static int[] arN;  
public static byte[] arStatus;  
public static String[] arName;  
public static String[] arText;  
public static String[] arClient;  
public static String[] arPerformer;  
public static int arCount;
```

Рисунок 11 – Объявление переменных для хранения заявок

Переменные объявляются как «public», чтобы доступ к ней возможен был из других классов программы. Параметр «static» говорит о том, что для доступа к переменной не нужно создавать экземпляр класса. Каждый из объявленных массивов будет содержать поле таблицы. Переменная arCount предназначена для хранения количества строк таблицы. Следующим шагом выделим память под массивы и заполним их значениями (Рисунок 12).

```

arN = new int [20];
arStatus = new byte [20];
arName = new String [20];
arText = new String [20];
arClient = new String [20];
arPerformer = new String [20];
arN[0] = 1;
arN[1] = 2;
arN[2] = 3;
arStatus[0] = 0;
arStatus[1] = 2;
arStatus[2] = 3;
arName[0] = "Сломался принтер";
arName[1] = "Проблема сети";
arName[2] = "Неполадки с монитором";
arText[0] = "Необходимо починить принтер";
arText[1] = "Пропап Интернет";
arText[2] = "Рябит монитор";
arClient[0] = "Ирдаев Алексей Дмитриевич";
arClient[1] = "Усулов Андрей Васильевич";
arClient[2] = "Тарашин Иван Алексеевич";
arPerformer[0] = "Арлекин Константин Игоревич";
arPerformer[1] = "Арлекин Константин Игоревич";
arPerformer[2] = "Косякин Антон Эдуардович";
arCount = 3;

```

Рисунок 12 – Реализация и заполнение массивов

Теперь рассмотрим компонентную модель, используемую в AndroidStudio. Каждый компонент имеет свойство id. С его помощью можно задать уникальное имя для каждого элемента приложения. Яване знает, какие компоненты есть в приложении, пока мы их не инициализируем. Для этого необходимо задать переменные класса того компонента, доступ к которому мы хотим получить (Рисунок 13).

```

TextView txtN[], txtStatus[], txtName[], txtText[], txtClient[], txtPerformer[];
txtN = new TextView[3];
txtStatus = new TextView[3];
txtName = new TextView[3];
txtText = new TextView[3];
txtClient = new TextView[3];
txtPerformer = new TextView[3];

```

Рисунок 13 – Объявление переменных, в которых будем хранить компоненты

Переменные объявили в виде массивов, на подобие массивов для заполнения заявок. В дальнейшем это значительно облегчит заполнение таблицы значениями. Следующий шаг инициализация каждой ячейки таблицы (Рисунок 14).

```

txtN[0] = (TextView) findViewById(R.id.txtN1);
txtN[1] = (TextView) findViewById(R.id.txtN2);
txtN[2] = (TextView) findViewById(R.id.txtN3);
txtStatus[0] = (TextView) findViewById(R.id.txtStatus1);
txtStatus[1] = (TextView) findViewById(R.id.txtStatus2);
txtStatus[2] = (TextView) findViewById(R.id.txtStatus3);
txtName[0] = (TextView) findViewById(R.id.txtName1);
txtName[1] = (TextView) findViewById(R.id.txtName2);
txtName[2] = (TextView) findViewById(R.id.txtName3);
txtText[0] = (TextView) findViewById(R.id.txtText1);
txtText[1] = (TextView) findViewById(R.id.txtText2);
txtText[2] = (TextView) findViewById(R.id.txtText3);
txtClient[0] = (TextView) findViewById(R.id.txtClient1);
txtClient[1] = (TextView) findViewById(R.id.txtClient2);
txtClient[2] = (TextView) findViewById(R.id.txtClient3);
txtPerformer[0] = (TextView) findViewById(R.id.txtPerf1);
txtPerformer[1] = (TextView) findViewById(R.id.txtPerf2);
txtPerformer[2] = (TextView) findViewById(R.id.txtPerf3);

```

Рисунок 14 – Инициализация ячеек таблицы

Разберем подробнее этот процесс. Запись представлена в следующем виде: «Имя переменной» = («Имя класса компонента») findViewById(R.id.«Идентификатор компонента»). Функция findViewById по заданному идентификатору возвращает ссылку на компонент, который приводится к классу компонента.

После этого заполним таблицу значениями из объявленных ранее массивов (Рисунок 15).

```

for (int i = 0; i < 3; i++) {
    txtN[i].setText(Integer.toString(arN[i]));
    switch (arStatus[i]) {
        case 0: txtStatus[i].setText("Открыта");
            break;
        case 1: txtStatus[i].setText("Принята");
            break;
        case 2: txtStatus[i].setText("Просрочено");
            break;
        case 3: txtStatus[i].setText("Выполнено");
            break;
    }
    txtName[i].setText(arName[i]);
    txtText[i].setText(arText[i]);
    txtClient[i].setText(arClient[i]);
    txtPerformer[i].setText(arPerformer[i]);
}

```

Рисунок 15 – Заполнение таблицы заявок

Рассмотрим код более детально. Основным оператором является цикл for. Он представлен следующим способом: for («инициализация цикла»;«условие продолжения цикла»;«инкремент или декремент») {«операторы»}. Благодаря тому, что ячейки таблицы были объявлены в качестве массивов, можно использовать цикл для их заполнения. Первым делом объявляется и сразу же инициализируется переменная i, выполняющая роль индекса в массиве. Т.к. у нас 3 строки в таблице, то ставим ограничение индекса значением 3.

Каждая ячейка обладает методом setText(), позволяющим записать в нее текст. Элементы массива arN(номер заявки) имеют целочисленный тип. Для записи их в компонент необходимо привести его к типу String. Выполняется это методом toString() класса Integer. Для определения статуса заявки используется оператор switch, позволяющий написать код для каждого из возможных значений переменной. В данном случае, исходя из значения переменной arStatus, в ячейку запишется одно из следующих возможных значений: «Открыта», «Принята», «Просрочено» или «Выполнено».

В результате в начале работы приложения таблица будет заполнена так, как представлено на рисунке 16.

№	Статус	Наименование	Содержание	Заказчик	Исполнитель
1	Открыта	Сломался принтер	Необходимо починить принтер	Ирдаев Алексей Дмитриевич	Арлекин Константин Игоревич
2	Просрочено	Проблема сети	Пропал Интернет	Усупов Андрей Васильевич	Арлекин Константин Игоревич
3	Выполнено	Неполадки с монитором	Рябит монитор	Тарашин Иван Алексеевич	Косякин Антон Эдуардович

Рисунок 16 – Приложение во время работы

Следующей функцией, реализованной в приложении, является добавление новых заявок. Для этого необходимо переключиться на другой экран. Реализовываться это будет нажатием на кнопку «плюс».

Создадим обработчик события нажатия на кнопку (Рисунок 17).

```
ImageButton imBNew = (ImageButton) findViewById(R.id.imBNew);
imBNew.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        setContentView(R.layout.add_new);
    }
});
```

Рисунок 17 – Обработчик события нажатия на кнопку

Для этого, прежде всего, необходимо инициализировать ее в языке. Методом компонента `setOnClickListener` создаем обработку события. Сам обработчик описывается структурой следом. Функция `setContentView(R.layout.add_new)` открывает на экране интерфейс модуля добавления заявок, который будет описан далее.

2.4.2 Программная реализация модуля добавления заявок

Главной задачей модуля является добавление новых заявок. Для этого на экране размещены множество полей ввода, в которые пользователи будут заносить всю необходимую информацию. Чтобы получить доступ к этой информации, необходимо инициализировать компоненты в языке (Рисунок 18).

```
public EditText tName, tStatus, tClient, tPerformer, tText;
tName = (EditText) findViewById(R.id.eName);
tStatus = (EditText) findViewById(R.id.eStatus);
tClient = (EditText) findViewById(R.id.eClient);
tPerformer = (EditText) findViewById(R.id.ePerformer);
tText = (EditText) findViewById(R.id.eText);
```

Рисунок 18 – Инициализация компонентов ввода информации

Теперь необходимо занести в переменные всю необходимую информацию. Выполняться перенос будет по нажатию кнопки «Создать». Создадим обработчик события нажатия на кнопку. В качестве хранилища информации будем использовать массивы, объявленные в классе предыдущего модуля (Рисунок 19).

```
Button bCreate = (Button) findViewById(R.id.bCreate);
bCreate.setOnClickListener((view) -> {
    MainActivity.arCount = MainActivity.arCount + 1;
    MainActivity.arN[MainActivity.arCount-1] = MainActivity.arN[MainActivity.arCount-2] + 1;
    if (tStatus.getText().toString()=="Открыта") MainActivity.arStatus[MainActivity.arCount-1] = 0; else
    if (tStatus.getText().toString()=="Принята") MainActivity.arStatus[MainActivity.arCount-1] = 1; else
    if (tStatus.getText().toString()=="Просрочено") MainActivity.arStatus[MainActivity.arCount-1] = 2; else
    if (tStatus.getText().toString()=="Выполнено") MainActivity.arStatus[MainActivity.arCount-1] = 3;
    MainActivity.arPerformer[MainActivity.arCount-1] = tPerformer.getText().toString();
    MainActivity.arClient[MainActivity.arCount-1] = tClient.getText().toString();
    MainActivity.arText[MainActivity.arCount-1] = tText.getText().toString();
    setContentView(R.layout.activity_main);
});
```

Рисунок 19 – Обработчик события нажатия на кнопку

Компоненты класса EditTextобладают методом getText(), возвращающим текст, введенный пользователем приложения. Но тип возвращаемого значения EditableText, и его необходимо преобразовать в строку. Для этого использовали метод toString(). Обращение к массивам происходит через имя класса, в котором они объявлены. После сохранения информации происходит возвращение на главный модуль функцией setContentView(), работу которой мы рассмотрели выше.

2.5 Выводы по разделу «Разработка приложения для мобильной платформы Android»

Рассмотрен интерфейс IDEAndroidStudio. Описаны все функциональные окна и особенности работы в среде. Показан процесс эмуляции мобильного приложения с заданными настройками, что является необходимым в процессе отладки приложения.

Разработано приложение на мобильной платформе Android в среде разработки Android Studio. Приложение выполняет функцию по обработке запросов системы vsDesk. При проектировании приложения возникли проблемы с подключением к базе данных, расположенной в локальной сети организации. Поэтому в приложении используются структуры данных для представления информации.

Разработан интерфейс для каждого из модулей, отличительной чертой которого является неперегруженность элементами с целью увеличения быстродействия работы. Каждый из модулей обладает своим функционалом: в главном модуле показываются все зарегистрированные заявки и представлены инструменты по работе с ними, в модуле добавления новых заявок можно дополнить уже существующие заявки.

3 Расчет показателей экономической эффективности от внедрения мобильного приложения в организацию «ЦИНТ Хакасии»

3.1 Расчет трудозатрат на разработку мобильного приложения

Трудозатраты измеряются в чел*час. Расчет производится по формуле:

$$T = t_{и} + t_{а} + t_{п} + t_{отл} + t_{д}, \quad (3.1)$$

где $t_{и}$ – затраты труда на изучение задачи;
 $t_{а}$ – затраты на разработку модели системы;
 $t_{п}$ – затраты на программирование;
 $t_{отл}$ – затраты на отладку программы;
 $t_{д}$ – затраты на подготовку документации.

Затраты труда на изучение задачи с учетом уточнения описания и квалификации программиста вычисляются по формуле:

$$t_{и} = (Q \cdot B) / (75 \dots 85) \cdot k, \quad (3.2)$$

где Q – условное число операторов в программе;
 B – коэффициент увеличения затрат в зависимости от сложности программы (1,2...5);
 K – коэффициент квалификации разработчика.

Составляющие затраты труда можно определить через условное число операторов в программном продукте. В их число входят те операторы, которые нужно учесть программисту в процессе работы над задачей с учетом возможных уточнений постановки задачи и совершенствования алгоритма.

$$Q = q \cdot c \cdot (1 + p), \quad (3.3)$$

где q – предполагаемое число операторов;

c – коэффициент сложности программы(от 1 до 2);
 p – коэффициент коррекции программы в ходе ее разработки(от 0,5до 1).

Для расчета затрат следует применить следующие значения:

$$q = 150;$$

$$c = 1,8;$$

$$p = 0,7;$$

Условное число операторов составляет:

$$Q = 150 \cdot 1,8 \cdot (1 + 0,7) = 459 \text{ операторов.}$$

Коэффициент увеличения затрат характеризует увеличение затрат труда вследствие недостаточно полного описания задачи, уточнений и некоторой доработки. Этот коэффициент может принимать значения от 1,2 до 5. Необходимо взять среднее для наиболее точных расчетов: $B = 3$

Коэффициент квалификации разработчика зависит от стажа работы программиста следующим образом:

$$\text{стаж до 2 лет} - k = 0,8;$$

$$\text{от 2 до 3 лет} - k = 1;$$

$$\text{от 3 до 7 лет} - k = 1,3...1,4;$$

$$\text{от 7 лет} - k = 1,5...1,6.$$

Так как предусмотрен небольшой набор операторов, можно будет нанимать разработчика с малым опытом работы: $k = 0,8$ (стаж работы до 2 лет)

Затраты труда на изучение задачи составят:

$$t_{и} = \frac{(459 \cdot 3)}{75 \cdot 0,8} = 22,95 \text{ чел} * \text{ час.}$$

Расчет затрат на разработку модели системы производится по формуле:

$$t_a = Q / (60 \dots 75) \cdot k, \quad (3.4)$$

где Q – условное число операторов;

k – коэффициент квалификации разработчика.

Затраты на разработку модели системы составят:

$$t_a = 459 / (70 \cdot 0,8) = 8,2 \text{ чел} \cdot \text{час.}$$

Далее необходимо рассчитать трудозатраты на отладку программы программистом.

Расчет трудозатрат на отладку производится по формуле:

$$t_{\text{отл}} = Q / (40 \dots 50) \cdot k \quad (3.5)$$

Затраты на отладку программы на компьютере составят:

$$t_{\text{отл}} = 459 / (45 \cdot 0,8) = 12,75 \text{ чел} \cdot \text{час.}$$

При комплексной отладке программы следует предусмотреть возрастающие в 1,5 раза затраты, поэтому окончательные трудовые затраты на отладку программы будут равны:

$$t_{\text{отл.ок}} = t_{\text{отл}} \cdot 1,5 \quad (3.6)$$

Затраты на окончательную отладку программы на ПК составят:

$$t_{\text{отл.ок}} = 12,75 \cdot 1,5 = 19,125 \text{ чел} \cdot \text{час.}$$

Трудовые затраты на подготовку документации включают в себя затраты труда на подготовку рукописного текста и затраты труда на редактирование, печать и оформление документации.

$$t_d = t_{дп} \cdot t_{др}, \quad (3.7)$$

где $t_{дп}$ – затраты труда на подготовку материалов в рукописи;

$t_{др}$ – затраты труда на редактирование, печать и оформление различной документации.

Трудозатраты на подготовку материалов в рукописи рассчитываются по формуле:

$$t_{дп} = Q / (150 \dots 200) \cdot k, \quad (3.8)$$

где Q – условное число операторов;

k – коэффициент квалификации разработчика.

Исходя из имеющихся данных, трудозатраты на подготовку материалов в рукописи составят:

$$t_{дп} = 459 / (175 \cdot 0,8) = 3,28 \text{ чел} \cdot \text{час}.$$

Затраты на редактирование, печать и оформление $t_{др}$ прямо пропорционально зависят от затрат на подготовку материалов в рукописи:

$$t_{др} = 0,75 \cdot t_{дп} \quad (3.9)$$

Результаты расчетов:

$$t_{др} = 0,75 \cdot 3,28 = 2,46 \text{ чел} \cdot \text{час}.$$

Таким образом, общие трудовые затраты на подготовку документации составят:

$$t_d = 3,28 + 2,46 = 5,74 \text{ чел} * \text{ час.}$$

Теперь необходимо провести расчет трудозатрат на программирование, то есть на написание исходного кода программы.

Затраты на программирование составляет примерно 20 – 30 % от общих трудозатрат на остальные этапы разработки программы.

Расчет затрат на программирование, как 25% от общих трудозатрат, посредством выражения из формулы (1) и подстановки известных значений:

$$t_{др} = (22,95 + 8,2 + 19,125 + 5,74) \cdot 0,25 = 14 \text{ чел} * \text{ час.}$$

Общие трудозатраты составят:

$$t_{др} = 22,95 + 8,2 + 19,125 + 5,74 + 14 = 70 \text{ чел} * \text{ час.}$$

Полученные трудозатраты на разработку приложения представлены в таблице 3.1.

Таблица 3.1 – Трудозатраты на разработку приложения

Вид трудозатрат	Трудоемкость, чел*час	Удельный вес, %
Затраты на изучение задачи	22,95	32,79
Затраты на разработку модели системы	8,2	11,71
Затраты на программирование	14	20
Затраты на отладку программы	19,125	27,3
Затраты на подготовку документации	5,74	8,2

Окончание таблицы 3.1

Итого	70	100
--------------	----	-----

В результате составленной таблицы наибольшее количество трудозатрат при разработке программного продукта приходится на отладку программы и изучение задачи.

3.2 Экономическая оценка разработки мобильного приложения

Для оценки экономического эффекта от внедрения мобильного приложения в организации «ЦИНТ Хакасия» необходимо рассчитать затраты на разработку.

Величина затрат на создание мобильного приложения определяется на основе метода калькуляций по отдельным статьям расходов и их последующим суммированием.

Разработка мобильного приложения состоит из нескольких стадий проектирования. В качестве разработчиков системы выступают: руководитель и программист (студент). В таблице 3.2 представлен комплекс работ по стадиям проектирования.

Таблица 3.2–Комплекс работ по стадиям проектирования мобильного приложения

Стадия проектирования	Исполнители	
	Руководитель	Программист
Теоретические исследования	+	+
Анализ требований	+	+
Проектирование	–	+
Программирование	–	+

Окончание таблицы 3.2

Тестирование	–	+
Итого	2	5

Таким образом, руководитель участвует 40%, а программист 100% в проекте. Программист принимается за единицу приведенного исполнителя, следовательно, руководитель составляет 0,4 приведенного исполнителя. Т.е. в разработке проекта участвует 1,4 приведенного человека.

$$\text{Трудоемкость} = 70/1,4 = 50 \text{ чел} * \text{ час.}$$

Срок исполнения, учитывая, что рабочий день составляет 8 ч., получается 6 дней работы над проектом.

В реальности длительность работы над проектом больше примерно на 30%, поэтому можно принять за срок разработки 8 дней.

3.2.1 Постоянные издержки

Постоянные издержки включают в себя амортизационные отчисления на компьютер и программное обеспечение и затраты на текущий ремонт.

Амортизационные отчисления

Амортизационные отчисления на компьютер и программное обеспечение производятся ускоренным методом с тем условием, что срок морального старения происходит через четыре года.

Балансовая стоимость ЭВМ включает отпускную цену, расходы на транспортировку, монтаж оборудования и его наладку и вычисляется по формуле:

$$C_{\text{бал}} = C_{\text{рын}} + Z_{\text{пр}}, \quad (3.10)$$

где $C_{бал}$ – балансовая стоимость ЭВМ, р.;

$C_{рын}$ – рыночная стоимость компьютера, р./шт.;

$Z_{пр}$ – прочие затраты (на доставку и установку, от 8 до 10% от стоимости ПК, р.).

В процессе разработки использовался персональный компьютер стоимостью 44 500 рублей. $Z_{пр} = 9\%$ от рыночной стоимости ПК.

$$Z_{пр} = 44\,500 \cdot 0,09 = 4\,005 \text{ руб.}$$

Балансовая стоимость при этом будет составлять:

$$C_{бал} = 44\,500 + 4\,005 = 48\,505 \text{ руб./шт.}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$A_{г} = C_{бал} \cdot H_{ам}/100, \quad (3.11)$$

где $A_{г}$ – сумма годовых амортизационных отчислений, р.;

$C_{бал}$ – балансовая стоимость компьютера, р./шт.;

$H_{ам}$ – норма амортизации, %.

Норма амортизации составит 25%. Итог расчета $A_{г}$:

$$A_{гЭВМ} = 48\,505 \cdot \frac{25}{100} = 12\,126,25 \text{ руб.}$$

Общая сумма амортизационных отчислений за период создания приложения равняется произведению амортизационных отчислений в день на количество дней эксплуатации компьютера и программного обеспечения:

$$A_{\text{ПЭВМ}} = \frac{12\,126,25 \cdot 8}{366} = 265,1 \text{ руб.}$$

Общая стоимость программного обеспечения, которое используется для разработки, представлена в таблице 3.3.

Таблица 3.3– Стоимость ПО, используемого для разработки

Программное обеспечение	Количество	Стоимость, руб.
ОС Microsoft Windows 7 Professional	1	9 040
Microsoft Office 2013	1	1 490
Итого		10 530

Остальные программные средства, используемые для разработки, предоставляются бесплатно.

Сумма годовых амортизационных отчислений на программное обеспечение составляет:

$$A_{\text{ПО}} = 10\,530 \cdot 0,25 = 2\,632,5 \text{ руб.}$$

Общая сумма амортизационных отчислений на программное обеспечение за период создания приложения:

$$A_{\text{ПЭВМ}} = \frac{2\,632,5 \cdot 8}{366} = 57,5 \text{ руб.}$$

Амортизация за время эксплуатации компьютера и программного обеспечения при проектировании и разработке вычисляется по формуле:

$$A_{\text{П}} = A_{\text{ЭВМ}} + A_{\text{ПО}}, \tag{3.12}$$

где $A_{эвм}$ – амортизационные отчисления на компьютер за время его эксплуатации;

$A_{по}$ – амортизационные отчисления на программное обеспечение за время его эксплуатации.

Таким образом, совокупные амортизационные отчисления за период разработки составляют:

$$A_{п} = 265,1 + 57,5 = 322,6 \text{ руб.}$$

Затраты на текущий ремонт

Затраты на текущий и профилактический ремонт принимаются равными 5% от стоимости ЭВМ, вычисляются по формуле:

$$Z_{гр} = \frac{C_{бал} \cdot P_p \cdot T_k}{366}, \quad (3.13)$$

где P_p – процент на текущий ремонт, %;

T_k – временные затраты на разработку приложения, дни.

Итог расчета затрат на текущий ремонт:

$$Z_{гр} = \frac{48\,505 \cdot 0,05 \cdot 8}{366} = 53 \text{ руб.}$$

Полученные данные составляют постоянные издержки и приведены в таблице 3.4. Большую долю в постоянных издержках занимают амортизационные отчисления на используемый компьютер и программное обеспечение.

Таблица 3.4 – Постоянные издержки

Вид постоянных издержек	Затраты, руб.	Удельный вес, %
Амортизационные отчисления	322,6	85,9
Текущий ремонт	53	14,1
Итого	375,6	100

3.2.2 Переменные издержки

Переменные издержки включают в себя:

- материальные затраты;
- затраты на потребляемую электроэнергию;
- затраты на оплату труда;
- отчисления по налогам.

Материальные затраты

К материальным затратам относятся отчисления на материалы, которые используются в процессе разработки и внедрения мобильного приложения. В стоимость затрат на материалы включаются транспортные расходы (5% от стоимости материалов).

Использованные для работы материалы представлены в таблице 3.5.

Таблица 3.5– Материальные затраты

Наименование материала	Цена, руб.	Количество	Стоимость
Бумага для печати	180	1	180
Канцелярские принадлежности	45	2	90
Заправка картриджа принтера	370	1	370
Итого			640

Затраты на материалы и покупные изделия с учетом транспортных расходов равны:

$$Z_M = 640 \cdot 1,05 = 672 \text{ руб.}$$

Затраты на электроэнергию

К затратам на электроэнергию относится стоимость потребляемой электроэнергии ЭВМ за период разработки.

Стоимость электроэнергии, потребляемой за год, рассчитывается по формуле:

$$Z_{эл} = P_{ЭВМ} \cdot T_{ЭВМ} \cdot C_{эл}, \quad (3.14)$$

где $P_{ЭВМ}$ – суммарная мощность ЭВМ, кВт;

$T_{ЭВМ}$ – время работы компьютера, часов;

$C_{эл}$ – стоимость 1 кВт/ч электроэнергии, р.

В процессе разработки использовался персональный компьютер, суммарная мощность которого равна $P_{ЭВМ} = 180$ Вт.

Цена 1 кВт/ч электроэнергии составляет $C_{эл} = 1,89$ руб.

Рабочий день составляет восемь часов. Исходя из представленных данных, необходимо рассчитать стоимость электроэнергии за период разработки:

$$Z_{эл} = 0,18 \cdot 8 \cdot 8 \cdot 1,89 = 18,1 \text{ руб.}$$

Затраты на оплату труда

Затраты на оплату труда разработчиков мобильного приложения включают в себя следующие компоненты:

- основная заработная плата разработчиков;

– дополнительная заработная плата разработчиков, которая входит в фонд заработной платы.

Расчет основной заработной платы производится на основе доли выполнения работы и величины месячного должностного оклада разработчика. Среднее количество рабочих дней в месяце равно 21.

При расчете основной заработной платы, представленном в таблице 3.7, за период разработки необходимо учесть, что руководитель участвует в разработке проекта только на этапах системного анализа и анализа требований, которые занимают 10% всего времени. Расчет оклада разработчиков за один рабочий день представлен в таблице 3.6.

Таблица 3.6 – Оклад разработчиков

Должность	Оклад в месяц, руб.	Стоимость одного рабочего дня, руб.
Руководитель	34 000	1 619
Программист	19 000	904,76

Таблица 3.7 – Расчет основной заработной платы

Исполнитель	Трудоемкость, дни.	Стоимость одного рабочего дня, руб.	Сумма, руб.
Руководитель	5	1 619	8 095
Программист	8	904,76	7 238,1
Итого			15 333,1

В дополнительную заработную плату входит районный коэффициент, равный 30% и северный коэффициент, также равный 30%.

Расчет дополнительной заработной платы производится от основной заработной платы и соответственно будет равен:

$$P = 15\,333,1 \cdot 0,6 = 9\,199,86 \text{ руб.}$$

Затраты на фонд заработной платы

Затраты на фонд заработной платы приведены в таблице 3.8.

Таблица 3.8 – Расчет фонда заработной платы за период разработки

Вид заработной платы	Руководитель, руб.	Программист, руб.	Сумма, руб.	Удельный вес
Основная	8 095	7 238,1	15 333,1	62,5
Дополнительная	4 857	4 342,86	9 199,86	37,5
Итого	12 952	11 580,96	24 532,96	100

Отчисления по налогам

Отчисления по налогам включают отчисления по единому социальному налогу (ЕСН), который составляет 30%. Затраты по данной статье определяются установленным нормативом от расходов на оплату труда работников, непосредственно занятых созданием, приведены в таблице 3.9 и вычисляются по формуле:

$$Z_{пн} = Z_{фз} \cdot N_{есн}, \quad (3.15)$$

где $Z_{пн}$ – затраты по ЕСН, р.;

$Z_{фз}$ – фонд заработной платы, р.;

$N_{ЕСН}$ – вид налоговых отчислений, %.

Таблица 3.9 – Отчисления по ЕСН

Исполнитель	Фонд заработной платы, руб.	Общая сумма ЕСН, руб.
Руководитель	12 952	3 885,6
Программист	11 580,96	3 474,288
Итого	24 532,96	7 359,888

Далее в таблице 3.10 приведен список переменных издержек.

Таблица 3.10– Переменные издержки

Вид переменных издержек	Величина, руб.	Удельный вес, %
Материальные затраты	640	1,97
Затраты на электроэнергию	18,1	0,06
Затраты на оплату труда	24 532,96	75,37
Отчисления по налогам	7 359,888	22,61
Итого	32 550,948	100

Далее необходимо рассчитать общие затраты. На эту статью относятся все издержки, которые были произведены при создании мобильного приложения.

Полная себестоимость разработки определяется суммированием постоянных и переменных издержек и вычисляется по формуле:

$$Z_{об} = Z_{пос} + Z_{пер}, \quad (3.16)$$

где $Z_{об}$ – себестоимость приложения;

$Z_{пос}$ – постоянные издержки;

$Z_{пер}$ –переменные издержки.

Таким образом, себестоимость создаваемого мобильного приложения равна $Z_{об} = 32\,926,548$ руб.

В таблице 3.11 приведена структура полных издержек. В результате, при создании программного продукта наибольший удельный вес занимают переменные издержки.

Таблица 3.11 – Структура полных издержек

Вид издержек	Величина, руб.	Удельный вес, %
Постоянные	375,6	1,14

Окончание таблицы 3.11

Переменные	32 550,948	98,86
Итого	32 926,548	100

В таблице 3.12 показано процентное соотношение всех категорий затрат на разработку приложения.

Таблица 3.12 – Соотношение категорий затрат на разработку приложения

Вид затрат	Величина, руб.	Удельный вес, %
Материальные затраты	640	1,94
Затраты на электроэнергию	18,1	0,05
Затраты на оплату труда	24 532,96	74,5
Отчисления по налогам	7 359,888	22,35
Амортизационные отчисления	322,6	0,98
Текущий ремонт	53	0,16
Итого	32 926,548	100

3.3 Расчет эффективности внедрения для организации «ЦИНТ Хакасии»

Значение экономической эффективности от использования разработанного приложения определяется по формуле:

$$\mathcal{E} = Z_6 - Z_9, \quad (3.17)$$

где Z_6 – затраты по базовому варианту (руб./год);

Z_9 – затраты при использовании программного обеспечения (руб./год).

Затраты по базовому варианту рассчитываются по формуле:

$$Z_6 = CЧ_{п} \cdot T_p \cdot (1/d_{зп}), \quad (3.18)$$

где $CЧ_{п}$ – часовая средняя заработная плата (457,1 руб./час);

T_p – трудоёмкость решаемой задачи;

$d_{зп}$ – доля заработной платы в общих затратах организации (75,3%).

Для решения задачи без использования разрабатываемого программного продукта необходимо примерно 40% действующего фонда рабочего времени ЭВМ, который рассчитывается по формуле:

$$T_{пк} = N_m \cdot N_d \cdot N_{ч}, \quad (3.19)$$

где N_m – количество месяцев в году (12);

N_d – количество рабочих дней в месяце (21);

$N_{ч}$ – средняя продолжительность рабочего дня (8 часов).

Действительный годовой фонд времени ЭВМ равен:

$$T_{пк} = 12 \cdot 21 \cdot 8 = 2016 \text{ часов/год}$$

Расчет 40% от результата:

$$T_p = 2016 \cdot 0,4 = 806,4 \text{ часов/год}$$

Затраты по базовому варианту составят:

$$Z_6 = 457,1 \cdot 806,4 \cdot \left(\frac{1}{0,75}\right) = 483\,090,4 \text{ руб./год}$$

Затраты при использовании разработанного приложения рассчитываются по формуле:

$$Z_3 = (T_r \cdot C_m + Z_p) / T_c, \quad (3.20)$$

где T_r —время, отводимое на работу с программой;
 C_m —стоимость одного машинного часа;
 Z_p —эксплуатационные затраты при использовании ПО;
 T_c — срок службы программного обеспечения.

Цена машино-часа измеряется руб./час.и рассчитывается по формуле:

$$C_m = (Z_a + Z_m + Z_{тр}) / T_{пер}, \quad (3.21)$$

где Z_a — затраты на амортизацию, руб.;
 $Z_{тр}$ — затраты на текущий ремонт компьютера, руб.;
 Z_m — затраты на материалы, руб.;
 $T_{пер}$ —период разработки программного продукта.

Стоимость 1 машино-часа составит:

$$C_m = (322,6 + 53 + 640) / (8 * 8) = 15,87 \text{ руб/час.}$$

Эксплуатационные расходы на разработку —общие расходы на разработку программного продукта $Z_p = 32\,926,548$ руб.

Срок службы программы в среднем равен $T_c = 5$ лет.

Затраты при использовании программы будут равны:

$$Z_3 = \frac{2016 \cdot 15,87 + 32\,926,548}{5} = 12984,1 \text{ руб./год}$$

Значение экономической эффективности от использования программного продукта будет составлять:

$$\mathcal{E} = 483\,090,4 - 12984,1 = 470\,106,3 \text{ руб./год}$$

После определения годового экономического эффекта необходимо рассчитать срок окупаемости затрат на разработку проекта по формуле:

$$T_{\text{ок}} = \frac{K}{\mathcal{E}}, \quad (3.22)$$

где K –затраты на разработку проекта, руб.;

\mathcal{E} – годовая эффективность, руб.

Срок окупаемости затрат на разработку программного продукта составит:

$$T_{\text{ок}} = \frac{32\,926,548}{470\,106,3} = 0,07 \text{ года} = 0,84 \text{ месяца} = 3 \text{ недели}$$

Таким образом, срок окупаемости проекта составляет примерно 3 недели. Малый срок объясняется тем, что проект не включает сложных программных разработок.

Коэффициент экономической эффективности разработки (E_{ϕ}):

$$E_{\phi} = \frac{\mathcal{E}}{K} \quad (3.23)$$

Нормативное значение коэффициента эффективности капитальных вложений $E_n = 0,33$, если $E_{\phi} > E_n$, то делается вывод об эффективности капитальных вложений.

Коэффициент экономической эффективности разработки (E_{ϕ}) равен:

$$E_{\phi} = \frac{470\,106,3}{32\,926,548} = 14,28$$

Так как $E\phi = 14,28 > E_n$, то разработка и внедрение разрабатываемого продукта является эффективным, т.е. эффект от использования мобильного приложения окупает все затраты, связанные с проектированием и эксплуатацией. В таблице 3.13 приведены сводные данные экономического обоснования разработки и внедрения проекта.

Таблица 3.13 – Таблица показателей экономической эффективности

Показатель	Величина
Трудозатраты на разработку проекта, чел*час	70
Затраты на разработку проекта, руб.	32 926,548
Экономическая эффективность проекта, руб.	470 106,3
Коэффициент экономической эффективности	14,28
Срок окупаемости, мес.	0,84

Таким образом, произведенный экономический анализ эффективности создания и эксплуатации приложения доказывает целесообразность его использования в организации «ЦИНТ Хакасии».

3.4 Выводы по разделу «Расчет показателей экономической эффективности от внедрения мобильного приложения в организацию «ЦИНТ Хакасии»

Общие трудозатраты на разработку мобильного приложения составляют 70 чел*час.

Затраты на разработку мобильного приложения составляют 32 926,548рублей.

Экономическая эффективность от использования увеличивается по сравнению с базовым вариантом и составляет 470 106,3 руб/год.

Срок окупаемости проекта составляет примерно 3 недели.

Так как $E\phi = 14,28 > E_n$, то разработка и внедрение разрабатываемого продукта является эффективным, т.е. эффект от использования мобильного приложения окупает все затраты, связанные с проектированием и эксплуатацией.

ЗАКЛЮЧЕНИЕ

В организации «ЦИНТ Хакасии» в настоящий момент используется система автоматизации обработки запросов vsDesk. С целью повышения эффективности работы сотрудников с системой была поставлена задача разработать мобильное приложение на платформе Android.

В качестве среды разработки выбрана AndroidStudio, обладающая широкими возможностями в проектировании приложений благодаря наличию объектно-ориентированного языка программирования Java. Выбор среды обусловлен также тем, что среда официально поддерживается разработчиками, что обеспечит сопровождение продукта в будущем.

Досконально изучена система vsDesk, выявлены структура разделов, функциональные особенности выбранных задач и влияние групп пользователей на интерфейс системы. При разработке мобильного приложения будут использованы две категории задач системы: обработка запросов и база знаний.

Рассмотрена структура IDEAndroidStudio, все ее функциональные окна и особенности тестирования мобильного приложения. Разработку мобильного приложения можно разделить на две части: проектирование интерфейса и написание программного кода, реализацию интерфейса. Каждый этап разработки детально описан, программный код представлен в приложениях.

Рассчитаны показатели экономической эффективности от внедрения приложения в организацию. Общие трудозатраты на разработку составляют 70 чел в час. Затраты на разработку приложения – 32 926,548 рублей. По сравнению с базовым вариантом увеличивается экономическая эффективность, и составляет 470 106,3 руб в год. Срок окупаемости проекта – примерно 3 недели. По всем показателям разработка и внедрение мобильного приложения является эффективным.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гусева, Т.И. Проектирование баз данных в примерах и задачах / Т.И. Гусева. – М.: Радио и связь, 1992. – 160 с.
2. Джексон, Г.А. Проектирование реляционных баз данных для использования с микро – ЭВМ / Г.А. Джексон. – М.: Мир, 1991. – 197 с.
3. 1С:Зарплата и кадры бюджетного учреждения 8, общие сведения [Электронный ресурс]: – © ООО «1С» 1С:Предприятие8 система программ.– Режим доступа: <http://v8.1c.ru/budghrm/>
4. Бухгалтерия государственного учреждения, общие сведения [Электронный ресурс]: – © ООО «1С» 1С:Предприятие8 система программ.– Режим доступа: <http://v8.1c.ru/stateacc/>
5. Возможности Битрикс24 [Электронный ресурс]: – © 2001-2016 «Битрикс», «1С-Битрикс». Работает на «1С-Битрикс: Управление сайтом».– Режим доступа: <https://www.bitrix24.ru/>
6. Васильев, А.Н. Java / А.Н. Васильев. – СПб.: Питер, 2011. – 270 с.
7. Методология IDEF [Электронный ресурс]: – Википедия, Свободная энциклопедия Wikipedia®. – Режим доступа: <https://ru.wikipedia.org/wiki/IDEF>
8. Сведения о компании [Электронный ресурс]:– 2016., Центр информатизации и новых технологий. – Режим доступа: <http://cintrh.ru/company/>
9. О системе vsDesk [Электронный ресурс]: –Copyright © 2012-2016 Иванов Александр. vsDesk самый простой helpdesk. – Режим доступа: <http://vsdesk.ru/main/about>
10. Органы власти [Электронный ресурс]: – © 2014-2016 Официальный портал Правительства Республики Хакасия. – Режим доступа: <http://www.r-19.ru/authorities/>

ПРИЛОЖЕНИЕ А

Программный код модуля MainActivity.java

```
package com.example.filanor.vsdesk;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    public static int[] arN;
    public static byte[] arStatus;
    public static String[] arName;
    public static String[] arText;
    public static String[] arClient;
    public static String[] arPerformer;
    public static int arCount;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView txtN[], txtStatus[], txtName[], txtText[], txtClient[],
txtPerformer[];
        txtN = new TextView[3];
        txtStatus = new TextView[3];
        txtName = new TextView[3];
        txtText = new TextView[3];
```

```
txtClient = new TextView[3];
txtPerformer = new TextView[3];
arN = new int [20];
arStatus = new byte [20];
arName = new String [20];
arText = new String [20];
arClient = new String [20];
arPerformer = new String [20];
arN[0] = 1;
arN[1] = 2;
arN[2] = 3;
arStatus[0] = 0;
arStatus[1] = 2;
arStatus[2] = 3;
arName[0] = "Сломался принтер";
arName[1] = "Проблема сети";
arName[2] = "Неполадки с монитором";
arText[0] = "Необходимо починить принтер";
arText[1] = "Пропал Интернет";
arText[2] = "Рябит монитор";
arClient[0] = "Ирдаев Алексей Дмитриевич";
arClient[1] = "Усупов Андрей Васильевич";
arClient[2] = "Тарашин Иван Алексеевич";
arPerformer[0] = "Арлекин Константин Игоревич";
arPerformer[1] = "Арлекин Константин Игоревич";
arPerformer[2] = "Косякин Антон Эдуардович";
arCount = 3;
txtN[0] = (TextView) findViewById(R.id.txtN1);
txtN[1] = (TextView) findViewById(R.id.txtN2);
txtN[2] = (TextView) findViewById(R.id.txtN3);
```

```

txtStatus[0] = (TextView) findViewById(R.id.txtStatus1);
txtStatus[1] = (TextView) findViewById(R.id.txtStatus2);
txtStatus[2] = (TextView) findViewById(R.id.txtStatus3);
txtName[0] = (TextView) findViewById(R.id.txtName1);
txtName[1] = (TextView) findViewById(R.id.txtName2);
txtName[2] = (TextView) findViewById(R.id.txtName3);
txtText[0] = (TextView) findViewById(R.id.txtText1);
txtText[1] = (TextView) findViewById(R.id.txtText2);
txtText[2] = (TextView) findViewById(R.id.txtText3);
txtClient[0] = (TextView) findViewById(R.id.txtClient1);
txtClient[1] = (TextView) findViewById(R.id.txtClient2);
txtClient[2] = (TextView) findViewById(R.id.txtClient3);
txtPerformer[0] = (TextView) findViewById(R.id.txtPerf1);
txtPerformer[1] = (TextView) findViewById(R.id.txtPerf2);
txtPerformer[2] = (TextView) findViewById(R.id.txtPerf3);
for (inti = 0;i<3;i++){
txtN[i].setText(Integer.toString(arN[i]));
switch (arStatus[i]) {
case 0:txtStatus[i].setText("Открыта");
break;
case 1:txtStatus[i].setText("Принята");
break;
case 2:txtStatus[i].setText("Просрочено");
break;
case 3:txtStatus[i].setText("Выполнено");
break;
}
txtName[i].setText(arName[i]);
txtText[i].setText(arText[i]);
txtClient[i].setText(arClient[i]);

```

```
txtPerformer[i].setText(arPerformer[i]);
    }
ImageButtonimBNew = (ImageButton) findViewById(R.id.imBNew);
imBNew.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        setContentView(R.layout.add_new);
    }
});
}
}
```

ПРИЛОЖЕНИЕ Б

Программный код модуля AddNewActivity.java

```
package com.example.filanor.vsdesk;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class AddNewActivity extends AppCompatActivity {
    public EditText tName, tStatus, tClient, tPerformer, tText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.add_new);
        tName = (EditText) findViewById(R.id.eName);
        tStatus = (EditText) findViewById(R.id.eStatus);
        tClient = (EditText) findViewById(R.id.eClient);
        tPerformer = (EditText) findViewById(R.id.ePerformer);
        tText = (EditText) findViewById(R.id.eText);
        Button bCreate = (Button) findViewById(R.id.bCreate);
        bCreate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                MainActivity.arCount = MainActivity.arCount + 1;
                MainActivity.arN[MainActivity.arCount-1] =
                MainActivity.arN[MainActivity.arCount-2] + 1;
```

```

if (tStatus.getText().toString()=="Открыта")
MainActivity.arStatus[MainActivity.arCount-1] = 0; else
if (tStatus.getText().toString()=="Принята")
MainActivity.arStatus[MainActivity.arCount-1] = 1; else
if (tStatus.getText().toString()=="Просрочено")
MainActivity.arStatus[MainActivity.arCount-1] = 2; else
if (tStatus.getText().toString()=="Выполнено")
MainActivity.arStatus[MainActivity.arCount-1] = 3;
MainActivity.arPerformer[MainActivity.arCount-1] =
tPerformer.getText().toString();
MainActivity.arClient[MainActivity.arCount-1] = tClient.getText().toString();
MainActivity.arText[MainActivity.arCount-1] = tText.getText().toString();
setContentView(R.layout.activity_main);
}
});
}
}

```

Бакалаврская работа выполнена мной самостоятельно. Используемые в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в 1 экземпляре.

Библиография 10 наименований.

« _____ » июня 2016 г.

(дата)

(подпись)(ФИО)

П.А. Бредюк