

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

Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А. И. Рубан
подпись
« _____ » _____ 2016 г.

ДИПЛОМНЫЙ ПРОЕКТ

230105.65 Программное обеспечение вычислительной техники и автоматизиро-
ванных систем

Мобильное приложение для планирования задач предприятия

Пояснительная записка

Научный руководитель _____ доцент, к. т. н. А. С. Кузнецов
подпись, дата

Нормоконтролер _____ доцент, к. т. н. О. А. Антамошкин
подпись, дата

Выпускник _____ А. И. Гусевский
подпись, дата

Красноярск 2016

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

Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А. И. Рубан
подпись
« _____ » _____ 2016 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме дипломного проекта**

Студенту Гусевский Александр Игоревич

Группа ЗКИ10-05 Направление (специальность) 230105.65, Программное обеспечение вычислительной техники и автоматизированных систем.

Тема выпускной квалификационной работы: «Мобильное приложение для планирования задач предприятия».

Утверждена приказом по университету № 4202/с от 28.03.16.

Руководитель ВКР А. С. Кузнецов, доцент кафедры «Информатика», канд. техн. наук.

Исходные данные для ВКР: спроектировать и разработать мобильное приложения для планирования задач предприятия.

Перечень разделов ВКР:

- введение;
- анализ предметной области;
- проектирование программного обеспечения;
- описание приложения;
- эргономика.

Перечень графического или иллюстративного материала с указанием основных чертежей, плакатов, слайдов: презентационные слайды Power Point.

Руководитель ВКР

(подпись)

А. С. Кузнецов

Задание принял к исполнению

(подпись)

А. И. Гусевский

« ____ » _____ 2016 г.

АННОТАЦИЯ

Выпускная квалификационная работа по теме «Мобильное приложение для планирования задач предприятия» содержит 48 страниц текстового документа, включая 2 приложения, 24 рисунка, 11 библиографических источников.

Целью работы являлась разработка мобильного приложения для планирования задач предприятия.

В дипломный проект входит введение, четыре главы и заключение.

Во введении определяется необходимость реализации и основные задачи проекта.

В первой главе более анализируется предметная область и приводится сравнение аналогичного программного обеспечения.

Во второй главе описывается проектирование программного продукта: выбор методологии, средств разработки, применение экспертного подхода.

В третьей главе описывается функциональность приложения.

В четвертой главе затронута тема эргономики приложения.

Заключение посвящено подведению итогов по всей проделанной работе.

					<i>ДП-230105.65 ПЗ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	Мобильное приложение для планирования задач предприятия	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Разраб.</i>		Гусевский А.И.						
<i>Провер.</i>		Кузнецов А.С.					4	48
<i>Н. Контр.</i>		Антамошкин О.А.				<i>Кафедра «Информатика»</i>		
<i>Утверд.</i>		Рубан А.И.						

Оглавление

ВВЕДЕНИЕ.....	6
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1 Этапы анализа	7
1.2 Характеристики объекта автоматизации	7
1.3 Назначение системы	8
1.4 Исполнители производственных и организационных работ	9
1.5 Выбор средств описания бизнес-процессов разработки	9
1.6 Анализ рынка мобильных приложений для планирования деятельности.....	12
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	17
2.1 Выбор модели жизненного цикла	17
2.2 Выбор платформы	19
2.3 Выбор средств разработки	21
2.4 Архитектура программного продукта	24
2.5 Проектирование модели IDEF0.....	27
3 ОПИСАНИЕ ПРИЛОЖЕНИЯ	30
3.1 Создание задач и подзадач.....	30
3.2 Контекстное меню задачи.....	31
3.3 Поиск задачи.....	32
3.4 Фильтры.....	32
3.5 Напоминания.....	33
3.6 Синхронизация	34
4 ЭРГОНОМИКА	36
4.1 Концепция пользовательского интерфейса	36
4.2 Эргономика интерфейса	37
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	40
ПРИЛОЖЕНИЕ 1. ОРГАНИЗАЦИОННАЯ СТРУКТУРА ПРЕДПРИЯТИЯ	41
ПРИЛОЖЕНИЕ 2. ДОКУМЕНТИРОВАННЫЙ ЛИСТИНГ ПРОГРАММНЫХ МОДУЛЕЙ.....	42

ВВЕДЕНИЕ

Планирование на предприятии — экономический метод управления, представляет собой процесс проектирования желаемого будущего, а также эффективных путей его достижения. Задачи планирования заключаются в выявление перспектив изменения внешнего окружения фирмы, формирование целей и стратегии развития, определение первоочередных задач и действий для их решения.

Постоянная неопределенность будущего является одной из причин, по которой планирование должно осуществляться непрерывно. В силу внешних изменений или ошибок управления события могут разворачиваться не так, как предполагало руководство при выработке планов. Поэтому планы необходимо пересматривать, чтобы они согласовывались с реальностью. Там, где есть планирование, неопределенность сокращается.

Обеспечить непрерывность и добавить новые возможности планирования помогут мобильные телефоны, которые давно перестали быть чем-то необычным и отлично справляются со своими функциями. По сути, все смартфоны стали небольшой копией компьютера, который постоянного можно иметь при себе.

В рамках данного дипломного проекта будет описано создание приложения для мобильного телефона, позволяющее планировать рабочий процесс, а так же контролировать выполнение задач в любом месте и в любое время.

Таким образом, определим основные задачи дипломного проекта:

1. Анализ предметной области и существующих решений;
2. Выбор методологии и средств разработки;
3. Анализ требований к программному обеспечению;
4. Проектирование и разработка программного обеспечения, удовлетворяющего основным требованиям.

					ДП – 230105.65 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		6

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Этапы анализа

При разработке программного продукта для обеспечения его максимальной эффективности очень важно учитывать все особенности информационной среды, в которой будет происходить эксплуатация. Именно поэтому на стадии анализа необходимо уделить большое внимание описанию предметной области функционирования продукта. Анализ предметной области является первым этапом построения информационной системы. Здесь осуществляется сбор информации об этой области, установление источников информации, выделение функциональных областей и процессов, происходящих в этих областях.

Анализ предметной области проводился в несколько этапов:

- выделение функциональных областей;
- выявление и анализ процессов, происходящих в каждой функциональной области;
- анализ информационных потоков (источники и потребители информации).

Программный продукт является дополнением к уже внедрённой стационарной системе планирования, но может использоваться и обособленно.

1.2 Характеристики объекта автоматизации

Муниципальное казённое образовательное учреждение Таежнинская школа № 20 создано с целью оказания муниципальных услуг, выполнения работ и исполнения муниципальных функций в целях обеспечения реализации предусмотренных законодательством Российской Федерации полномочий органов местного самоуправления в сфере образования.

Школа является некоммерческой организацией, созданной в соответствии с Гражданским кодексом Российской Федерации.

Учредителем и собственником имущества школы является Муниципальное образование Богучанский район.

					ДП – 230105.65 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		7

Школа создает условия для реализации гражданами Российской Федерации гарантированного государством права на получение общедоступного и бесплатного общего образования (начального общего, основного общего и среднего (полного) общего), если образование данного уровня гражданин получает впервые.

В своей деятельности учреждение руководствуется Конституцией Российской Федерации, Федеральными законами, указами и распоряжениями Президента Российской Федерации, постановлениями и распоряжениями Правительства Российской Федерации, решениями органов управления образованием, Типовым положением об общеобразовательном учреждении, иными правовыми актами Российской Федерации, Красноярского края, муниципальными правовыми актами, настоящим Уставом, договором с Учредителем.

Основными целями учреждения являются:

- формирование общей культуры личности обучающихся на основе усвоения обязательного минимума содержания общеобразовательных программ;
- адаптация обучающихся к жизни в обществе;
- создание основы для осознанного выбора и последующего освоения профессиональных образовательных программ;
- воспитание гражданственности, трудолюбия, уважения к правам и свободам человека, любви к окружающей природе, Родине, семье;
- формирование здорового образа жизни.

1.3 Назначение системы

Разрабатываемый программный продукт предназначен для:

- Создания задач и подзадач неограниченной вложенности
- Обеспечения оперативного планирования;
- Обеспечения оперативного доступа к задачам;
- Напоминания о задачах;
- Контроля сроков выполнения задач;
- Обмена данными с существующими настольными системами;

					ДП – 230105.65 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		8

1.4 Исполнители производственных и организационных работ

Субъектами производственного процесса и организационных работ являются:

- Администрация учреждения (директор, заместители);
- Руководители предметных методических объединений
- Классные руководители;
- Учителя-предметники;
- Другие педагогические работники;

Организационная структура учреждения представлена в приложении 1.

1.5 Выбор средств описания бизнес-процессов разработки

Технология создания информационных систем предъявляет высокие требования к методикам реализации и программным инструментальным средствам. Такие средства должны поддерживать все стадии реализации проектов и обеспечивать координацию усилий больших коллективов разработчиков.

Средства описания бизнес-процессов отличаются по функциональным возможностям, и выбрать нужное средство для поддержки проекта по оптимизации бизнес-процессов сложно. На сегодняшний день получили распространение следующие системы описания бизнес-процессов: Visio, ARIS-Toolset, Rational Rose и BPWin [1].

Ниже представлены основные характеристики этих программных средств.

Visio – наиболее простое и доступное средство моделирования процессов. Этот продукт имеет стандартные, привычные всем панели управления в стиле MS Office и легко интегрируется с любыми приложениями этого пакета, что упрощает работу с ним для неопытных пользователей. Однако для временного или стоимостного анализа требуется разработка отчетов, что значительно усложняет использование этого продукта. Типовые отчеты явно не достаточны для анализа бизнес-процессов. Несмотря на это, *Visio* является распространенным средством для описания бизнес-процессов как в России, так и за рубежом. *Visio* поддержи-

					ДП – 230105.65 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

вает IDEF и UML форматы для описания бизнес-процессов. Возможна также самостоятельная разработка форматов.

ARIS рассматривает предприятие как совокупность четырех взглядов (views):

- взгляд на организационную структуру;
- взгляд на структуру функций;
- взгляд на структуру данных;
- взгляд на структуру процессов.

ARIS позволяет составлять диаграмму целей, связывая процессы через цели с миссией компании. В результате после построения бизнес-модели получается комплексное видение компании: Цели – Процессы – Оргструктура – Данные – Продукты/услуги в виде отдельных, но связанных через объекты диаграмм. Это означает, что при изменении названия должности на одной диаграмме сразу корректируются названия во всех процессах, где она присутствует, и в оргструктуре.

При этом каждый из данных взглядов разделяется еще на три подуровня:

- описание требований;
- описание спецификации;
- описание внедрения.

ARIS предлагает рассматривать организацию с позиции 4-х аспектов, отображающих разные взгляды на предприятие, а также разную глубину этих взглядов. Для описания бизнес-среды предлагается использовать 85 типов моделей (обычно в практической деятельности применяется не более 6-7 типов моделей), каждая из которых принадлежит тому или иному аспекту. ARIS Toolset является, с одной стороны, достаточно сложной для освоения системой. С другой стороны, диаграммы бизнес-процессов в готовом виде понятны даже неподготовленным сотрудникам, это позволяет эффективно организовывать работу команд, не прибегая к тотальному обучению всех работающих над проектом сотрудников.

Rational Rose – CASE-средство фирмы Rational Software Corporation (США), предназначено для автоматизации этапов анализа и проектирования ПО, а также

					ДП – 230105.65 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

для генерации кодов на различных языках и выпуска проектной документации. Rational Rose использует синтез-методологию объектно-ориентированного анализа и проектирования, основанную на подходах трех ведущих специалистов в данной области: Буча, Рамбо и Джекобсона. Разработанный ими универсальный язык для моделирования объектов (UML - Unified Modeling Language) претендует на роль стандарта в области объектно-ориентированного анализа и проектирования. Конкретный вариант Rational Rose определяется языком, на котором генерируются коды программ (C++, Delphi, Smalltalk, PowerBuilder, Ada, SQLWindows и ObjectPro). Основной вариант – Rational Rose/C++ – позволяет разрабатывать проектную документацию в виде диаграмм и спецификаций, а также генерировать программные коды на C++. Кроме того, Rational Rose содержит средства реинжиниринга программ, обеспечивающие повторное использование программных компонент в новых проектах.

BPWin занимает промежуточное место, отличаясь достаточной простотой и большими возможностями анализа. Функциональность BPWin заключается не только в создании диаграмм, но и в проверке целостности и согласованности модели. BPWin обеспечивает логическую четкость в определении и описании элементов диаграмм, а также проверку целостности связей между диаграммами. Инструмент обеспечивает коррекцию наиболее часто встречающихся ошибок при моделировании. Кроме того, BPWin поддерживает пользовательские свойства, которые применяются к элементам диаграммы для описания специфических свойств, присущих данному элементу. Основным ограничением этой системы является положенный в ее основу стандарт IDEF, в котором существуют жесткие ограничения при построении моделей. Это упрощает задачу при описании простых процедур, но усложняет описание больших процессов. Схемы IDEF при описании сложных процессов начинают представлять бесчисленное множество взаимосвязанных схем, внешне очень похожих, что затрудняет понимание процесса в целом. Часто не удается представить нужную степень точности описания на одной схеме [2].

При написании данного проекта мною были использованы CASE-средства компании Computer Associates (CA). CA предлагает полный набор программных средств, обеспечивающих эффективное управление, анализ, проектирование и кодогенерацию информационных систем.

Для проведения анализа и реорганизации бизнес-процессов CA предлагает CASE - средство верхнего уровня - BPwin, поддерживающий методологии IDEF0 (функциональная модель), IDEF3 (WorkFlow Diagram) и DFD (DataFlow Diagram).

1.6 Анализ рынка мобильных приложений для планирования деятельности

Рынок мобильных приложений достаточно богат, рассмотрим некоторые мобильные приложения для планирования задач из таких магазинов приложений как Google Play, App Store и Windows Store.

Google Play - магазин приложений от компании Google, позволяющий владельцам устройств с операционной системой Android приобретать и устанавливать различные приложения. В Google Play можно найти много полезных и разнообразных приложения. В магазине присутствуют как платные, так и бесплатные приложения. 31 октября 2012 года компания Google объявила, что количество приложений достигло 700 тыс. приложений, и количество скачиваний достигло 25 млрд [3].

App Store — магазин приложений, раздел онлайн-магазина iTunes Store, содержащий различные приложения для мобильных телефонов iPhone, плееров iPod Touch и планшетов iPad, а также для персональных компьютеров Mac и позволяющий их купить, либо скачать бесплатно. App Store предлагает более 1,5 млн. приложений для iPhone и iPod Touch и около 725 тыс. для iPad (на 10 июня 2015), число загрузок превысило 100 миллиардов, а пользовательская база составляет порядка 575 миллионов человек. Стоимость большинства продающихся приложений составляет от \$0,99 до \$9,99, некоторые профессиональные приложения стоят существенно больше [4].

Windows Phone Store — магазин приложений Microsoft для собственной мобильной платформы Windows Phone, которая позволяет пользователям устанавливать и приобретать различные приложения и игры. Интернет-магазин доступен как с телефона под управлением Windows Phone, так и с помощью браузера. В конце июня 2012 года Microsoft было официально заявлено о более 100 тысячах доступных приложений в магазине [5].

В процессе сбора данных для решения задачи автоматизации планирования деятельности был проанализирован рынок программных продуктов, способных решить поставленную задачу. В качестве критериев сравнения было выбрано следующее:

- наличие бесплатной версии
- наличие уведомлений
- возможность поиска по задачам
- открытый интерфейс для обмена данными с настольными системами

Any.do в Google Play— простой планировщик задач с интерфейсом на русском языке. Any.do имеет минималистичный дизайн. В этом приложении задачи создаются как в отдельности, так и в проектах. Дела можно планировать на несколько дней вперед. В Any.do есть несколько интересных функций. К примеру, выполненные задачи можно удалить путем встряхивания смартфона.

Возможности:

- добавление задач и подзадач
- составление списков
- напоминания
- возможность поделиться с другими пользователями Any.do своими задачами и списками
- существует сильно ограниченная бесплатная версия

ToDoist в Google Play - проект, который появился в 2007 году и с тех пор постоянно развивается. К 2013-му году число пользователей ToDoist достигло заветной цифры в 1 млн. пользователей, а на сегодняшний день по некоторым дан-

ным их уже более 4 млн. ToDoist отличает легкий и простой интерфейс и в то же время хорошая функциональность.

Возможности:

- добавление задач и проектов (в бесплатной версии существует ограничение на количество создаваемых проектов)
- возможность дать доступ к проекту кому угодно – коллегам, друзьям, членам семьи – для совместной работы над общими планами и целями
- разбивка задач на подзадачи
- уведомления на электронную почту или push-уведомления
- поиск по задачам (только в платной версии).

Wunderlist в Google Play, App Store и Windows Store – разработка немецкой команды программистов 6 Wunderkinder GmbH. Wunderlist во многом похож на ToDoist. Практически тот же функционал, легкий и понятный интерфейс, возможность командной работы, но при этом практически все функции в Wunderlist бесплатны как для Android и iOS, так и для мобильных устройств с операционной системой Windows.

Возможности:

- добавление задач и подзадач (ограничение количества подзадач в бесплатной версии - 25)
- составление списков
- установка крайнего срока выполнения задачи
- уведомления на электронную почту или push-уведомления
- поиск по задачам

OmniFocus в App Store – популярный на iOS менеджер задач, разработанный компанией The Omni Group. Стоимость приложения для iPhone составляет 19.99\$. Приложение славится тем, что в нем можно создавать неограниченное количество проектов и делать любые иерархии. Существует настольная версия для MacOS, стоимость стандартной и Pro версии программы составляет 39.99\$ и 79.99\$ соответственно.

					ДП – 230105.65 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

Возможности:

- добавление задач и подзадач
- составление списков
- установка крайнего срока выполнения задачи
- push-уведомления
- поиск по задачам

ToDo в App Store – отличное средство для тайм менеджмента с удобным интерфейсом и набором интересных функций. Это одно из самых первых в мире приложений с функцией списка текущих дел и задач. Среди преимуществ утилиты – красочный современный интерфейс, гибкость и простота использования и планирование методом перетаскивания. В приложении предусмотрены проекты и контрольные списки (с подзадачами), полнофункциональный поиск, повторяющиеся задачи. Стоимость мобильного приложения для iOS составляет 4.99\$. Существует так же и настольная версия программы для MacOS, её стоимость составляет 14.99\$.

Возможности:

- добавление задач и подзадач
- составление списков
- повторяющиеся задачи
- push-уведомления
- поиск по задачам

Perfect Note в Windows Store - Один из лучших органайзеров на Windows Phone. Приложение обладает большим набором возможностей, среди которых создание подтем с приоритетами, добавление контактов, регулировка прогресса, резервное копирование данных и облачная синхронизация, установка напоминаний и прочее. Программа бесплатна.

Возможности:

- добавление задач и подзадач
- push-уведомления

- поиск по задачам

Все вышеописанные программные продукты, несомненно, являются качественными и выполняют возложенные на них функции, но в наше учреждение уже внедрена настольная система для планирования задач и в мобильном приложении необходимо открытое API для обмена данными. Отсутствие открытого API для обмена с настольными системами не позволяет использовать ни один из описанных выше программных продуктов в нашем учреждении.

Таблица 1.1. Сравнительный анализ существующих аналогов.

Наименование	Наличие бесплатной версии	Наличие уведомлений	Поиск по задачам	Открытый API для обмена
Any.do	+	+	-	-
ToDoist	+	+	+	-
Wunderlist	+	+	+	-
OmniFocus	-	+	+	-
ToDo	-	+	+	-
Perfect Note	+	+	+	-

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

От того, насколько правильно будет спроектирована архитектура программного комплекса и отдельных его компонентов, зависят очень многие качества проекта. В тесной зависимости от применённой архитектуры находятся способ реализации программного кода и время работы над проектом. В немалой степени от этого будет зависеть применённая модель жизненного цикла и методология работы над проектом. От выбранной архитектуры будет зависеть и степень лёгкости интеграции различных модулей в построенную систему.

2.1 Выбор модели жизненного цикла

Одним из базовых понятий при проектировании любой информационной системы является модель жизненного цикла (ЖЦ). Жизненный цикл – это непрерывный процесс, который начинается с момента принятия решения о необходимости его создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации [6].

В настоящее время остаются распространёнными две модели жизненного цикла программного обеспечения, которые многие специалисты считают наиболее перспективными для малых групп разработчиков: спиральная и эволюционная. О таких моделях как RUP или XP в данном проекте речи быть не может, поскольку разработчик всего один. Но так же будет рассматриваться одна из современных гибких методологий - Scrum.

При использовании спиральной модели ЖЦ для проектирования программного продукта основной упор делается на начальные этапы: анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии комплекса, на нем уточняются цели и характеристики проекта, оценивается его качество, и планируются работы следующего витка спирали.

В случае спиральной модели процесса разработки программного обеспечения последовательность: анализ требований – проектирование – реализация – тес-

					ДП – 230105.65 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

тирование – выполняется более одного раза. Для этого может быть несколько причин. Основная причина обычно связана с необходимостью предупреждения рисков. Другой причиной может быть необходимость предоставить заказчику частичную версию проекта для получения отзывов и пожеланий. Если разрабатываемая программа достаточно сложна, необходимо выполнять промежуточные интеграции, не откладывая эту фазу на самый конец, как это предписывает, например, каскадная модель. Общая же идея спирального процесса заключается в том, чтобы на каждой итерации строить очередную версию программы, используя в качестве основы ее предыдущую версию.

Разработка итерациями отражает объективно существующий спиральный цикл создания системы. При спиральном цикле возможно неполное завершение работ на каждом этапе с переходом на следующий этап. При итеративном способе разработки недостающую работу можно будет выполнить на следующей итерации. При этом достигается главная задача – как можно быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований, что в конечном итоге способствует получению продукта максимально соответствующего требованиям заказчика.

Эволюционная модель жизненного цикла является своего рода развитием спиральной. Основное ее отличие заключается в том, что в начале работ определяются не все требования. Работа ведется итерациями, каждая из которых состоит из четырех этапов. На этапе планирования осуществляется сбор или дополнение требований и дальнейшее планирование проекта. Затем, на основе результатов первого этапа, анализируются риски, и принимается решение о целесообразности дальнейшей работы над проектом. Далее идет конструирование, по сути – реализация запланированных изменений в программе. Наконец, новая версия продукта демонстрируется заказчику, который проводит оценку, и начинается новая итерация с планирования.

Методология Scrum представляет собой итеративный процесс разработки программного обеспечения. При такой разработке для программного продукта

создается много последовательных выпусков, в которых постепенно добавляется требуемая функциональность. Итеративный подход позволяет по завершению текущей итерации продемонстрировать заказчику работоспособный программный продукт, возможно с ограниченной функциональностью, получить отзыв, замечания и дополнительные требования, которые будут учтены в следующих итерациях. Основными артефактами в методологии Scrum являются рабочие элементы, отчеты, книги и панели мониторинга.

Цикл выпуска продукта в Scrum состоит из ряда итераций, которые называются спринтами. Спринт имеет фиксированную длительность, как правило, 1-4 недели. Элементы работы, включенные в очередной спринт, не подлежат изменению до его окончания. Хотя спринт завершается подготовкой работоспособного программного продукта, его текущей функциональности может быть недостаточно для оформления выпуска, имеющего ценность для заказчика. Поэтому выпуск работоспособного программного продукта, который заказчик может использовать, включает, как правило, несколько спринтов.

При разработке данного программного обеспечения в качестве модели жизненного цикла была выбрана методология Scrum. В дальнейшем данное решение показало свою эффективность. После первоначального проектирования и выпуска первой альфа версии была сформирована система работы. Спринты длились 2-3 недели. В этот момент тестировалась версия предыдущего спринта, разрабатывался функционал текущего спринта и планировались задачи на последующие спринты. В результате данного подхода заказчики в течение первого месяца разработки получили программное обеспечение, с которым уже можно было работать. В дальнейшем его функциональность наращивалась с каждым спринтом.

2.2 Выбор платформы

Так как на рынке присутствуют устройства на различных платформах, необходимо выбрать для какой из них будет вестись разработка. На сегодняшний

					ДП – 230105.65 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		19

день лидирующие позиции занимают три операционные системы: Android, iOS и Windows Phone. Немного рассмотрим каждую из них:

- Android - портативная (сетевая) операционная система для коммуникаторов, планшетных компьютеров, электронных книг, цифровых проигрывателей, наручных часов и нетбуков основанная на ядре Linux. Изначально разрабатывалась компанией Android Inc., которую затем купила Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (ОНА), который сейчас занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать библиотеки и компоненты приложений, написанные на Си и других языках;
- iOS (до 24 июня 2010 года - iPhone OS) - мобильная операционная система, разрабатываемая и выпускаемая американской компанией Apple. В настоящее время Android развивается экспоненциально: каждый год число пользователей данной ОС неуклонно растет.
- Windows Phone — мобильная операционная система, разработанная Microsoft, вышла 11 октября 2010 года. 21 октября начались поставки первых устройств на базе новой платформы. Операционная система является преемником Windows Mobile, хотя и несовместима с ней.

Согласно данным аналитической компании IDC на мировом рынке доли вышеописанных ОС распределились следующим образом:

- Android – 78%
- iOS – 18,3%
- Windows Phone – 2,7%

Устройства на других платформах, в число которых входят Symbian, Bada, Blackberry и другие приходится порядка 1%, поэтому их поддержка целесообраз-

					ДП – 230105.65 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

на только в исключительных случаях. В России показатели немного отличаются, в частности, несколько выше доля Windows Phone (около 7%), в основном за счет уменьшения доли Android, но общая картина не сильно отличается от мирового рынка.

Также необходимо учитывать специфику устройств. Android – в основной массе недорогие устройства со средними показателями производительности. Устройства сильно различаются по размерам, характеристиками, версиям Android. Google осуществляет меньший контроль над приложениями в Play Market, что предоставляет большую свободу разработчикам. Устройства на базе iOS – значительно более высокая ценовая категория, даже по сравнению с флагманами на Android, все устройства выпускаются самой компанией Apple и вследствие этого на рынке нет такого многообразия различных параметров устройств. Приложения в App Store проходят достаточно строгую модерацию перед публикацией. Windows Phone – также в массе своей достаточно доступны, разнообразие моделей гораздо меньше, чем для Android, приложений под данную платформу – тоже.

Решающим фактором в выборе платформы - Android стал опрос «Какая ОС на вашем мобильном?», проведенный в учреждении. Были получены следующие результаты:

- Android используют 84,5% работников
- iOS – 3%
- Другие платформы – 12,5%

2.3 Выбор средств разработки

После того, как была выбрана модель жизненного цикла программного продукта и платформа, встает вопрос выбора средств программной реализации. Это немаловажно, поскольку от этого будет зависеть надёжность созданного программного средства, а так же простота дальнейшего сопровождения и модернизации.

					ДП – 230105.65 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		21

Разработка приложений для платформы Android связана с группой инструментов, которые предоставляются набором Android SDK или NDK. Также понадобятся интегрированная среда разработки.

В качестве среды разработки будем использовать IDE, признанную официальным средством разработки Android приложений, Android Studio, основанную на программном обеспечении IntelliJ IDEA от компании JetBrains. Данная среда разработки доступна для Windows, OS X и Linux. Android Studio работает как с SDK, так и с NDK.

SDK — software development kit, комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

NDK — native development kit пакет инструментариев и библиотек, позволяющий реализовать часть приложения на языке C/C++. NDK рекомендуется использовать для разработки участков кода, критичных к скорости.

К сожалению, Android Studio не поддерживает сборку C/C++ кода так же хорошо, как Java кода. Поддержка NDK в ней на данный момент очень ограничена. Так, единственно поддерживаемыми NDK приложениями являются только те, которые состоят из одного собираемого модуля (финальной динамической библиотеки), держат все исходные коды на C/C++ в каталоге 'jni', в которых также отсутствуют любые зависимости от других библиотек, и которые нельзя разбить на несколько модулей (т.е. набор статических и динамических библиотек). Не предоставляется никаких возможностей для настройки сборки нативных модулей, за исключением очень ограниченного набора опций. Google рекомендует прибегать к использованию NDK только в редчайших случаях.

Для работы с SDK необходим набор Java Development Kit (JDK) - бесплатно распространяемый компанией Oracle Corporation (ранее Sun Microsystems) комплект разработчика приложений на языке Java, включающий в себя компилятор

Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE).

Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle).

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание [7].

Часто к недостаткам концепции виртуальной машины относят снижение производительности. Ряд усовершенствований несколько увеличил скорость выполнения программ на Java:

- применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде,
- широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,
- аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами фирмы ARM).

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik(начиная с Android 5.0 Lollipop виртуальная машина заменена на ART). Для такой компиляции используется дополнительный инструмент, а именно Software Development Kit, разработанный компанией Google [8].

2.4 Архитектура программного продукта

Приложения для Android в своей работе использует окна (аналогично Windows), однако в данной системе вышеуказанные окна носят иное название – Activity. Каждая activity представлена в проекте классом, реализованном на языке Java, хранящемся в одноименном файле с расширением .java. Каждой activity соответствует xml файл-описание. В xml-файле описано в виде xml-кода расположение визуализируемых объектов. При запуске activity система Android автоматически распознает размер экрана мобильного устройства и приводит выводимый контент в соответствие с разметкой, описанной в xml-файле. Таким образом, одна и та же activity будет выглядеть одинаково независимо от диагонали используемого устройства. Также, для каждого приложения Android должен существовать xml-файл, в котором в виде xml-кода будут прописаны минимальные требования к системе, а также activity, вызываемая при запуске приложения. Как и в Windows, каждое окно имеет свой жизненный цикл и свои особенности. При создании нового окна вызывается метод onCreate(), при разработке данный метод переопределяется и в нем происходит инициализация приложения и его компонентов. Далее вызываются методы onStart() и onResume(). Оба метода вызываются перед отображением окна при его создании, либо восстановлении (при переключении из другого приложения, при разворачивании свернутого приложения и т.п.). При сворачивании вызываются методы onPause() и onStop(). При закрытии приложения и окна вызывается onDestroy(), в данном методе можно сохранить пользовательские данные и параметры [9].

Функционально, приложение состоит из приведенных ниже модулей (activity).

- Основная activity состоит из списка задач (представлен элементом ListView), элемента выбора даты и выдвигаемого меню, позволяющего выбрать фильтры по статусу, открыть настройки программы, а так же провести синхронизацию с настольной системой. При запуске данной activity происходит обращение к базе данных для загрузки списка задач.
- Activity просмотра задачи состоит из набора элементов TextView и EditText, при запуске activity происходит обращение к базе данных посредством класса SqlDataHelper, далее происходит загрузка данных задачи по её идентификатору.
- Activity создания новой задачи состоит из набора элементов EditText, TextView и кнопки сохранения, при нажатии на которую происходит запись задачи в базу данных.
- Activity настроек программы позволяет нам указать настройки сервера синхронизации, а так же установить время, за которое следует уведомлять о наступлении времени выполнения задачи.

Приложение работает со встраиваемой реляционной базой данных SQLite. SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле [10].

SQLite поддерживает типы TEXT (аналог String в Java), INTEGER (аналог long в Java) и REAL (аналог double в Java). Все остальные типы следует конвертировать, прежде чем сохранять в базе данных.

Библиотека Android содержит абстрактный класс SQLiteOpenHelper, с помощью которого можно создавать, открывать и обновлять базы данных. Это основной класс, с которым осуществляется работа в проекте. При реализации этого вспомогательного класса от вас скрывается логика, на основе которой принимается решение о создании или обновлении базы данных перед ее открытием.

Класс SQLiteOpenHelper содержит два абстрактных метода: onCreate() - метод, который вызывается при первом создании базы данных, а так же onUpgrade(), который вызывается при модификации базы данных.

В приложении создан класс DatabaseHelper, наследуемый от SQLiteOpenHelper. В этом классе реализованы методы onCreate() и onUpgrade(). В них описана логика создания и модификации базы данных.

В результате выполнения запросов к базе данных возвращается объект Cursor, содержащий таблицу с результатами запроса. Cursor предполагает последовательную работу со строками результата. В каждый момент времени активна одна строка, на которую ссылается указатель. Перебирая записи последовательно, можно получить доступ к данным.

Взаимодействие с настольной системой происходит с помощью отправки SOAP запросов на сервер приложений.

SOAP (Simple Object Access Protocol) это основанный на XML протокол, предназначенный для обмена данными между распределенными приложениями. SOAP строится поверх существующих web протоколов, например HTTP.

SOAP запрос/ответ передается как пакет SOAP Envelope, состоящий из заголовка SOAP Header и тела SOAP Body (рис. 2.1). SOAP Header - вспомогательный компонент, содержащий конфиденциальную информацию для приложения, например параметры аутентификации. SOAP Body - это сообщение с полезной информацией. Заголовок может также содержать SOAP Action, который определяет функцию, запрашиваемую сервисом.

					ДП – 230105.65 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

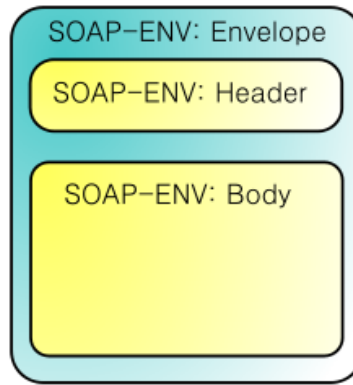


Рисунок 2.1 Структура SOAP запроса

Общая схема взаимодействия мобильного клиента с настольной системой показана на рисунке 2.2.

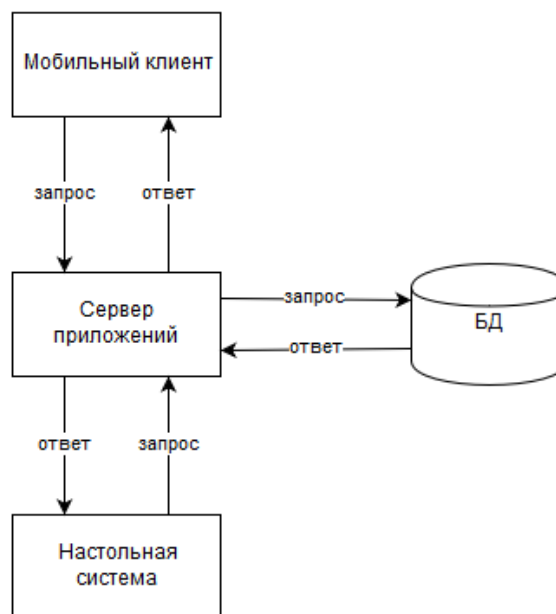


Рисунок 2.2 Диаграмма взаимодействия с настольной системой

2.5 Проектирование модели IDEF0

Моделируя деятельность предприятия, определяем как входную, так и выходную информацию, а также учитываем внешние факторы, влияющие на деятельность предприятия – это устав учреждения, законодательство, техническое обеспечение и др.

Анализируя функции, выполняемые учреждением, выделим процесс инициации, назначение и решения какого-либо поручения (рис. 2.3).

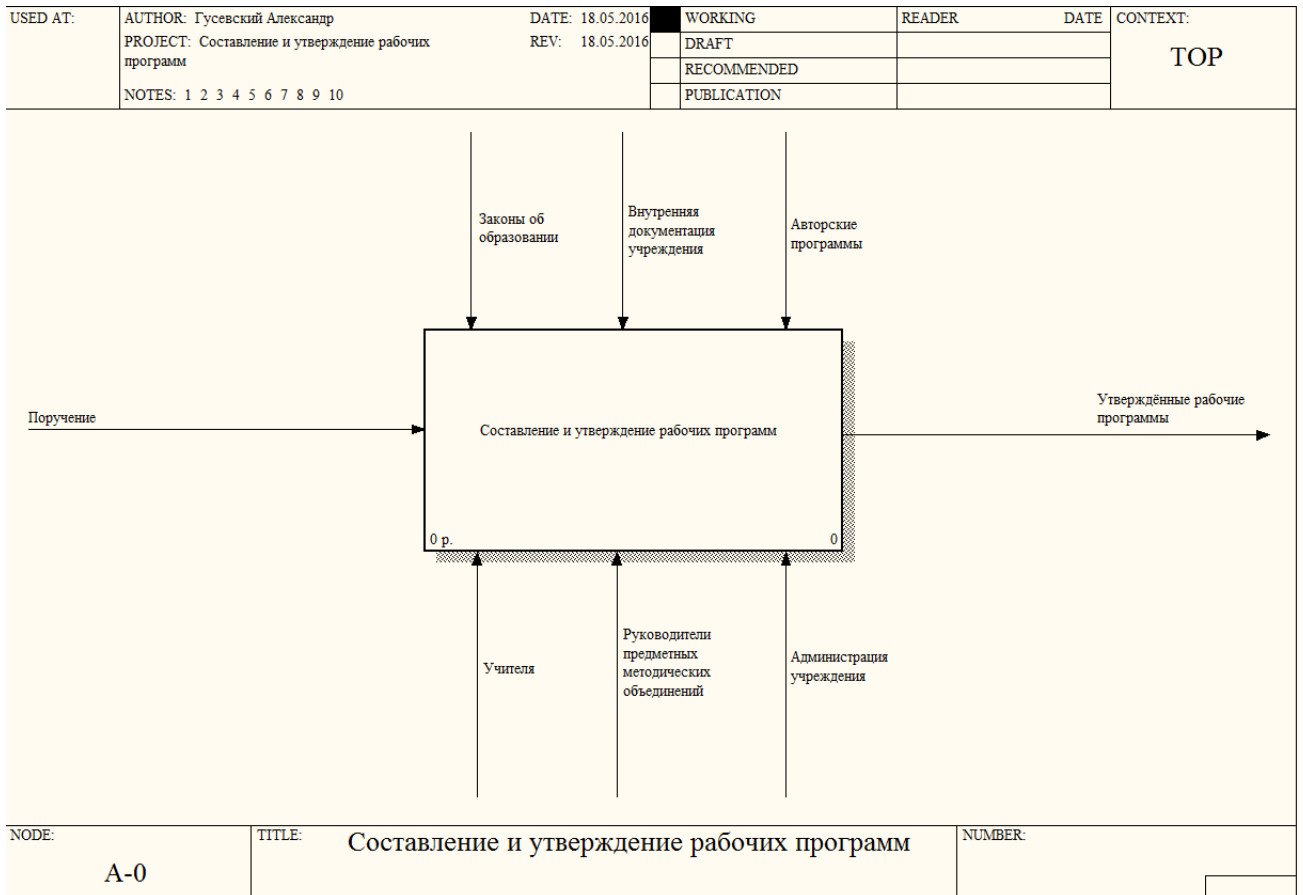


Рисунок 2.3 Контекстная диаграмма

На рисунке 2.4. представлена декомпозиция процесса составления и утверждения рабочих программ, она состоит из следующих процессов:

- Составление рабочих программ;
- Проверка рабочих программ;
- Утверждение рабочих программ.

USED AT:	AUTHOR: Гусевский Александр	DATE: 18.05.2016	WORKING	READER	DATE	CONTEXT:
	PROJECT: Составление и утверждение рабочих программ	REV: 18.05.2016	DRAFT			
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMMENDED			
			PUBLICATION			A-0

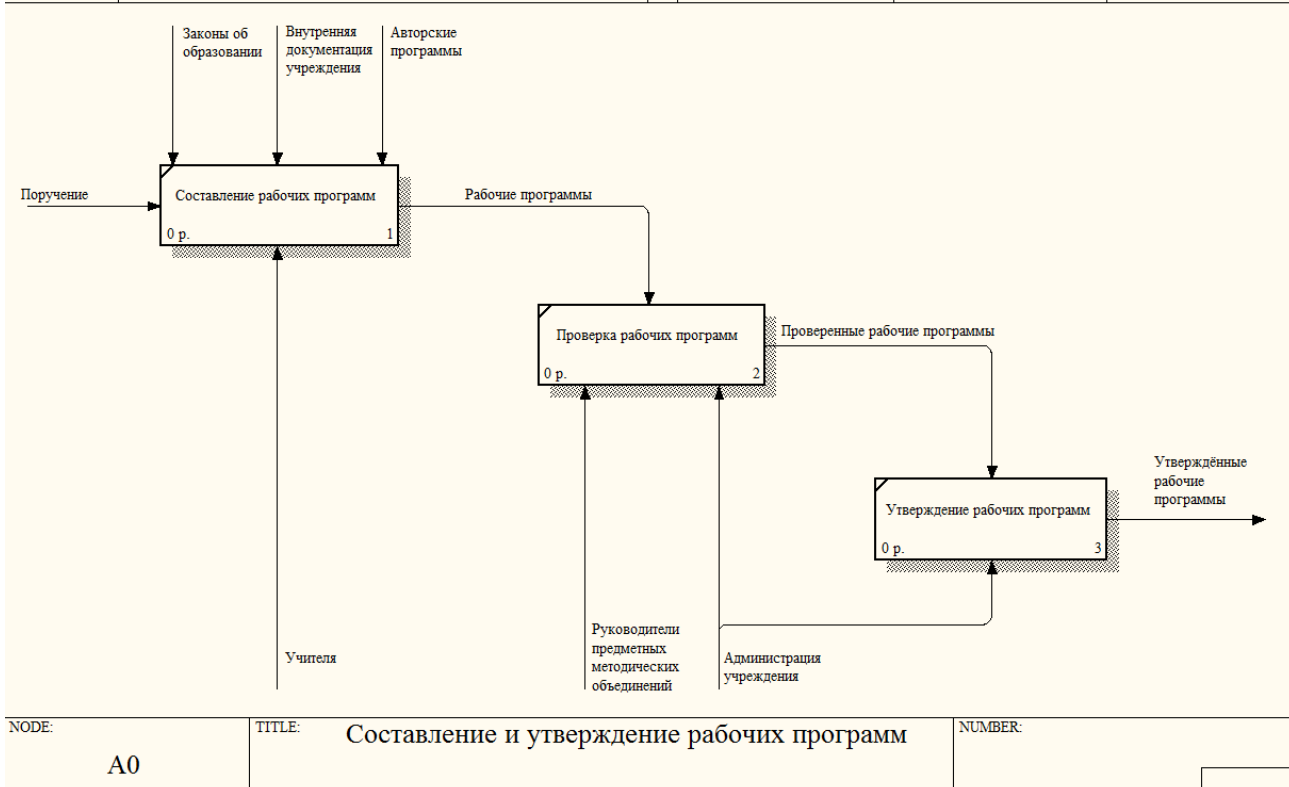


Рисунок 2.4. Диаграмма декомпозиции А0.

На рисунке 2.5 представлена диаграмма вариантов использования программного продукта.

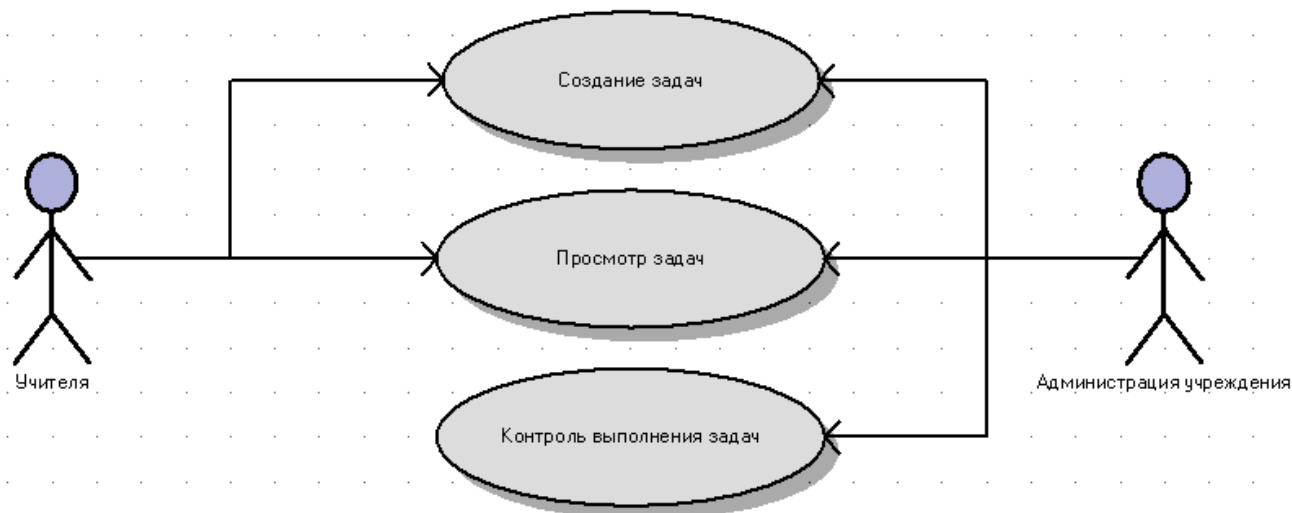


Рисунок 2.5. Варианты использования.

Изм.	Лист	№ докум.	Подпись	Дата

ДП – 230105.65 ПЗ

3 ОПИСАНИЕ ПРИЛОЖЕНИЯ

3.1 Создание задач и подзадач

Приложение имеет функцию добавления задач, а так же подзадач с неограниченной глубиной вложенности. Для каждой задачи можно указать имя, описание, дату выполнения и время (рис. 3.1). Если дата и время не будут указаны, то программа не будет напоминать о задаче. После заполнения данных необходимо нажать кнопку «Сохранить». По умолчанию новой задаче будет присвоен статус «Новая».

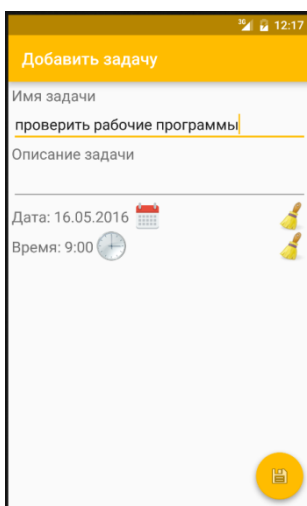


Рисунок 3.1 Форма создания задачи

Для создания подзадачи, необходимо открыть родительскую задачу и выбрать пункт «Создать подзадачу» в меню дополнительных действий, там же можно и удалить задачу (рис. 3.2).

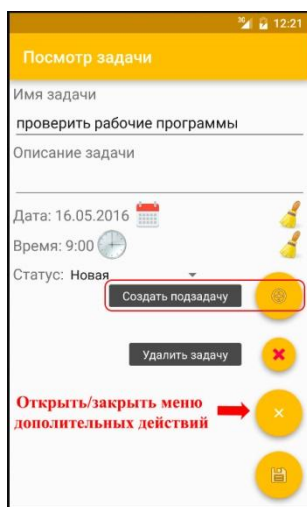


Рисунок 3.2 Создание подзадачи

Изм.	Лист	№ докум.	Подпись	Дата

3.2 Контекстное меню задачи

При длинном нажатии на задачу в списке будет вызвано контекстное меню задачи. В данном меню можно осуществить просмотр задачи, смену статуса задачи, просмотр всех подзадач выбранной задачи, а так же удалить задачу (рис. 3.3).

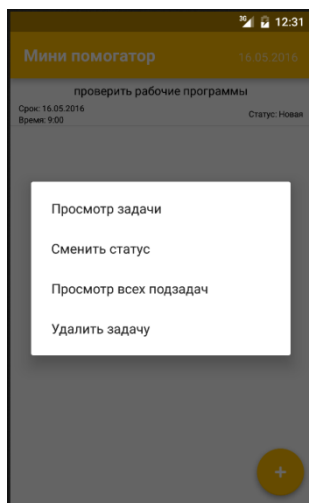


Рисунок 3.3 Контекстное меню задачи

При выборе пункта «Сменить статус» будет предложено выбрать статус задачи (рис. 3.4).

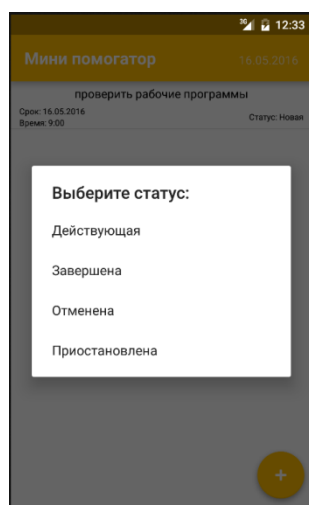


Рисунок 3.4 Смена статус задачи

Если же выбрать пункт «Удалить задачу», то в случае если задача не имеет подзадач, она будет удалена, иначе будет выведено диалоговое окно с предоставлением выбора: удалить все подзадачи или отменить удаление (рис. 3.5).

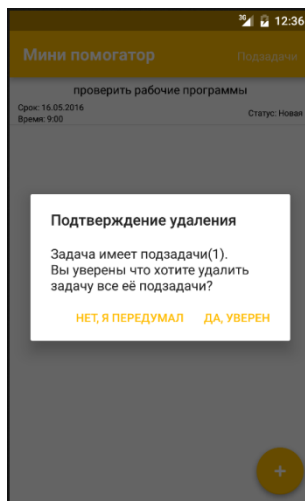


Рисунок 3.5 Удаление задачи, которая имеет подзадачи

3.3 Поиск задачи

Для оперативного доступа к задачам существует функция поиска задач по имени. Для выполнения поиска необходимо выдвинуть меню слева (рис. 3.6) и выбрать пункт меню «Поиск». Откроется форма поиска (рис. 3.7). При клике на задачу в окне поиска откроется форма просмотра задачи. Исполняемый код модуля поиска находится в приложении 2.

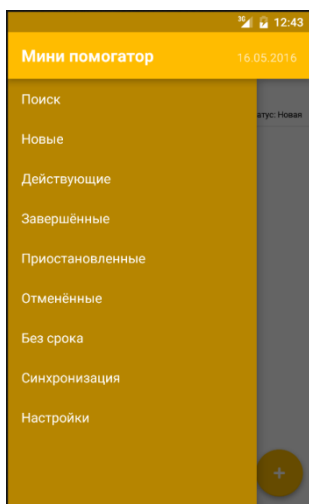


Рисунок 3.6 Выдвижное меню

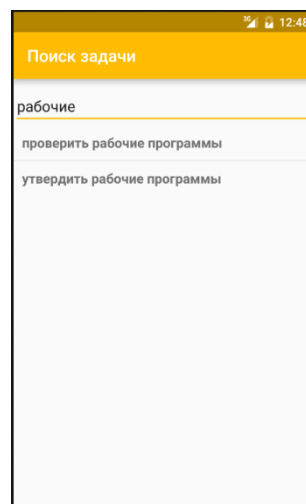


Рисунок 3.7 Пример поиска задач

3.4 Фильтры

Для удобной навигации по задачам реализованы фильтры по статусу и дате. Чтобы вывести задачи по дате, необходимо нажать на дату в Action Bar (рис. 3.8) и выбрать дату (рис. 3.9), по умолчанию выбрана текущая дата.

Изм.	Лист	№ докум.	Подпись	Дата



Рисунок 3.8 Фильтр по дате

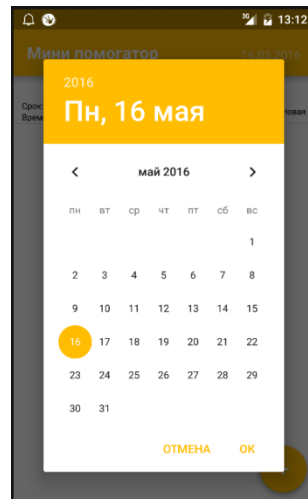


Рисунок 3.9 Диалог выбора даты

Если необходимо вывести задачи без срока, нужно выдвинуть меню слева и выбрать пункт меню «Без срока». Аналогично можно отфильтровать задачи по статусу (рис. 3.10).

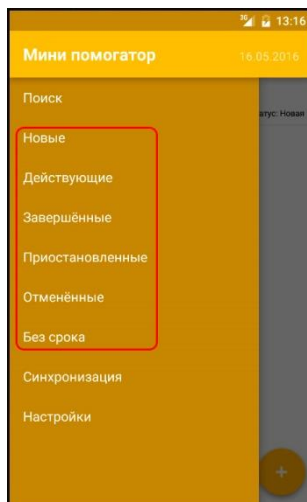


Рисунок 3.10 Фильтр «без срока» и фильтры по статусу

3.5 Напоминания

В данном приложении реализована служба, позволяющая напоминать о задачах за определённое время до начала выполнения задачи. По умолчанию приложение напоминает за 10 минут до времени выполнения задачи, но в настройках программы можно сменить время, за которое следует напоминать о задаче, а так же можно совсем отключить напоминания (рис. 3.11). Исполняемый код службы напоминаний находится в приложении 2.

Изм.	Лист	№ докум.	Подпись	Дата

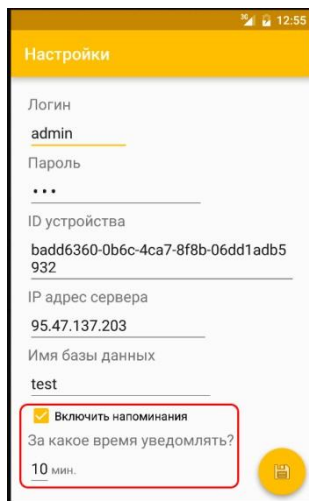


Рисунок 3.11 Форма настроек

Напоминание производится путём создания сообщения в области уведомлений устройства (рис. 3.12), звукового сигнала, а так же светодиодной индикации, если для этого имеется техническая возможность.

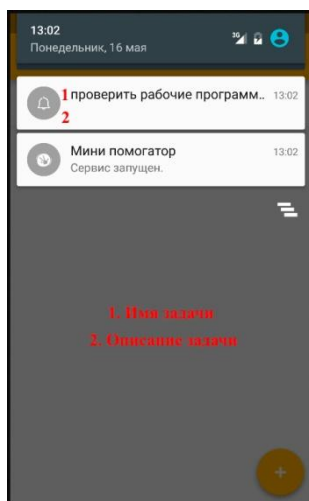


Рисунок 3.12 Напоминание о задаче

3.6 Синхронизация

В мобильном приложении реализована функция обмена данными с настольной системой. В текущей версии обмен происходит вручную (по нажатию соответствующего пункта выдвигного меню), в будущих версиях будет реализована фоновая синхронизация. В настоящий момент реализован обмен с настольной системой реализованной с использованием платформы «1С Предприятие».

Изм.	Лист	№ докум.	Подпись	Дата

ДП – 230105.65 ПЗ

Лист

34

Обмен происходит через стандартные механизмы веб. В данном случае разрабатываемая система является потребителем веб-сервисов.

Web-сервисы - это один из механизмов платформы, используемых для интеграции с другими информационными системами. Он является средством поддержки SOA (Service-Oriented Architecture) - сервис-ориентированной архитектуры, которая является современным стандартом интеграции приложений и информационных систем (рис. 3.13).

Значительным преимуществом сервис - ориентированной архитектуры является то, что она позволяет развивать инфраструктуру предприятия однородным образом, без разрушения уже существующих решений. Ее использование позволяет минимизировать издержки за счет интеграции разнородных и унаследованных систем в современный ландшафт предприятия. Она позволяет реализовывать слабо связанные программные компоненты с тем, чтобы максимально повысить их повторную используемость [11].

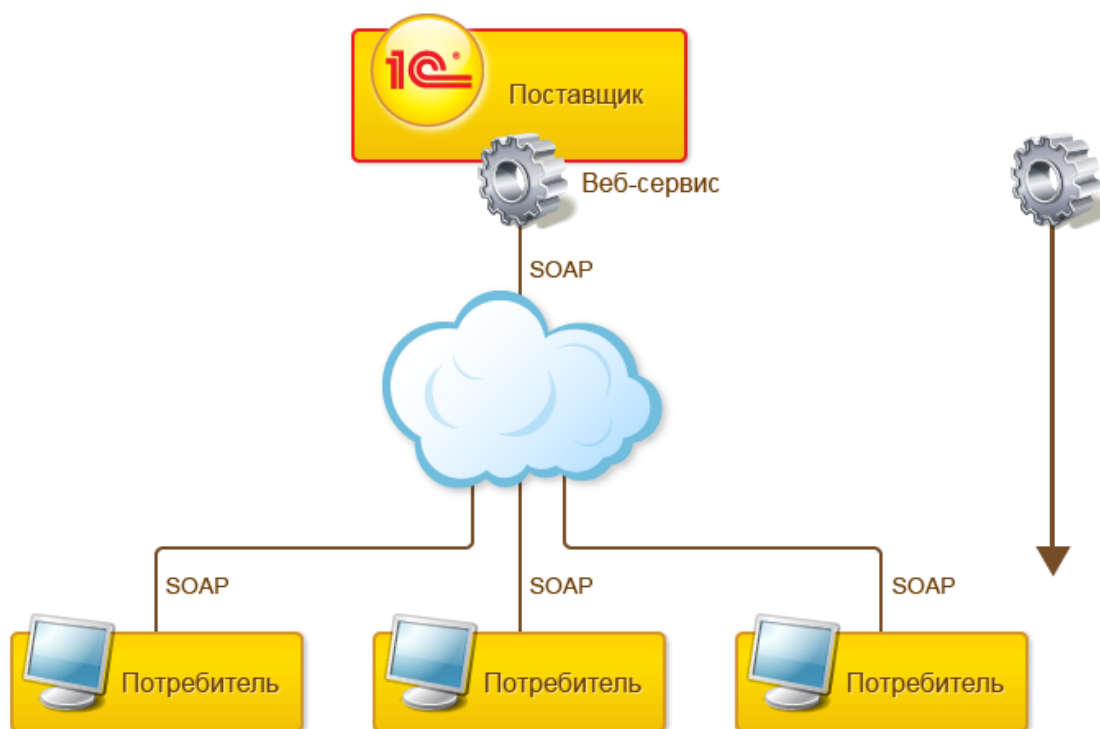


Рисунок 3.13 Механизм работы веб-сервисов

Текущая архитектура позволяет без особых трудозатрат реализовать обмен практически с любой настольной системой, имеющей открытый API для обмена.

Изм.	Лист	№ докум.	Подпись	Дата

4 ЭРГОНОМИКА

4.1 Концепция пользовательского интерфейса

При запуске приложения открывается основная activity (рис. 4.1). Она содержит минимальный набор элементов: элемент выбора даты, список задач (включая краткую информацию о задаче) и кнопка для создания новых задач.

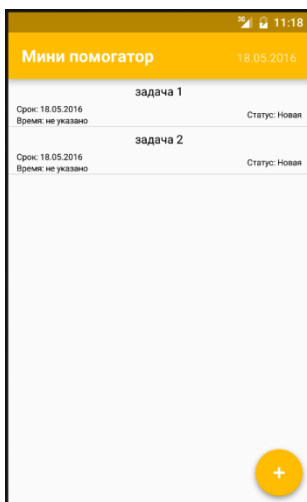


Рисунок 4.1 Основная activity приложения

Если необходимо оперативно сменить статус задачи или удалить задачу и все её подзадачи, можно воспользоваться контекстным меню доступным по длинному нажатию на задаче (рис. 4.2).

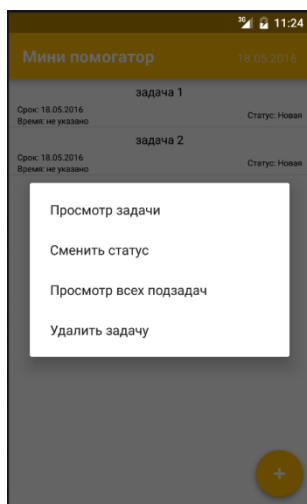


Рисунок 4.2 Контекстное меню задачи

Фильтры, настройки и синхронизация вынесены в выдвигаемое меню (рис. 4.3), войти в которое можно проведя пальцем по экрану мобильного телефона слева направо.

Изм.	Лист	№ докум.	Подпись	Дата

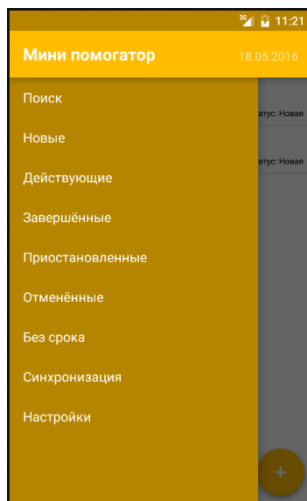


Рисунок 4.3 Выдвижное меню

4.2 Эргономика интерфейса

В библиотеках Android имеются диалоги, которые позволяют взаимодействовать пользователю с интерфейсом приложения оперативно и продуктивно, ниже приведены некоторые из них:

DatePickerDialog – диалог выбора даты (рис. 4.4), в данной работе используется для фильтрации задач по дате и выбора срока выполнения задачи.

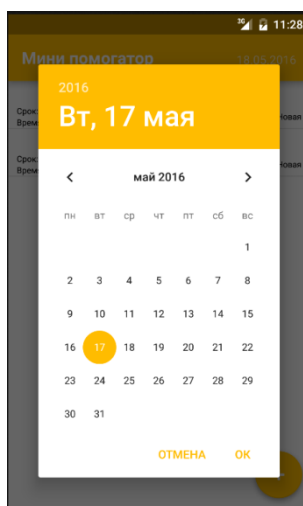


Рисунок 4.4 Диалог выбора даты

TimePickerDialog – диалог выбора времени (рис. 4.5), в данной работе используется для выбора времени выполнения задачи.

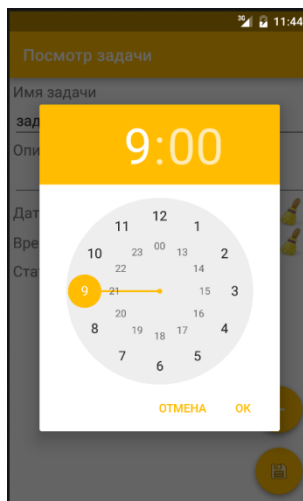


Рисунок 4.5 Диалог выбора времени

Так же для ускоренного набора текста можно использовать голосовой набор, по умолчанию распознавание голоса происходит при подключенном интернет соединении, но загрузив языковой пакет можно пользоваться голосовым вводом в режиме оффлайн. Для активации голосового набора необходимо нажать иконку микрофона на виртуальной клавиатуре (рис. 4.6).

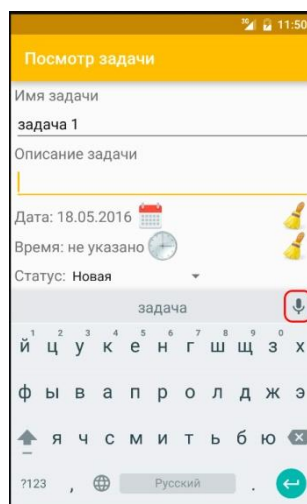


Рисунок 4.6 Кнопка активации голосового набора

ЗАКЛЮЧЕНИЕ

В результате работы была изучена предметная область, выбрана методология разработки Scrum, выбрана операционная система Android в качестве платформы для разработки приложения, Android Studio в качестве среды разработки, разработана модель программного продукта, разработан программный продукт, который позволяет:

- создавать задачи и подзадачи неограниченной вложенности;
- обеспечить оперативный доступ к задачам;
- осуществлять поиск по имени задачи;
- напоминать о задачах;
- контролировать сроки выполнения задач;
- синхронизировать задачи с настольными системами.

В дальнейшем функционал приложения планируется расширить такими функциями как:

- напоминания по GPS координатам
- метки задач
- прикрепление файлов к задачам и др.

					ДП – 230105.65 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		39

ПРИЛОЖЕНИЕ 1. ОРГАНИЗАЦИОННАЯ СТРУКТУРА ПРЕДПРИЯТИЯ

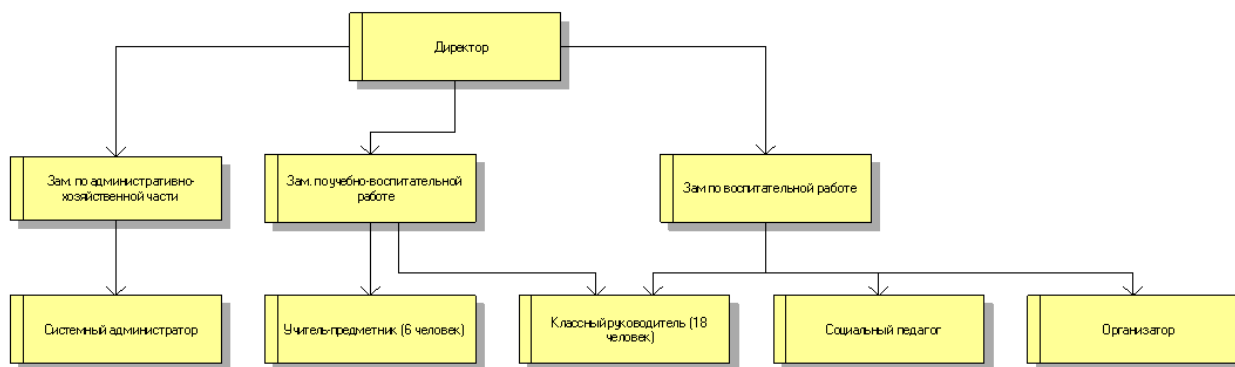


Рисунок 1. Организационная структура предприятия.

Изм.	Лист	№ докум.	Подпись	Дата

ДП – 230105.65 ПЗ

Лист

41

ПРИЛОЖЕНИЕ 2. ДОКУМЕНТИРОВАННЫЙ ЛИСТИНГ ПРОГРАММНЫХ МОДУЛЕЙ

```
//Класс RemindService – служба напоминаний

import android.annotation.TargetApi;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.res.Resources;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.os.Build;
import android.os.IBinder;
import android.support.v7.app.NotificationCompat;
import android.util.Log;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;
import java.util.Timer;
import java.util.TimerTask;

public class RemindService extends Service {

    NotificationManager nm;
    private int lastId = 1;
    public static ArrayList trash = new ArrayList<Task>();

    final String LOG_TAG = "myLogs";

    int interval = 300000; // 5 min
    private Timer timer;
    private TimerTask tTask;

    public RemindService() {
    }
}
```

Изм.	Лист	№ докум.	Подпись	Дата

ДП – 230105.65 ПЗ

Лист

42

```

@Override
public void onCreate()
{
    runAsForeground();
    nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    timer = new Timer();
    schedule();
}

void schedule() {
    SharedPreferences sPref = getSharedPreferences("Preferences", Context.MODE_PRIVATE);
    if (sPref.getBoolean("remind", true)) {
        if (tTask != null) tTask.cancel();
        tTask = new TimerTask() {
            public void run() {
                ArrayList<Task> Tasks = getTasks();
                if (Tasks.size() > 0) {
                    for (int i = 0; i < Tasks.size(); i++) {
                        if ((trash.size() == 0)||(!SearchTask(Tasks.get(i), trash))) {
                            if (checkTimesTasks(Tasks.get(i).date, Tasks.get(i).time)) {
                                sendNotif(Tasks.get(i).name, Tasks.get(i).discription);
                                trash.add(Tasks.get(i));
                            }
                        }
                    }
                }
            }
        };
        timer.schedule(tTask, 1000, interval);
    }
}

public ArrayList getTasks () {
    DatabaseHelper mDatabaseHelper = new DatabaseHelper(this, "tm.db", null, 1);
    SQLiteDatabase db = mDatabaseHelper.getWritableDatabase();

    SimpleDateFormat dateFormatter;
    dateFormatter = new SimpleDateFormat("dd.MM.yyyy", Locale.US);

    Calendar newDate = Calendar.getInstance();
    String selectDate = dateFormatter.format(newDate.getTime());
}

```

					Лист
					43
Изм.	Лист	№ докум.	Подпись	Дата	ДП – 230105.65 ПЗ

```

ArrayList<Task> Tasks = new ArrayList<>();
Task task;

Tasks.clear();

Cursor cursor = db.query("tm", new String[]{DatabaseHelper.ID,
DatabaseHelper.PARENTID, DatabaseHelper.PATH, DatabaseHelper.NAME,
DatabaseHelper.DISCRPTION, DatabaseHelper.DATE, DatabaseHelper.TIME,
DatabaseHelper.STATUS},
        "(" + DatabaseHelper.DATE + " = ? AND " + DatabaseHelper.TIME + " != ?
AND (" + DatabaseHelper.STATUS + " = ? OR " + DatabaseHelper.STATUS + " =
?))", new String[]{selectDate, "", "Новая", "Действующая"}, null, null,
DatabaseHelper.PATH);

if (cursor.getCount() != 0) {
    cursor.moveToFirst();
    for (int i = 0; i < cursor.getCount(); i++) {
        task = new Task(
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.ID)),
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.PARENTID)),
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.PATH)),
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.NAME)),
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.DISCRPTION)),
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.DATE)),
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.TIME)),
            cursor.getString(cursor.getColumnIndex(DatabaseHelper.STATUS)));
        Tasks.add(task);
        cursor.moveToNext();
    }
    cursor.close();
}
return Tasks;
}

public boolean SearchTask (Task task, ArrayList<Task> trash) {
    boolean exit = false;
    for (int i = 0; i < trash.size(); i++) {
        if (task.id.equals(trash.get(i).id))
            exit = true;
    }
    return exit;
}
}

```

					ДП – 230105.65 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

```

public static boolean DeleteTaskFromTrash (Task task, ArrayList<Task> trash) {
    boolean exit = false;
    for (int i = 0; i < trash.size(); i++) {
        if (task.id.equals(trash.get(i).id))
            trash.remove(i);
            exit = true;
        }
    return exit;
}

public boolean checkTimesTasks (String date, String time) {
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy hh:mm");
    String dateForCheck = date + " " + time;
    Date timeTask = new Date();
    try {
        timeTask = dateFormat.parse(dateForCheck);
    }
    catch (ParseException e) {
        Log.d("Error pars timeTask", e.toString());
    }
    Date timeNow = new Date(System.currentTimeMillis());

    SharedPreferences sPref = getSharedPreferences("Preferences", Con-
text.MODE_PRIVATE);
    int timeRemind = Integer.valueOf(sPref.getString("time", "10"));
    long diff = (timeTask.getTime() - timeNow.getTime())/60000;
    if ((diff > 0)&(diff <= timeRemind)) return true;
    else return false;
}

@TargetApi(Build.VERSION_CODES.JELLY_BEAN)
int sendNotif(String title, String text) {
    Context context = getApplicationContext();

    Intent notificationIntent = new Intent();
    PendingIntent contentIntent = PendingIntent.getActivity(context,
        0, notificationIntent,
        PendingIntent.FLAG_CANCEL_CURRENT);

    Resources res = context.getResources();
    Notification.Builder builder = new Notification.Builder(context);

    builder.setContentIntent(contentIntent)
        .setSmallIcon(R.drawable.ic_remind)

```

```

        // большая картинка
        .setTicker("Приближается время выполнения задачи!")
        .setWhen(System.currentTimeMillis())
        .setAutoCancel(true)
        .setContentTitle(title)
        .setContentText(text);

Notification notification = builder.build();

//Стандартный звук и вибро
notification.defaults = Notification.DEFAULT_SOUND |
        Notification.DEFAULT_VIBRATE;

//Светодиодная индикация
notification.ledARGB = Color.YELLOW;
notification.flags = Notification.FLAG_SHOW_LIGHTS;
notification.ledOnMS = 1000;
notification.ledOffMS = 1000;

NotificationManager notificationManager = (NotificationManager) context
        .getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(lastId, notification);

return lastId++;
}

private void runAsForeground(){
    Intent notificationIntent = new Intent(this, RemindService.class);
    PendingIntent pendingIntent=PendingIntent.getActivity(this, 0,
        notificationIntent, 0);

    Notification notification=new NotificationCompat.Builder(this)
        .setSmallIcon(R.drawable.ic_pomogator)
        .setContentTitle("Мини помогатор")
        .setContentText("Сервис запущен.")
        .setContentIntent(pendingIntent).build();

    startForeground(lastId, notification);
    lastId++;
}

@Override
public IBinder onBind(Intent intent) {
    // TODO: Return the communication channel to the service.

```

					ДП – 230105.65 ПЗ	Лист 46
Изм.	Лист	№ докум.	Подпись	Дата		

```

        throw new UnsupportedOperationException("Not yet implemented");
    }

    @Override
    public void onDestroy()
    {
        if (tTask != null)
            tTask.cancel();
    }
}

```

//Класс Search – поиск по задачам

```

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.HashMap;

public class Search extends AppCompatActivity {
    private ListView lv;
    SearchAdapter adapter;
    EditText inputSearch;
    public static boolean flagsearch;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.search);

        lv = (ListView) findViewById(R.id.list_view);
        inputSearch = (EditText) findViewById(R.id.inputSearch);
        inputSearch.addTextChangedListener(new TextWatcher() {

            @Override

```

						<i>Лист</i>
						47
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>ДП – 230105.65 ПЗ</i>	

```

public void onTextChanged(CharSequence cs, int arg1, int arg2, int arg3) {
    // When user changed the Text
    Search.this.adapter.getFilter().filter(cs);
}

@Override
public void beforeTextChanged(CharSequence arg0, int arg1, int arg2,
    int arg3) {
    // TODO Auto-generated method stub

}

@Override
public void afterTextChanged(Editable arg0) {
    // TODO Auto-generated method stub
}
});

// Adding items to listview
adapter = new SearchAdapter(this, Main.Tasks);
lv.setAdapter(adapter);

//Обработка нажатий на элементы списка
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View itemClicked, int position,
        long id) {
        flagsearch = true;
        Main.SelectTask = (Task) parent.getAdapter().getItem(position); //Получаем
экземпляр класса Task
        Intent intent = new Intent(Search.this, ViewTask.class);
        startActivity(intent); //показать activity ViewTask
    }
});
}
}

```