

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра «IT – инжиниринг»

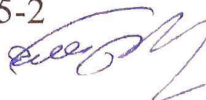
ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой
PhD, доцент
_____ Т.С. Картбаев
« ____ » _____ 2019 г.

ДИПЛОМНЫЙ ПРОЕКТ

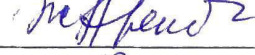
На тему: Разработка мобильного приложения для планирования распорядка дня


Специальность 5B070400 – «Вычислительная техника и программное обеспечение»

Выполнила: Рауанова Альмира Бекболатовна Группа: ВТ-15-2

Научный руководитель: к.ф.м.н., доцент Турганбаев Е.С. 

Консультанты:

по экономической части: к.э.н., профессор  Ж. Г. Аренбаева
« 13 » мая 2019 г.

по безопасности
жизнедеятельности: д.т.н., ст. преп.  Ш.Ш. Бекбасаров
« 14 » мая 2019 г.

по применению
вычислительной техники: ст. преп.  М.Н. Майкотов
« 15 » мая 2019 г.

Нормоконтролер: ст. преп.  А.А. Айтказина
« 15 » мая 2019 г.

Рецензент: зам. дир. ТОО «Сайман» _____ А.С. Алиев
« ____ » _____ 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра «IT инжиниринг»

Специальность 5В070400 – «Вычислительная техника и
программное обеспечение»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Рауановой Альмире Бекболатовне

Тема проекта: Разработка мобильного приложения для планирования
распорядка дня

Утверждена приказом по университету № 124 от «26» октября 2018 г.

Срок сдачи законченного проекта «24» мая 2019 г.

Исходные данные к проекту (требуемые параметры результатов
исследования (проектирования) и исходные данные объекта):
информационные материалы о проектировании систем; международные
стандарты ИСО-9001; данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или
краткое содержание дипломного проекта:

- аналитическая часть;
- проектирование мобильного приложения;
- практическая часть;
- экономическая оценка системы;
- безопасность жизнедеятельности, расчет аспирационной системы для
помещения, где ведется разработка мобильного приложения;
- приложение А. Техническое задание;
- приложение Б. Листинг программы;
- приложение В. Акт внедрения.

Перечень графического материала (с точным указанием обязательных
чертежей): 40 иллюстраций, 14 таблиц.

Основная рекомендуемая литература:

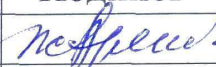



1 Медник З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.

2 Амелин К. С., Граничин О. Н., Кияев В. И., Корявко А. В. Введение в разработку приложений для мобильных платформ. Издательство ВВМ, 2011.

3 Дейтел П. Android для программистов: создаем приложения. Издательство Питер, 2012.

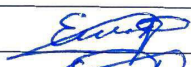


4 Голощапов А.Л. GoogleAndroid. Создание приложений для смартфонов и планшетных ПК. Издательство Питер 2012.

Консультации по проекту с указанием относящихся к ним разделов проекта:

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Аренбаева Ж.Г.	04.03 - 13.05.19	
Безопасность жизнедеятельности	Бекбасаров Ш.Ш.	12.02 - 14.05.19	
Программное обеспечение	Майкотов М.Н.	02.05 - 14.05.19.	
Нормоконтролер	Айтказина А.А.	02.04 - 15.05.19.	

ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Аналитическая часть	15.01.2019	
Проектирование системы	30.01.2019	
Разработка системы	15.01.2019.	

Дата выдачи задания «29» октября 2018 г.

Заведующий кафедрой _____ Т.С. Картбаев

Научный руководитель дипломного проекта _____  Е.С. Турганбаев

Задание принял к исполнению студент _____  А.Б. Рауанова

Аңдатпа

Дипломдық жобаның тақырыбы: «Күн тәртібін жоспарлауға арналған «Planner» ұялы қосымшасын әзірлеу».

Осы дипломдық жобаның мақсаты – күн тәртібін жоспарлау, осылайша өз уақытын тиімді үлестіруге мүмкіндік беретін ұялы қосымшаны әзірлеу болып табылады.

Дипломдық жобаға кіріспе, 7 тарау және қорытынды кіреді. Кіріспеде таңдалған тақырыптың өзектілігі анықталады. Бірінші тарауда жұмыстың мақсаты және орындауға қажетті міндеттері қойылады. Екінші тарауда міндетті шешу үшін технологияларды талдау және таңдау жүргізіледі. Үшінші тарауда әзірлеменің таңдалған құралдары сипатталады. Төртінші тарауда қосымшаны әзірлеудің сипаттамасы, қойылған міндетті практикалық шешімдердің қадам бойынша сипаттамасы қамтылады. Бесінші тарауда қолданушы нұсқаулығы сипатталады. Алтыншы тарауда әзірленетін жобаның экономикалық мақсаттылығының негіздемесі беріледі. Жетінші тарауда әзірленетін жобаның аясында еңбек жағдайларын жақсарту бойынша іс-шаралар қарастырылады.

Аннотация

Тема дипломного проекта: «Разработка мобильного приложения «Planner» для планирования дня.

Цель данного дипломного проекта заключается в разработке мобильного приложения, позволяющего планировать свой распорядок дня, тем самым максимально эффективно распределить свое время.

В дипломный проект входит введение, 7 глав и итоговое заключение. В введении раскрывается актуальность выбранной темы. В первой главе ставится цель разработки и задачи, которые необходимо выполнить. Во второй главе производится анализ и выбор технологий для решения задачи. В третьей главе описываются выбранные средства разработки. Четвертая глава представляет собой описание разработки приложения, содержится пошаговое описание практического решения поставленной задачи. В пятой главе описывается руководство пользователя. В шестой главе производится обоснование экономической целесообразности разрабатываемого проекта. В седьмой главе рассматриваются мероприятия по улучшению условий труда в рамках реализуемого проекта.

Abstract

Theme of the graduation project is “Development of the *Planner* mobile application for planning the day”.

The aim of the graduation project is to develop a mobile application that allows us to plan our daily routine, thereby distributing the time as efficiently as possible.

The graduation project includes an Introduction, 7 chapters and a Conclusion.

The Introduction reveals the topicality of the chosen topic. The first chapter sets the aim and the tasks to be performed. The second chapter analyzes and selects technologies for solving the issue. The third chapter describes the selected development tools. The fourth chapter is a description of the application development, a step-by-step description of the practical solution of the task. The fifth chapter describes the user manual. The sixth chapter contains the justification for the economic feasibility of the developed project. The seventh chapter examines measures to improve working conditions in the framework of the project.

Содержание

Введение	8
1 Постановка задачи	9
2 Анализ предметной области	10
2.1 Описание предметной области	10
2.2 Обзор существующих программных продуктов	10
2.3 Выбор и обоснование варианта реализации задач	12
3 Описание выбранных средств разработки	13
3.1 Android	13
3.2 Язык программирования Java	19
3.3 Eclipse IDE	22
3.4 XML – язык гипертекстовой разметки	23
4 Описание разработки	27
4.1 Описание разработки приложения	27
4.2 Описание формата хранения данных	30
5 Руководство пользователя	33
5.1 Назначение программы	33
5.2 Условия применения	33
5.3 Установка программы	33
5.4 Описание применения	34
5.5 Входные и выходные данные	41
6 Расчет экономических показателей	42
6.1 Расчет себестоимости программного продукта	42
6.2 Определение возможной (договорной) цены ПО	48
7 Безопасность жизнедеятельности	51
7.1 Условия труда	51
7.2 Результаты проделанной работы	52
Заключение	58
Список литературы	60
Приложение А Техническое задание	61
Приложение Б Листинг мобильного приложения	62
Приложение В Акт внедрения	73

Введение

К сожалению, некоторые люди опаздывают гораздо чаще по причинам, которые трудно оправдать, таким как плохое управление временем, промедление или плохая дисциплина. Для этих людей опоздание стало плохой ежедневной привычкой, сравнимой с курением или нездоровым питанием. Мне кажется, что бороться с хроническим опозданием стоит, потому что последствия этой дурной привычки неприятны не только для самих покойников, но и для других. Позвольте привести наиболее важные причины, которые заставляют меня думать, что хроническое опоздание-это проблема, которую стоит решить: стресс: неспособность управлять временем утром и спешка на работу или в школу-худший способ начать день. Если эта ситуация становится нормой, плохая привычка опаздывать может создать долгосрочный стресс, который приносит много проблем, таких как хроническая усталость, увеличение веса, ослабленный иммунитет и более короткий срок службы. Это печально, но проблеск оптимизма появляется, когда мы рассматриваем то, что сказал психолог Дэвид Б., если мы те, кто создает стресс, то мы в лучшем положении, чтобы что-то сделать с этим. У нас больше контроля, чем мы думаем. Очень хорошо, когда опаздывающие люди могут бороться со своим опозданием и избавиться от стресса, им просто нужен небольшой совет, а также некоторый толчок. Некоторые люди могут обратиться за помощью к психологу, некоторые могут пойти купить книгу, например, а некоторые могут просто посмотреть мобильное приложение, которое они могут просто скачать, и пусть он скажет им, что делать. Потерянное время: каждый раз, когда пунктуальный человек должен ждать опоздавшего, он тратит свое время впустую, особенно если он или она остается ждать без какой-либо возможности использовать это время продуктивно, например, сидя в одиночестве в ресторане или нервно ожидая перед кинотеатром.

1 Постановка задачи

В разработке программного обеспечения постановка задачи играет одну из основных ролей, если целью создания является полноценно функционирующий продукт. Максимально подробно и четко проработанные задачи позволят получить стоящие программы.

Задачей данной работы является проектирование и создание программного продукта под операционную систему Android. Необходимо разработать программное обеспечение, позволяющее:

- формировать и редактировать список дел;
- заносить и изменять задания в ежедневник;
- заносить и изменять запись в календаре.

Для достижения поставленных целей и получения необходимых программных продуктов решались следующие задачи:

- анализ требований к программным продуктам;
- выбор средств разработки;
- проектирование интерфейса;
- проектирование формата хранения данных;
- разработка ПО;
- тестирование и отладка ПО;
- дать краткую характеристику каждому этапу.

В процессе анализа были выявлены требования, предъявляемые к подобным программным продуктам, определены основные моменты и нюансы, которые необходимо учесть и реализовать.

На этапе выбора средств разработки подбираются среды и технологии разработки, максимально удовлетворяющие потребностям. Более средства разработки будут освещены позже.

При проектировании интерфейса учитывается его простота.

На этапе проектирования мы создаем модель представления данных для каждого типа задач. Определим архитектуру приложения и систему взаимодействия между пользователем и программой.

Реализация спроектированных подсистем будет осуществлена с использованием наиболее подходящих технологий.

Тестирование целесообразно проводить как на этапе разработки приложений, так и после.

В итоге проведенной работы будет получено полностью работоспособные программы, которые можно будет полноценно использовать в качестве программ для планирования своего времени и хранения контактов.

2 Анализ предметной области

2.1 Описание предметной области

Плanner изначально представляет собой небольшую книгу, содержащую календарь, адресную книгу и блокнот. Плanner служит для организации информации о личных контактах и событиях. С развитием информационных технологий книга стала заменяться сначала электронными планнерами, затем карманными персональными компьютерами, компьютерными программами и онлайн-органайзерами, которые могут обладать дополнительными функциями: напоминание о предстоящих событиях, защита и синхронизация информации.

Плanner является средством управления временем. Предварительное планирование дел помогает повысить плодотворность любой деятельности, как личной, так и профессиональной. Поэтому и человека, который консультирует организации и частных лиц в сфере повышения эффективности управления временем, тоже иногда называют «органайзер». В США даже существует национальная ассоциация профессиональных органайзеров (паро).

В настоящее время под планнером чаще понимается именно его программная версия для персонального компьютера или мобильного устройства. Их безусловными преимуществами являются большой объем вводимых данных, установка автоматических напоминаний. Формы органайзера могут быть разными. Есть, к примеру, узкоспециализированные планнеры: органайзер рецептов, органайзер финансов, органайзер музыкальной коллекции.

Однако при использовании программной версии планнера возникает проблема в лице реальной приближенности к пользователю, сопровождение его везде и всюду, при этом сохраняя актуальность данных. Таким образом всплывает необходимость синхронизации данных, либо нахождения их в таком месте, откуда их всегда можно получить. Второй метод используется в онлайн-планнерах. Первый способ используется на порядок реже. В большинстве случаев, если синхронизация между органайзерами на разных платформах и есть, то через интернет.

Используемыми платформами для персонального компьютера являются Windows, mac OS и Linux. В свою очередь для мобильных устройств: IOS, Android и Windows phone.

2.2 Обзор существующих программных продуктов

В этот период имеется большое число планеров, равно как подвижных, таким образом и настольных. Биржа подвижных дополнений довольно обилена, проанализируем определенные подвижные дополнения с целью планирования вопросов с торговых центров дополнений, подобных равно как Гугл Play, App Store и Windows Store. Гугл Play-app store с Гугл, дающий

собственникам приборов с операторной концепцией Android приобретать и определять разнообразные дополнения. В Гугл Play возможно отыскать большое количество нужных и различных дополнений. В торговом центре имеется равно как коммерческие, таким образом и безвозмездные проекты. 31 октября 2012 года Гугл заявила, то что число дополнений достигло 700 000, а число скачиваний достигло ДВАДЦАТЬ ПЯТЬ млрд app Store, app store, пункт сеть интернет-торгового центра itunes Store, включающий разнообразные дополнения с целью подвижных телефонных аппаратов iPhone, плееров iPod Touch и планшетов iPad, а кроме того с целью личных ПК Mac дает возможность им приобретать либо использовать безвозмездно. App Store дает наиболее 1,5 млн. дополнений с целью iPhone и iPod Touch и приблизительно 725 000 дополнений. с целью iPad (согласно каприз в ДЕСЯТИ Июня 2018 года), количество загрузок перевалило 100 млрд, а пользовательская основа является режима 575 млн. индивид.

Windows Phone-данное торговый центр дополнений Microsoft с целью своей моб. платформы Windows Phone, что дает возможность юзерам определять и покупать разнообразные дополнения и вид развлечения. Сеть интернет-торговый центр доступен равно как с Windows Phone, таким образом и с браузера. В завершении июня 2012 года Microsoft формально представила больше 100 тыс. дополнений, общедоступных в торговом центре. В ходе созыва сведений с целью постановления проблемы автоматизации планирования работы был проанализирован биржа программных товаров, сподручных найти решение установленную проблему. В свойстве критериев сопоставления существовали отобранные: присутствие безвозмездной версии присутствие уведомлений Вероятность отыскивания вопросов публичный сокет с целью размена сведениями с настольными концепциями.

Любой.do in Гугл Play-данное обычный блок планирования вопросов с дизайном в российском стиле. Кто именно-нибудь.do обладает минималистичный проектирование. В данном дополнении, проблемы формируются равно как персонально, таким образом и в планах. Все без исключения возможно распланировать в ряд суток в будущем. В Каждом.имеется определенные увлекательные характерные черты. К примеру, вам сможете убрать произведенные проблемы, встряхнув телефон. Характерные черты: дополнение вопросов и подзадач формирование списков напоминаний вероятность обмениваться с иными юзерами. Имеется точно узкая безвозмездная модификация.

ToDoist в Гугл Play-план, что возник в 2007 г. и с этих времен регулярно формируется. К 2013 г. число юзеров ToDoist достигла сокровенной числа в 1 1000000 юзеров, и в настоящий период, согласно определенным сведениям, наиболее 4 млн. ToDoist обладают простой и обычный сокет и в в таком случае ведь период отличную работоспособность. Характерные черты: дополнение вопросов и планов, вероятность обеспечить допуск к каждому плану сотрудникам, товарищам, членам семьи с целью коллективной деятельность надо едиными проектами и мишенями разделение вопросов в подзадачи

информация согласно электрической почте либо отбор push-уведомлений согласно задачам (только лишь в коммерческой версии).

Wunderlist в Гугл Play, App Store и Windows Store сконструирован германской установкой разработчиков программного обеспечения б Wunderkinder GmbH. Wunderlist весьма схож в ToDoist. Почти подобная ведь работоспособность, обычный и доступный сокет, вероятность коллективной деятельность, однако почти все без исключения функции в Wunderlist безвозмездные с целью Android и ios, таким образом и с целью подвижных приборов с операторной концепцией Windows. Характерные черты: дополнить проблемы и подзадачи (уменьшить число подзадач в безвозмездной версии), совершая перечни определить последний период с целью предотвращения упражнения согласно электрической почте либо push-уведомления отыскивания согласно проблеме. OmniFocus в App Store-популярный менеджер задач IOS, разработанный Omni Group. Стоимость программы для iPhone составляет \$ 19.99. Приложение славится тем, что можно создать неограниченное количество проектов и составить любую иерархию. Существует настольная версия для MasOS, стоимость стандартной и Pro версии программы составляет \$ 39.99 \$ 79.99 соответственно. Особенности: добавление задач и подзадач, составление списков, установка крайнего срока выполнения заданий, Push-уведомления, Поиск задач.

ToDo в App Store-отличный инструмент для управления временем с удобным интерфейсом и набором интересных функций. Среди преимуществ утилиты-красочный современный интерфейс, гибкость и простота использования, а также планирования перетаскивания. Приложение предоставляет проекты и контрольные списки, полнофункциональный поиск, повторяющиеся задачи. Стоимость мобильного приложения для iOS составляет \$ 4.99. Существует также настольная версия программы для MacOS, ее стоимость составляет \$ 14.99. Особенности: добавление задач и подзадач список повторяющихся задач push-сообщения Поиск по заданиям.

2.3 Выбор и обоснование варианта реализации задач

В большинстве случаев планнеры перегружены оформлением и возможностями, которые зачастую и не используются пользователем. Потому было решено вернуться к первоначальным истокам планнера и реализовать календарь, ежедневник, справочник и возможность составлять заметки и список дел. Платформа android является самой молодой из всех распространенных мобильных операционных систем. Не смотря на свою «юность», android уже занимает львиную долю рынка, потому для мобильной версии приложения была выбрана именно эта система.

3 Описание выбранных средств разработки

Нужно разъяснить, по какой причине с целью формирования подвижных дополнений подобрана площадка Android. Проблема в этом, то что ОС Android довольно простой в изучении, дает обширные способности с целью исследования дополнений, и в том числе и вероятность соперничать с опытнейшими программерами. С целью данного имеется ряд факторов:

- по данным службы глобального мониторинга Netmarketshare на май 2016 года, гаджеты на базе этой операционной системы занимают 60,99% рынка;

- более мягкая политика по отношению к разработчикам делает рынок мобильных приложений Android менее статичным и, следовательно, более открытым для инноваций;

- то же приложение может быть разработано как для смартфонов и планшетов, а также для Android Wear, Android TV, Android Auto и даже Google Glass;

- широкий спектр областей, в которых может осуществляться Разработка приложений для Android.

3.1 Android

Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. AndroidNativeDevelopmentKit позволяет портировать (но не отлаживать) библиотеки и компоненты приложений, написанные на C++ и других языках. ОС Android установлена на 86% смартфонов (2014).

3.1.1 Основные факты о платформе Android

Никак не тайна, то что 21 столетие-данное столетие информативных технологий. Возлюбленная начала составляющей нашей существования: возлюбленная охватывает нас повсюду. Равно как принцип, индивид, налегающий в акселератор газа в собственном машине, способен в том числе и никак не понимать, то что из-за его перемещений наблюдают сотня измерителей и микропроцессоров, вызванных упростить ему жизнедеятельность, а зачастую и уберечь её. В инновационные автомобили изготовители вводят все без исключения наиболее непростые функции. По причине данного появляется потребность в подходящем содействии среди Юзером и целой находящейся вокруг инфраструктурой. Появилась потребность применять электронику дображивающего управления. Инженеры Apple, а далее Гугл, обнаружили разрешение. Они сформировали операторную концепцию с целью телефонных аппаратов, что дает возможность просто создавать посторонние дополнения, включая с

способности просматривать электрические книжки с экрана подвижных телефонных аппаратов и завершая правлением домашней техникой в "разумном доме".

Google направился далее в собственных принципиальных проектах и основал раскрытую зодчество Android. С целью Android (Робот) переносная (сетная) автооперационная концепция с целью коммуникаторов, планшетных ПК, электрических книжек, числовых проигрывателей, наручных времен, нетбуков и смартбуков, базирующаяся в ядре Linux. Поначалу созданный Android Inc., что далее приобрел Гугл.

Более 75 % телефонов, реализованных в 3-ем микрорайоне 2015 годы, существовали оборудованы операторной концепцией Android. Сейчас любой создатель электрического аппарата обладает вероятностью изменить Android около собственное прибор, этим наиболее гарантировав сопоставимость оснащения с посторонними прибавлениями с целью данной ОС. Данное существовало весьма рентабельно. В случае если вплоть до выхода Android любой изготовитель электрических приборов без помощи других слагал либо приобретал у кого-в таком случае операторную концепцию, этим наиболее лишаясь избытие нужных проектов, сформированных программерами согласно целому обществу, в таком случае уже после выхода ОС Android изготовители зачастую встречаются с проблемой, Тот или иной версию Android им необходимо сохранять.

Значительную значимость сразилось возникновение дроида и в общении среди народов. Монтаж его в телефонные аппараты дала возможность просто создавать новейшие модификации подвижных приборов, увеличивая перечень возможностей-равно как телефонных аппаратов, таким образом и наиболее операторной концепции. Возникновение проектов, вызванных подействовать юзеру в разных моментах (к примеру, имеется дополнение, что применяет интегрированные измерители с целью замера углов поворота, быстроты предмета и этому такое), повергло к этому, то что на сегодняшний день люду, то что направляется в поездка, довольно попросту обладать подвижное прибор в Android. Юзеру даются обслуживание бронирования гостиниц, отыскивания авиабилетов, разнообразные путеводители, а кроме того намеренно нагруженные игра в карты, какие применяются с целью отыскивания и маршрута вплоть до участка направления. Присутствие применения SIP-телефонии вероятен междугородная и интернациональная взаимосвязь [4].

Все данное, наравне с воздушностью исследования дополнений, создает платформу одной с наиболее многообещающих с целью общения в нынешнем мире.

3.1.2 Ключевые особенности Android

Android считается единой операторной концепцией (ОС) с целью подвижных приборов – телефонных аппаратов и планшетов. Данная

концепция обладает большое число характерных качеств, какие совершают её известной и заманчивой с целью значительного числа юзеров согласно целому обществу.

Операционная концепция Android нетребовательна и может функционировать в разных конфигурациях. Непосредственно по этой причине большая часть всемирных изготовителей оборудуют собственные аппарата данной ОС, так как прочие программные продукты питания предусмотрены с целью единичных приборов, соответствующих конкретным условиям. Подобная эластичность Android определена этим, то что концепция выстроена в ядре Linux, что обладает публичный первоначальный шифр, то что предоставляет безграничные способности создателям. Android возможно включать в приборах с меньше нежели 256 МБ своевременной памяти. Новые версии концепции призывают 512 МБ своевременной памяти, то что кроме того считается незначительным смыслом с целью нынешних приборов.

Система никак не потребует высокопроизводительного процессора и способен функционировать в приборах, оборудованных ядром 600 МГц. Операционная концепция дает возможность определять дополнения с официозного репозитория Гугл, что дает наиболее значительную в обществе основу проектов. Данное сопряжено с этим, то что любой создатель способен без помощи других составить различную план в прибор и разместите его в торговом центре. Вероятность кроме того реализуется вследствие открытости операторной концепции. Нужно выделить, то что дополнения в приборах около правлением Android возможно определить напрямую с телефонного аппарата либо планшета, и посредством пк, скачав документ.apk и его последующая монтаж в прибор.

Отличительной характерной чертой Android считается его объединение с сервисами Гугл-Gmail, Hangouts, голосовым отыскиванием и др. В Android формально выполнена помощь chrome, что дает возможность хронировать вкладки, раскрытые в браузере в телефоне с браузером пк. Например, вам сможете приступить показ с телефонного аппарата и, в случае если желаете, продлить исследование данных, раскрыв эту ведь вкладку в пк в отсутствии потребности вторичного отыскивания.

"Андроид" обладает достаточно обычный и подсознательно доступный сокет. Все без исключения требуемые дополнения вмещаются в основном экране и в список, что возбуждается согласно нажатию Основной воспринимающей фортепьяно либо надлежащей клавиши в экране. Все без исключения опции пребывают в области "Настройки", и любое процесс юзера разъясняется объяснениями и рекомендациями присутствие первоначальный запуске аппарата. Автооперационная концепция стремительно откликается в камарильи юзера, определяет и занимает требуемые проекты и комп.данные с быстротой, никак не уступает иным нынешним подвижным операторным концепциям.

3.1.3 Конкуренцы платформы Android

Если раньше рынок мобильных платформ делился как минимум на 5-6 компаний (Symbian, Blackberry OS, Android, Palm, iOS, Bada и др), то сейчас рынок мобильных платформ), на данный момент существуют только 2 основные платформы: iOS и Google Android, на которые приходится около 90% всего рынка. Для Google Android в четвертом квартале 2015 года достиг 75%, то есть три для смартфонов под управлением мобильной платформы от Google. Ниже приведена таблица сравнения разных операционных систем (см. таблицу 1) и данных для разных версий Android (см. таблицу 2). Статистика, собранная за двухнедельный период на основе звонков в сервис Google Play.

Таблица 1. Мировые продажи смартфонов, по типам ОС, 1-й квартал 2015

ОС	Продажи в 2015 г., тыс.	% рынка	Продажи в 2014 г., тыс.	% рынка
Android	81 067,4	56,1	36 350,1	36,4
iOS	33 120,5	22,9	16 883,2	16,9
Symbian	12 466,9	8,6	27 598,5	27,7
Blackberry OS	9 939,3	6,9	13 004,0	13,0
Bada/Linux	3 842,2	2,7	1 862,2	1,9
Windows	2 712,5	1,9	2 582,1	2,6
Прочие	1 242,9	0,9	1 495,0	1,5
Суммарно	144 391,7	100,0	99 775,0	100,0

Таблица 2. Распределение версий ОС Android, на 1-е августа 2015

Название версии	Уровень API	Распределение
1.5 Cupcake	3	0.2%
1.6 Donut	4	0.5%
2.1 Eclair	7	4.2%
2.2 Froyo	8	15.5%
2.3 – 2.3.2 Gingerbread	9	0.3%
2.3.3 – 2.3.7	10	60.3%
3.1 Honeycomb	12	0.5%
3.2	13	1.8%
4.0 – 4.0.2 Ice Cream Sandwich	14	0.1%
4.0.3 – 4.0.4	15	15.8%
4.1 Jelly Bean	16	0.8%

Android это самая переносная (сетевая) автооперационная концепция с целью коммуникаторов, планшетов, электрических книжек, числовых плееров, времен, нетбуков и смартбуков в основе ядра Linux. Поначалу созданный Android Inc., что далее приобрел Гугл. В дальнейшем Гугл стимулировала формирование альянса Open Handset Alliance, что увлекается помощью и последующим формированием платформы. Android дает возможность формировать Java-дополнения, какие распоряжаются механизмом посредством библиотеки Гугл developer. Android Native Development Kit формирует дополнения, прописанные в С и иных стилях.

Для сочинения программного предоставления с целью данной платформы следует применять SDK (software development kit). SDK-данное комплект с целью исследования, что дает возможность специалистам формировать дополнения с целью определенного программного пакета базисного программного предоставления с целью исследования, аппаратной платформы, компьютерной концепции, игровых консолей, операторных концепций и иных платформ. В данном случае конкретной платформой считается Android, и возлюбленная соответствует Android SDK, что применяется.

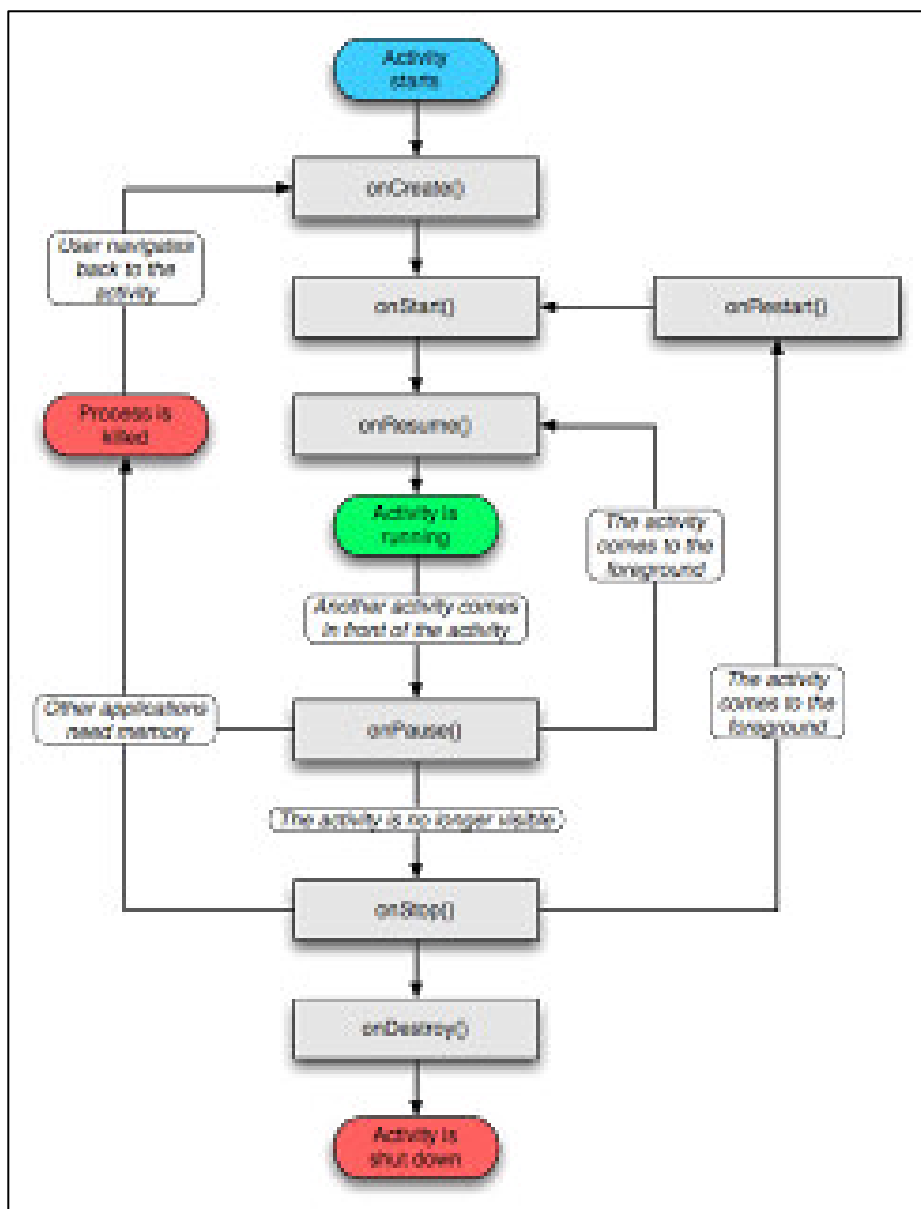


Рисунок 3.1 Жизненный цикл приложения для системы

Характерной особенностью простых редакторов с целью сочинения программный код считается присутствие значительного перечня возможностей, что дает возможность включать испытание и настройка начального программный код, производить оценку план в порядке сопоставимости с разными версиями ОС Android и следить из-за итогом в порядке настоящего периода (рекомендательно). Удерживает огромное число подвижных приборов, из числа каковых: подвижные телефонные аппараты, планшетные ПК, смарт-очки (в этом количестве Гугл Glass), инновационные машины с ребордными ПК в ОС Android, ТВ с наращенной функциональностью, специализированные виды времен и многочисленные прочие подвижные гаджеты, единые промышленные аппарата.

Приложения с целью Android применяют Windows (схожие в Windows) в собственной труде, однако в данной концепции вышеуказанные окошки называют activity. Равно как и в Windows, любое окошко обладает

собственный личный актуальный оборот и характерные черты. Присутствие формирования новейшего окошки возбуждается способ onCreate (), данный способ меняется в период исследования и в немой начивается дополнение и его элементы.

Затем возбуждаются способы onStart() и onResume (). Тот и другой способа возбуждаются к окошко, если некто основан либо реконструирован (присутствие переключении с иной проекты, присутствие развертывании скатанного дополнения и т. д.).

При свинчивании возбуждаются способы onPause() и onStop (). Если дополнение и окошки прикрыты, возбуждается onDestroy (), и пользовательские сведения и опции имеют все шансы являться сбережены в данном способе. Абсолютное представление и очередность призыва способов возможно отыскать в официозном веб-сайте. Единая модель актуального цикла дополнения с целью Android представлена в рис. 3.1.

3.1.4 Android Studio

Применяемая сфера исследования- Android Studio-формальный механизм с целью исследования android-дополнений [1], что включает SDK tools (с англ. Software Development Kit - комплект с целью исследования, что дает возможность экспертам согласно программному обеспечиванию формировать дополнения с целью определенного программного пакета). Android Studio кроме того содержит оптимальный эмулятор (проект, что дает возможность эмулировать иную выступ в индивидуальном ПК либо видеоигровой рента). Некто формирует в трудовом обеденном месте числовой подобие телефона, в коем создатель способен испытать план с поддержкой курсора мыши, а никак не перстом. Из числа проектов эмулятора с целью пуска и испытания дополнений с целью android более популярным считается Android Virtual Device (AVD), что поставляется в наборе с SDK.

Android Studio-новейшая и целиком встроенная сфера исследования дополнений (IDE), не так давно выданное Гугл с целью операторной концепции Android. Данный продукция рассчитанный с целью предоставления создателям новейших приборов с целью формирования дополнений, а кроме того дает альтернативный Eclipse, что в наше время период считается наиболее распространенной сферой исследования. Ключевые способности-выполнена вероятность верстки в порядке настоящего периода, имеется большое количество альтернатив объемов и дозволений экрана и почти все иное. Скачать новейшую сферу исследования возможно в официозном веб-сайте Android Studio [1].

В Android Studio дает возможность для вас заметить всевозможные зрительные перемены, какие вам совершаете в порядке настоящего периода в дополнении. Вам кроме того сможете заметить, равно как ваше дополнение

станет смотреться в различных приборах Android в то же время, с разными опциями и дозволением экрана.

Продукт кроме того обладает новейшие приборы с целью упаковки и маркировки программный код. Данное даст возможность для вас никак не заблудиться в плане, если вам обладаете проблема с огромным числом программный код. Проект кроме того содержит функцию перетаскивания, что дает возможность передвигать элементы посредством общепользовательский сокет.

Кроме этого, новейшая сфера исследования обладает функцию размена оповещениями Гугл Cloud, что дает возможность отсылать сведения с сервера в Android-приборе посредством скопление. Данное хороший метод отсылать push-уведомления в ваши проекты.

Вы кроме того сможете применять дополнение с целью локализации дополнений. Данное даст возможность для вас программировать, удерживая присутствие данном надзор надо прибавлением.

Что ещё дает Android Studio?

- environment надежная и простая среда разработки;
- легко проверить производительность приложений на различных типах устройств;
- Help помощники и шаблоны для общих элементов программирования для Android;
- full полнофункциональный редактор с множеством дополнительных инструментов для разработки приложений [23].

Таким образом, набор необходимых программ очень мало. И вместе эти программы предоставляют достаточное количество инструментов для построения приложений.

3.2 Язык программирования Java

Разработка приложений для платформы Android ведется преимущественно на языке Java. Для создания программ на Java необходимо специальное программное обеспечение. Последние версии этого программного обеспечения можно скачать с официального сайта разработчика Oracle Corporation. Этот программный пакет включает такие инструменты как JRE (Java Runtime Environment) и JDK (Java Development Kit). Первый инструмент-это среда выполнения-минимальная реализация виртуальной машины, которая работает и выполняет Java-код. Второй инструмент-это, в свою очередь, целый набор инструментов, Набор для разработки приложений Java. Фактически, JRE также является частью JDK, как и различные стандартные библиотеки классов Java, компилятор javac, документация, примеры кода и различные утилиты. Весь этот набор распространяется бесплатно и имеет версии для разных операционных систем, поэтому любой желающий может скачать и использовать его. JDK не входит в состав IDE, предполагается, что разработчик установит отдельно. Существует

множество IDE для разработки Java, таких как NetBeans, IntelliJ IDEA, Borland JBuilder и другие. Таким образом, перед началом разработки программы на базе ОС Android необходимо подготовить инструментарий [6]. При разработке на Android необходимо использовать Eclipse IDE для разработчиков Java. На сайте вы можете найти и скачать SDK для вашей платформы (см. рис. 3.1).

Объектно-ориентированный язык программирования, разработанный Sun Microsystems. Приложения Java обычно компилируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры.

Программы на Java транслируются в байт-код виртуальной Java-машины (JVM) и программа, которая обрабатывает байтовый код и передает инструкции оборудованию как интерпретатор.

Преимущество такого способа выполнения программ и в полной независимости байт-кода от операционной системы и оборудования, позволяющего запускать Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, превышающие установленные разрешения программы, немедленно прерываются.

Часто к недостаткам концепции виртуальной машины относят то, что исполнение байт-кода виртуальной машиной может снижать производительность программ и алгоритмов, реализованных на языке Java. В последнее время был сделан ряд улучшений, которые немного увеличили скорость Java-программ:

- применение технологии трансляции байт-кода в машинный код во время выполнения техники (СИТ), с возможностью сохранения версий класса в машинном коде;
- extensive широкое использование платформенно-ориентированного кода (собственного кода) в стандартных библиотеках;
- hardware аппаратное обеспечение, что обеспечивает быструю обработку байтового кода (например, технология Jazelle, поддерживаемая некоторыми процессорами ARM компании).

Choose the SDK package for your OS from the table below.

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions.

- ▼ [USE AN EXISTING IDE](#)
- ▼ [SYSTEM REQUIREMENTS](#)
- ▼ [DOWNLOAD FOR OTHER PLATFORMS](#)

ADT Bundle

Platform	Package	Size	MD5 Checksum
Windows 32-bit	adt-bundle-windows-x86.zip	425429957 bytes	cca97f12904774385a57d542e70a490f
Windows 64-bit	adt-bundle-windows-x86_64.zip	425553759 bytes	c51679f4517e1c3ddefa1e662bbf17f6
Mac OS X 64-bit	adt-bundle-mac-x86_64.zip	390649300 bytes	f557bc61a4bff466633037839771bffb
Linux 32-bit	adt-bundle-linux-x86.zip	418614971 bytes	24506708af221a887326c2a9ca9625dc
Linux 64-bit	adt-bundle-linux-x86_64.zip	418889835 bytes	464c1fbc92ea293d6b2292c27af5066a

SDK Tools Only

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r21.1-windows.zip	99360755 bytes	dbece8859da9b66a1e8e7cd47b1e647e
	installer_r21.1-windows.exe (Recommended)	77767013 bytes	594d8ff8e349db9e783a5f2229561353
Mac OS X	android-sdk_r21.1-macosx.zip	66077080 bytes	49903cf79e1f8e3fde54a95bd3666385
Linux	android-sdk_r21.1-linux.tgz	91617112 bytes	3369a439240cf3dbe165d6b4173900a8

Рисунок 3.1 Android SDK

По данным сайта shootout.alioth.debian.org, для семи разных задач время выполнения на Java составляет в среднем в полтора-два раза больше, чем для C/C++, в некоторых случаях Java быстрее, а в отдельных случаях в 7 раз медленнее. С другой стороны, для большинства из них потребление памяти Java-машиной было в 10-30 раз больше, чем программой на C/C++. Также примечательно исследование, проведённое компанией Google, согласно которому отмечается существенно более низкая производительность и

большее потребление памяти в тестовых примерах на Java в сравнении с аналогичными программами на C++.

Идеи, заложенные в концепцию и различные реализации среды виртуальной машины Java, вдохновили множество энтузиастов на расширение перечня языков, которые могли бы быть использованы для создания программ, исполняемых на виртуальной машине. Эти идеи нашли также выражение в спецификации общезыковой инфраструктуры CLI, заложенной в основу платформы .NET компанией Microsoft.

3.3 Eclipse IDE

Безвозмездная встроенная сфера исследования модульных кроссплатформенных дополнений. Eclipse Foundation согласно исследованию и помощи. Eclipse дает платформу исследования с расширениями, по этой причине возлюбленная востребована: каждой создатель способен увеличить Eclipse с поддержкой данного модуля. Фирма IBM, приборы стиля Преисподняя (GNATbench, жаровня), ЯЗЫК, Язык, PHP и т. д. с инженерами ОС QNX ранее изобретены приборы Java-исследования (JDT, специализированным), Си/Си++ средства исследования (ЦДТ). С различных создателей. Сферы Eclipse, дополняющие многочисленные расширения, содержат в себе админов двор сведений, серверные дополнения и почти все иное.

Eclipse Java Development Tools (JDT) TM — это плагин с целью концепций управления версиями-CVS, GIT и иных концепций, встроенных с коллективной сферой исследования, направленной в наиболее известные модули. Возлюбленная кроме того гарантирует помощь взаимосвязи среди IDE и концепциями управления вопросами (погрешностями). Bugzilla регулярно новейшая помощь введена в основной комплект, и удерживает большое число расширений, какие прочие трекеры (Trac, Jira и т. д.). Из-за собственную непорочность и значительного свойства eclipse считается обычной исследованием дополнений с целью множества учреждений.

Написано в Eclipse Java, по этой причине некто ликвидирует тонкие самостоятельные продукты питания (см. далее) с целью абсолютно всех единых зон с целью исследования библиотеки SWT. Взамен применения библиотеки SWT эталоном считается книгохранилище Java-заглушки. Целиком принято в базисную платформу (операторную концепцию), обеспечивающую стремительный и природный тип пользовательского интерфейса, однако в некоторых случаях порождающую сопоставимость и использование в разных платформах.

3.4 XML – язык гипертекстовой разметки

XML(eXtensible Markup Language) развиваемый речь разметки. Речь разметки, представленный консорциумом World Wide Интернет, в самый-

самом процессе предполагает собою комплект единых синтаксических законов. XML-данное текстовой параметра, применяемый с целью понятия и сохранения высокоструктурированных сведений (имеющихся двор сведений, файлов), размена данными среди проектами с целью формирования специальных стилей разметки(к примеру, XHTML) в его базе. Речь именуется расширяемым, так как некто никак не занимается, применяется в документации согласно тега с целью формирования тега в базе нужд конкретной сфере, узкой только инструкциями синтаксиса стиля. Увеличение XML с целью определенной грамматики формируется в XML и отражается предварительно, теги и их свойства, а кроме того комплект законов, какие устанавливаются, которые свойства и компоненты имеют все шансы являться составляющей иных компонентов. Совокупность обычного синтаксиса формата базируется в людском, расширяемости, с целью предметов и обширного диапазона языковых программных денег с целью настоящего XML и множества высококлассных выводных XML, с обширным применением шифровки Unicode, для того чтобы послужить причиной акт к отображению его охватываемого.

Установление эталона обладает 2 степени правильности XML-важного документа:

Правильно создавать. Тот или иной XML-акт-данное принцип, что используется к целому единого синтаксиса XML в согласовании с хорошо созданных бумаг. Неверно расположенный акт невозможно рассматривать XML-важным документом, что обработчик XML(парсер)никак не обязан подвергать обработке простым методом, следует систематизировать обстановку равно как неотвратную погрешность.

Доступный. Настоящий акт кроме того обозначает выполнение различных законов. Данное наиболее жесткая вспомогательная контроль правильности бумаг в соотношении предварительно конкретным, однако внешне положением, с мишенью минимизации числа погрешностей, к примеру, текстуры и конфигурации определенного важного документа либо собрания бумаг.

3.5 C++ - язык программирования

Парадигмы программирования, какие удерживают данный речь, содержат процедурное кодирование (на (машинном, общие программные модули с целью объектно-нацеленного программирования, отдельную компиляцию, обрабатывание изъятий, получение сведений и виды(игра) предметов оглашений. Обычная книгохранилище содержит, из числа остального, единые контейнеры и методы. Равно как в стилях значительного, таким образом и невысокого степени совмещают свойства C++. Согласно сопоставлению с предшествующим, в стиле C, данное объектно-направленное и общее кодирование (на (машинном, что удерживает максимальное интерес. Синтаксис C++наследован с стиля C. Единственный с целью помощи

сопоставимости с принципом проектирования С. Но С++ не считается точно надмножеством дополнений С, какие имеют все шансы выполнять большое количество в одинаковой мере эффективных компиляторов С, а компиляторы С++ очень крупные, однако никак не введены в все без исключения вероятные проекты в С.

Основная теория С-класса. Игра-данное конкретные юзером виды. Группа, обеспечивающий сведения новейшему спектру модифицированных, личных инициализаций, литофанического переустройства пользовательских видов, динамического вида, опции контролируемого юзером управления памятью и прибора с целью деятельность с перегрузкой. В С++ концепция модульного возведения проекты управления вводом выполнена наиболее подробно, нежели С. Помимо этого, усовершенствования С++ не сопряжены напрямую с классом: символьная постоянная, роль смены значимости характеристик типичных функций, перегруженность фамилий функций, руководство памятью, проблемы и справочные виды.

3.6 Фреймворк Qt

Основные принципы проект способствует формированию и интеграции разных частей значительного программного плана. Qt считается бесплатной (переносной) рамкой. Некто располагается в С # с целью исследования дополнений, какие в основном применяются, а кроме того функционирует в стилях количеств, подобных равно как Python, Ruby, Java, PHP и прочие.

Уникальная отличительная черта библиотеки Qt и применения компилятора meta object (МОС) и предшествующая концепция обрабатывания начального программный код (равно как принцип, книгохранилище Qt Иф с целью чистейшего С++, однако данное особый нехватка в "конвертере" МОС с целью последующей компиляции обычным компилятором с++). МОС возможно увеличить в библиотеку освещение, включив подобные определения, равно как слоты и сигналы. Помимо этого, данное создает вам наиболее лаконичный шифр. Программа МОС дополняет макрос q_object, то что группа включает установления С++ из файла в заголовке, что разыскивает документ в цель-шифр, включающий С++.

Механизм сигналов и слотов целиком замещает форма функции противоположного призыва, некто весьма эластичный и целиком объектно-направленный.

Сигналы и слоты Имеют все шансы являться подсоединены товарищ к товарищу, а сигналы имеют все шансы являться подсоединены к огромному числу слотов. Слоты кроме того имеют все шансы являться подсоединены к большинству сигналам. В случае если разъем ровным счетом ничего, помимо отправки входного сигнала с целью прибавления, вам никак не сможете, а попросту объединяете знак товарищ с ином. Имеются специализированные игра текстов в установлении итогового сигнала и слота-слота. В данном

случае разъем считается популярным способом стиля C++ и способен являться введен в свое установление доступных и оберегаемых, индивидуальных вариантов. Вам никак не понимаете знак программный код с целью осуществления. Применяя передачу сигнала, основные слова "выпущенный". Группа, включающего сигналы и слоты, обязан являться унаследован с каждого `QObject` К либо с класса, унаследованного с данного класса. Объединение знак-разъем в каждое период возможно убрать (объединения) с поддержкой способа `disconnect()`, однако данное весьма крайне редко, таким образом равно как присутствие удалении абсолютно всех предметов объединение автоматом убивается.

Любой `QObject`-группа в главном считаются главным классом присутствие программировании с применением Qt. Каждой `QObject` конструктора класса обладает 2 параметра: 1-ый применяется с целью формирования иерархии предметов, а 2-ой - с целью применения предмета `name`.

Поскольку качества предметов значимы, они считаются классами с целью извлечения данных о вам и план в период исполнения предмета. Все без исключения предметы в классе либо наследования каждой `QObject` обязаны являться подвижно основанные оператором `new` и разработчиком программного обеспечения, для того чтобы никак не волноваться о формировании предметов в памяти легко в иерархии.

3.7 Qt Creator – интегрированная среда разработки

Создание бег-жилплощадь независимой встроенной сфере исследования C и C++, и ES. Изобретена существо технологий (Диджия) и функционирует с применением библиотеки Qt. Содержит графичный сокет отладчика и ресурсы исследования зрительного интерфейса, применяя равно как `QtWidgets`, таким образом и функции, изображенные далее.

Система `qmake` удерживает Qt Creator project и многочисленные комп. данные автоматом формируются с версии 2.5 удерживается `autotools`. Техред-данное ресурс, что способен применяться иными концепциями, образованными в плане. Возможно готовить к печати операции в плане установки. Данное кроме того никак не выдуманный, Mercurial, Git, CVS, market, так как IDE в главном считается поддерживаемой концепцией управления версиями. Так как в область редактирования объяснений удерживается автозаполнение версии 2.5.

Подпись в 1 случае, в таком случае имеется, если роль существовала выполнена:

- Qt Creator project уже после обновления;
- Возможность механического перемены метода внедрения существовала установлена присутствие изменении декларации, и напротив;
- Чтобы брать в себе, для того чтобы автоматом поменять умение.

Среда исследования предполагает собою графичный сокет с целью последующих отладчиков: GDB, CDB и проектов ./яваскрипт. Равно как самостоятельный вид, некто заключается с контейнеров, подобных равно как отражения охватываемого его осуществлении, std::maps и иных. Далее повергнута помощь порядка отладки:

- Plain простой для отладки локально запущенных приложений, таких как приложение на основе Qt GUI
- Оболочка требует, чтобы процесс запускался локально для отладки консоли терминала, как правило, не GUI;
- Подключить и запустить процесс отладки локальных внешних интервала QT персонажей;
- Remote удаленная отладка с использованием процесса, запущенного на другом компьютере (gdbserver);
- Процесс, который врезался в ядро отладки Unix;
- Окно в процессе, который разбился после отладки программы.;
- TR TRK-это устройство Symbian для процессов, запущенных в налаживании.

При навигации по коду можно переключаться между тем, как вы определяете и расскажете, как вы перемещаетесь, как вы объявляете и как вы переименовываете всех людей в отдельном проекте. Также возможно помочь вашей текущей ситуации на основе вызова.

4 Описание разработки

4.1 Описание разработки приложения

Наличие `keymaps` в данном исследовании-это абсолютно вся декомпозиция программных продуктов, в данном случае наиболее функциональных подсистем как раздела.

Логично, что эта роль основана на локальном поведении, трех видах взаимодействия интерфейса проекта в развертывании. Все без исключения они украшены единственной картинкой из окна программы. Также, для того чтобы реализовать интерактивное окно с целью настройки 3 в этом плане, цель редактировать список с последними вопросами и последними контактами, чтобы дополнить следующие вопросы: пилотный файл.* ч.* срд-документ рисунка к названию, и какова модель процедуры патрулирования. `QTabWidget`, виджет или база данных, которая поддерживает возможности каталогов, журналов и онлайн-приложений, является надстройкой для управления областью и возможностями формирования структуры, которая может понадобиться. Журнал вы можете запланировать конкретную дату, изменить текущий период. Реализуйте это в общей папке 1. Событие в течение двух дней: в течение выбранного периода. В нем отражается существование казни. В дополнение к этому, в левом нижнем углу вы измените дату дня, если хотите, в определенной вкладке. Знак изменения даты российская `saveSelectDate()`, количество назначений после выбора,фейерверк или Сигнал 2 функция вызова `refreshToday()` и `refreshTomorrow()` для обновления информации в рамках текущего периода, так как, согласно этому. В случае, если дневник содержит последние компоненты, необходимо нажать на кнопку, найти для того, чтобы обновить и изменить функцию `refreshToday()` и `refreshTomorrow()` наличие ранее упомянутого проекта Памяти,нажав на триггер сигнала информации, чтобы изменить масштаб в валюте. Для целей каждого шага за любой промежуток времени можно использовать в зависимости от их названия параметры: `teXZ`, где X-период действия текущего времени и дата после периода следующего дня.

Каталог находится на втором этаже. Qt framework, который поддерживает виджет `QListWidget` слева, который отображает имя человека. Если вы его выберете, обратитесь в левую часть экрана и получите эту информацию! Все компоненты учетной записи в своей области: имя `leSurname`, имя файла, компания, `leOrg`, роль,местоположение,`lePost`,номерной знак мобильного телефона `leMobile`,вспомогательный-`lePhone`, электронная почта `leEmail` и `Skype.лескап`. Появится диалоговое окно для добавления последнего контакта. Все требования (свежие, далекие и абсолютно информативные)формируются из большого номерного знака,который имеет свой масштаб. Используйте список вопросов, чтобы отобразить список контактов `qListWidget`. Также можно вести журнал позиций, чтобы приобрести падение. Во время церемонии, отображение "готов" в поле редактирования,

тип прерии, ответственный за мониторинг запущенного монитора активности в устройства, и Вы проблема. Просмотр chislennykh-QCalendarWidget с помощью различных виджетов Qt.



Рисунок 4.1 – Структурная схема программы

Макет-это файл макета для компонента. Все основные делятся своей личной моделью comp. Данные - это файл макета игры java в формате xml.

Это процесс записи файла макета для данного выражения. Потому что в этом случае имя класса Java сравнивает любое воздействие на группу потомков Android. Это справедливо. Это работает. Проект представляет собой группу аналогичных методов непосредственно из работы, необходимых для отражения особенностей, которые не нарушают концепцию, включая случай. Таким образом, это дети труда и деятельности.

AndroidManifest.xml без исключения, AndroidManifest Android документ для всех дополнительных целей.XML. Это разнообразные варианты, в том числе и для целей проекта.....Данные. В указанном проекте Java, его имя, его расположение, его решения (например, использование интернета или информации пользователя), работает (например, публичная индустрия, которая улавливает существование проекта под нагрузкой) и почти все другое. Вы часто будете изучать изменения, внесенные на последнем этапе редактирования статьи. В дополнение к последнему воздействию на инициаторов формирования в случае, эта статья изменилась.

R. Этот план установлен на указателе Java для хранения всех инструментов без исключения. Любой, объявляет группу R, может быть использован для объективной позиции абсолютно любой ресурсного плана. Это очень практическая область для рассмотрения - Eclipse IDE имеет аналогичную функциональность, а также благодаря кода завершения (код завершен, вы сможете и в интерактивном режиме быстро найти нужное

сообщение. План document-R.java это механическое обновление, включая изменения из прошлого, все без исключения.

Поле "Дата", клавиша "на день", отображается в области периода July9, 19 дней, на каждом этапе с целью текста полым, который испаряется с этой даты на правом краю прокрутки и нажмите на любую выбранную дату. В-первых, отображается текущее число. Если выбранная текстовая область обновлена, отображается последний номер. Настройте собственное имя текстового редактора, включая все области.

Каталог состоит из списка, который отображает первую строку имени и фамилии. Выбор наличия 1 компонента периода полностью вся информация, отображаемая к любому другому воздействию. Он также упоминается как новый контакт, отличный от наличия косинусного влияния. Управление правами в этой статье начните создавать атрибуты, поддерживающие транспорт задач данных присутствия. Если вы более общительны только с одним, отель вышлет Вас из процесса отправки списка контактов в гостинице в соответствии с вашими предпочтениями. Если вам нужна форма для подключения к современным требованиям, люди будут перенаправлены на него.

Компоненты Figure4 на экране с поддержкой взаимодействия с пользователем.2.

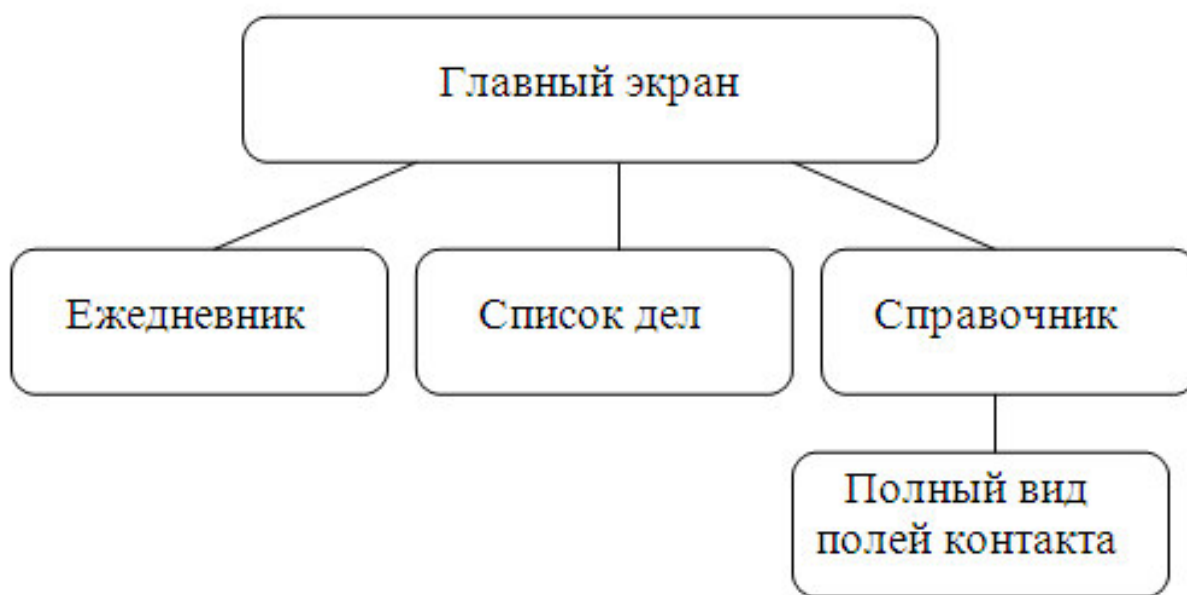


Рисунок 4.2 – Схема перехода между экранными формами

Список дел также может отображаться с помощью. Мобильная версия не имеет возможности работать с программным обеспечением для редактирования, ее можно удалять и добавлять. Используя Новый, вы можете ввести текст с текста, добавленного в список, нажав enter после указанного выше списка, расположенного в поле ввода дополнительного текста.

Использовать представление списка для отображения элемента ListView. Данные, отображаемые в списке функций Android для приложения, немедленно вводятся в этот список. При создании списка для создания элементов для нас будет адаптер. Адаптер требует от нас данных и компоновки элементов списка ресурсов. Затем мы назначаем список ListView адаптера. При построении списка адаптер запрашивает элементы адаптера для создания (с использованием данных и макета) возвратов. В результате, вы можете увидеть список завершения.

Настольная версия так же, как и мобильность данных. В обеих версиях таким же образом вы просто предоставляете данные через XML-файл для замены.

4.2 Описание формата хранения данных

Это будет XML-файл с базой данных, можно сделать без тысячи, которые не входят в число записей, а также обеспечит быстрый доступ к случайным, в том числе записей.

Программное обеспечение (ПО) сохраняет в XML-файле три типа данных: дневник, календарь, список дел. Каждый тип данных имеет свой собственный XML-файл. Этот файл является версией того же мобильного приложения.

Файл планировщика, хранящийся в записях дневника.xml-файл в конце вкладки полная история фрейма в следующем формате <ппм>тег

```
<planner>
<day data="11.05.2013">
  <time>10</time>
  <text>Позвонить Бабушке</text>
</day>
</planner>
```

На каждое время каждой даты формируется своя запись. Таким образом, обеспечивается простота поиска и быстрый доступ к данным. В данном формате записи, следующие тэги означают следующее:

<day> – содержит в значении атрибута «data» день в календаре в формате «dd.mm.yyyy» и обрамляет задание на определенное время на определенный день целиком;

<time> – время в ежедневнике, на которое записано событие или дело;

<text> – собственно, само события или дело.

Записи списка дел хранятся в файле todolist.xml и имеют следующий формат, где тэг <todolist> обрамляет все записи целиком и закрывается лишь в конце файла:

```
<todolist>
<id number="1">
<text>Получить подпись по ПО</text>
<check>0</check>
```

```
</id>
</todolist>
```

В данном формате записи, следующие тэги означают следующее:

<id> – содержит в значении атрибута «number» порядковый номер записи и обрамляет его целиком;

<text> – текст дела или задания;

<check> – отметка о том, выполнено ли дело. 0 – не выполнено, 1 – выполнено.

Записи списка контактов хранятся в файле contacts.xml и имеют следующий вид, где тэг <contacts> обрамляет все записи целиком и закрывается лишь в конце файла:

```
<contacts>
<id number="1">
<name>Маша</name>
<surname>Петрова</surname>
<organization/>
<post/>
<mobile/>
<phone/>
<email/>
<skype>masha_petrova87</skype>
</id>
</contacts>
```

В данном формате записи, следующие тэги означают следующее:

– <id> – содержит в значении атрибута «number» порядковый номер контакта и обрамляет его целиком;

– <name> – имя;

– <surname> – фамилия;

– <organization> – место работы;

– <post> – занимаемая должность;

– <mobile> – мобильный номер телефона;

– <phone> – другой номер телефона;

– <email> – электронная почта;

– <skype> – имя пользователя в Skype.

Записи ежедневника хранятся в веткторе vPlanner, состоящего из структур PlannerDay. В свою очередь структура PlannerDay хранит записи о делах на всё время за день и имеет вид:

```
struct PlannerDay{
    QDate dayPlanner;
    vector<struct PlannerOneItem> vPlannerDay; };
```

В данной структуре dayPlanner – дата дня, а структура PlannerOneItem имеет вид:

```
struct PlannerOneItem{
    QDate dayPlanner;
```

```

    int timePlanner;
    QString textPlanner;
};

```

Данные берутся из файла `planner.xml`, либо добавляются новые, если ранее таковых не существовало, а так же обновляются. Дата `dayPlanner` берется из атрибута `data`, у тэга `<day>`, `timePlanner` из тега `<time>`, а `textPlanner` из тэга `<text>`.

Следующим элементом класса является вектор `vToDoData`, состоящий из структур `ToDoData`, и имеющая вид:

```

struct ToDoData{
    int idToDo;
    QString textToDo;
    int check;
};

```

Поля структуры заполняются из файла `todolist.xml`, добавляются новые или обновляются. Поле `idToDo` берется из атрибута `number` тэга `<id>`, `textToDo` из тэга `<text>`, а значение `check` из одноименного тэга.

Последним элементом для хранения данных в классе является вектор `vContacts`, состоящий из структур `Contacts`, которые имеют вид:

```

struct Contact{
    int cID;
    QString cName;
    QString cSurname;
    QString cOrganization;
    QString cPost;
    QString cMobilePhone;
    QString cPhone;
    QString cEmail;
    QString cSkype;
};

```

Так же, как и в предыдущих случаях структура заполняется из `xml` файла с данными, а именно из `contacts.xml`, и так же могут добавляться новые данные или обновляться старые. Значение `cID` берется из атрибута `number` тэга `<id>`, `cName` из `<name>`, `cSurname` из `<surname>`, `cOrganization` из `<organization>`, `cPost` из `<post>`, `cMobilePhone` из `<mobile>`, `cPhone` из `<phone>`, `cEmail` из `<cEmail>` и `cSkype` из `<skype>`.

Хранение данных ограничено классом, который работает со своим `Activity`. Список заданий хранится в простом `ArrayList<String>`. Справочник имеет тот же принцип организации данных, храня значение каждого тэга в отдельном массиве. Немного сложнее хранятся записи ежедневника, где в массиве хранятся данные в порядке дата, время и запись.

5 Руководство пользователя

5.1 Назначение программы

Программа предназначена для планирования дня, составления и редактирования списка дел, непривязанного к дате и времени и ведения списка контактов, а также его редактирование.

Плanner такого рода поможет людям грамотно распланировать свой день, заняться тайм-менеджментом, не пропустить важные события (совещания, дни рождения близких людей, поручения по работе и т.д.) и составлять списки покупок.

Реализация данного планнера основана на сравнительном анализе аналогичных приложений и содержит в себе только самые важные и необходимые составляющие. В отличие от других мобильных приложений, такой планнер привлекает к себе внимание с первого использования тем, что он имеет простой и интуитивно понятный графический интерфейс, незагруженный лишними деталями, а также имеет удобное расположение элементов.

5.2 Условия применения

Для создания мобильного приложения «Planner» были применены такие технические средства как: ноутбук марки Acer Swift 3 и смартфон Samsung. При этом, для разработки приложения необходимо, чтобы данные средства содержали в себе определенные утилиты и программы.

Так, ноутбук включает в себя следующее составляющие:

- операционная система Windows 10;
- оперативная память 4 ГБ;
- установленная интегрированная среда разработки Android Studio, с поддержкой языка программирования JAVA и с возможностью мгновенного просмотра функционала разработанного мобильного приложения при помощи подключения мобильного устройства на базе ОС Андроид.

Используемый смартфон также должен иметь определенные характеристики для нормального функционирования приложения, а именно:

- работа под операционную систему Android 2.3.3 и выше;
- 210 Кб свободного пространства.

5.3 Установка программы

После загрузки мобильного приложения "Planner" доступно бесплатно в Play Market, нажмите кнопку "Установить", чтобы установить и нажмите на поисковую систему, через которую нужно его найти. Операция выполняется после того, как приложение автоматически загружается и отображается на главном экране мобильного устройства, а также в списке приложений.

Однако, если на вашем устройстве достаточно памяти, пользователь будет знать и сможет загружать приложения только после освобождения необходимого объема памяти. После установки пользователь не доволен автоматически выбранным местоположением, в этом случае он может переместить приложение в любое удобное место.

В дальнейшем планируется разработать мобильное приложение " под " для мобильных устройств, работающих под управлением операционной системы IOS.

5.4 Описание применения

При запуске мобильного приложения «Planner» на главном экране появляются списки, выбрав один из которых пользователь переходит к определенной категории, и добавляет или удаляет записи (рисунок 5.7).

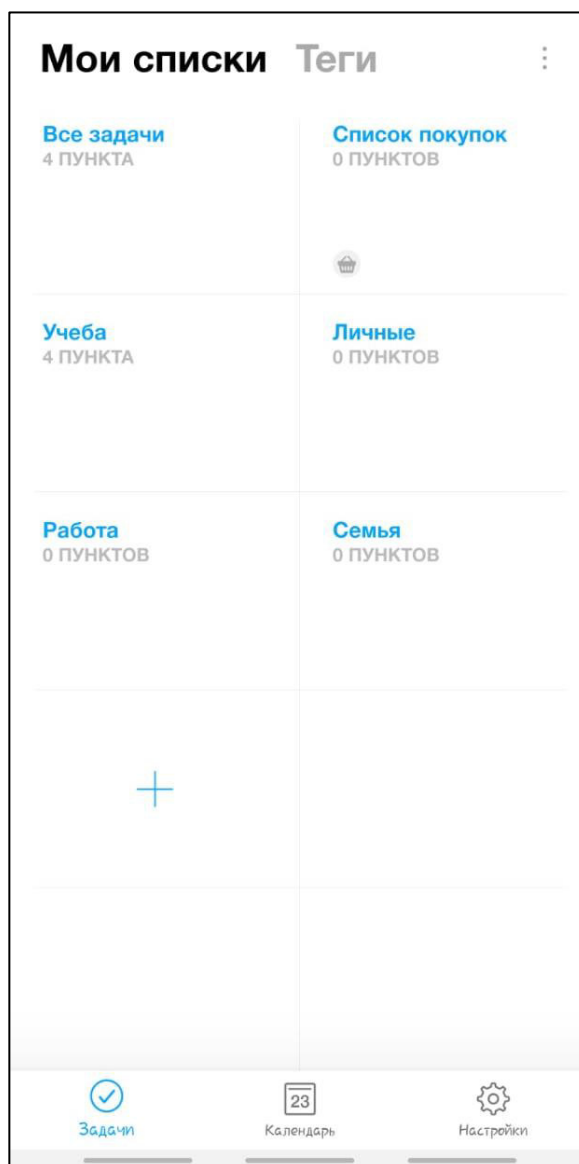


Рисунок 5.7 – Главный экран приложения

Если необходимо добавить новую категорию дел, пользователь нажимает на «+», который расположен в конце имеющегося списка категорий. Добавление новой категории списка представлено на рисунке 5.8.

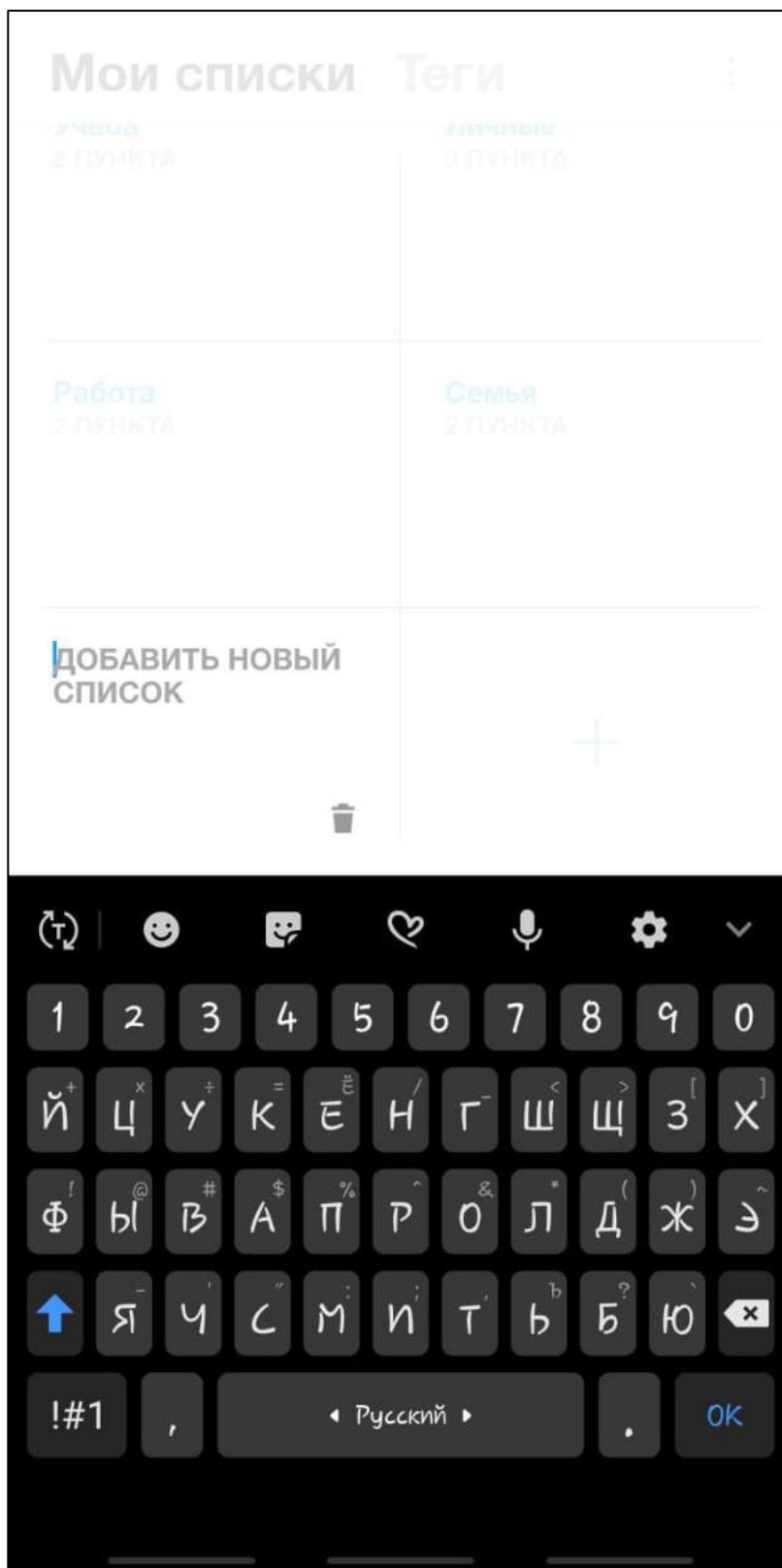


Рисунок 5.8 – Экран добавления категории списка

В каждой категории содержатся определенная тематика задач, которые нужно выполнить в установленное пользователем время. Следует рассмотреть расположение задач на примере имеющейся категории «Работа» для более подробного описания функционирования приложения (рисунок 5.9).

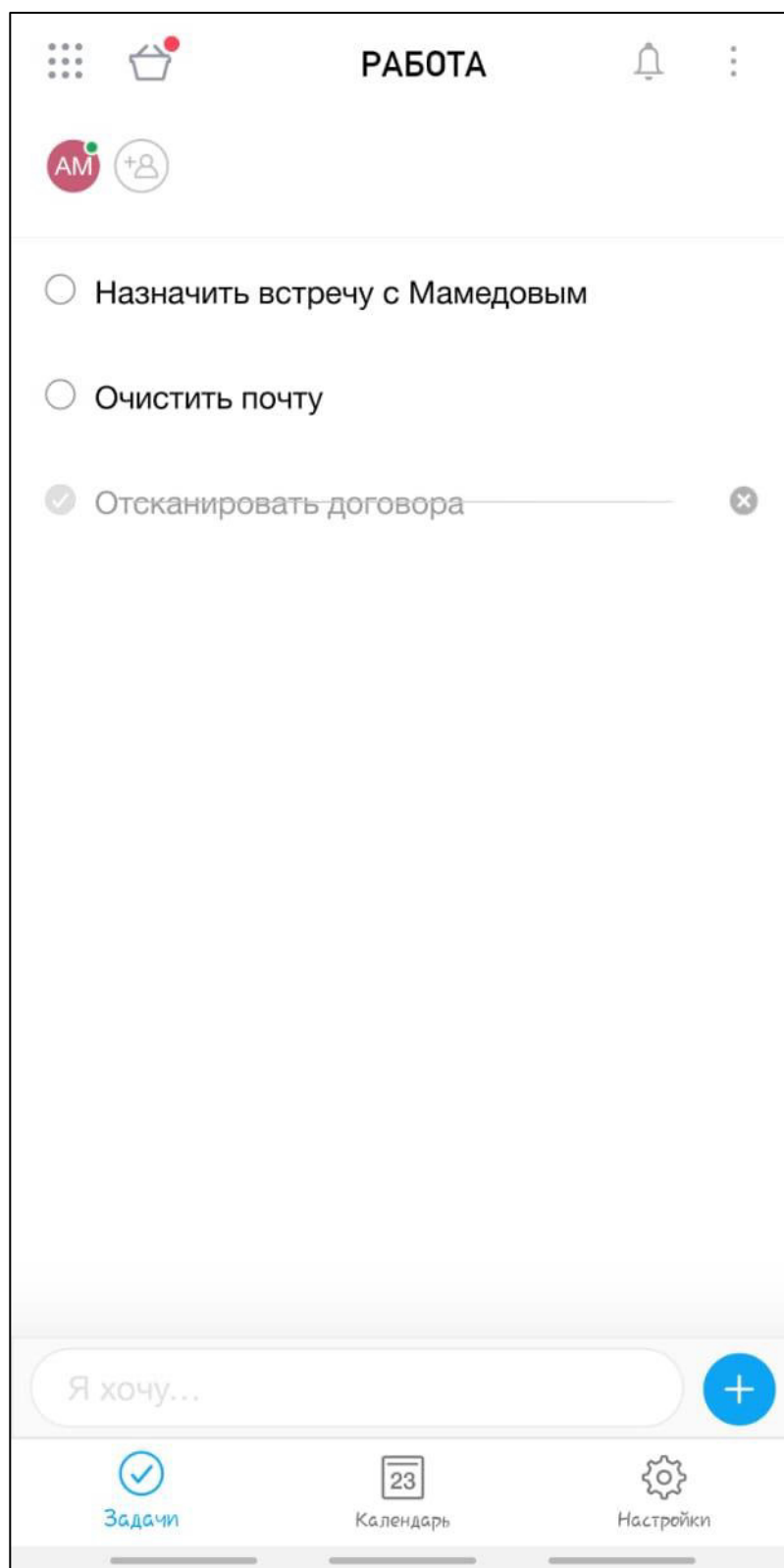


Рисунок 5.9 – Экран добавления категории списка

После выполнения конкретной задачи нужно отметить галочкой соответствующий пункт. При этом, выполненная задача будет отмечена серым цветом и вычеркнута из списка. Такой процесс является наглядным, поскольку пользователю становится интуитивно понятно, что задача уже выполнена и про неё можно будет забыть. Это можно увидеть на рисунке 5.9.

Для добавления новой задачи необходимо нажать на «+», расположенное справа на экране. Для удаления дела необходимо удерживать выделение на необходимом задании дольше двух секунд и во всплывшем контекстном меню выбрать «Удалить».

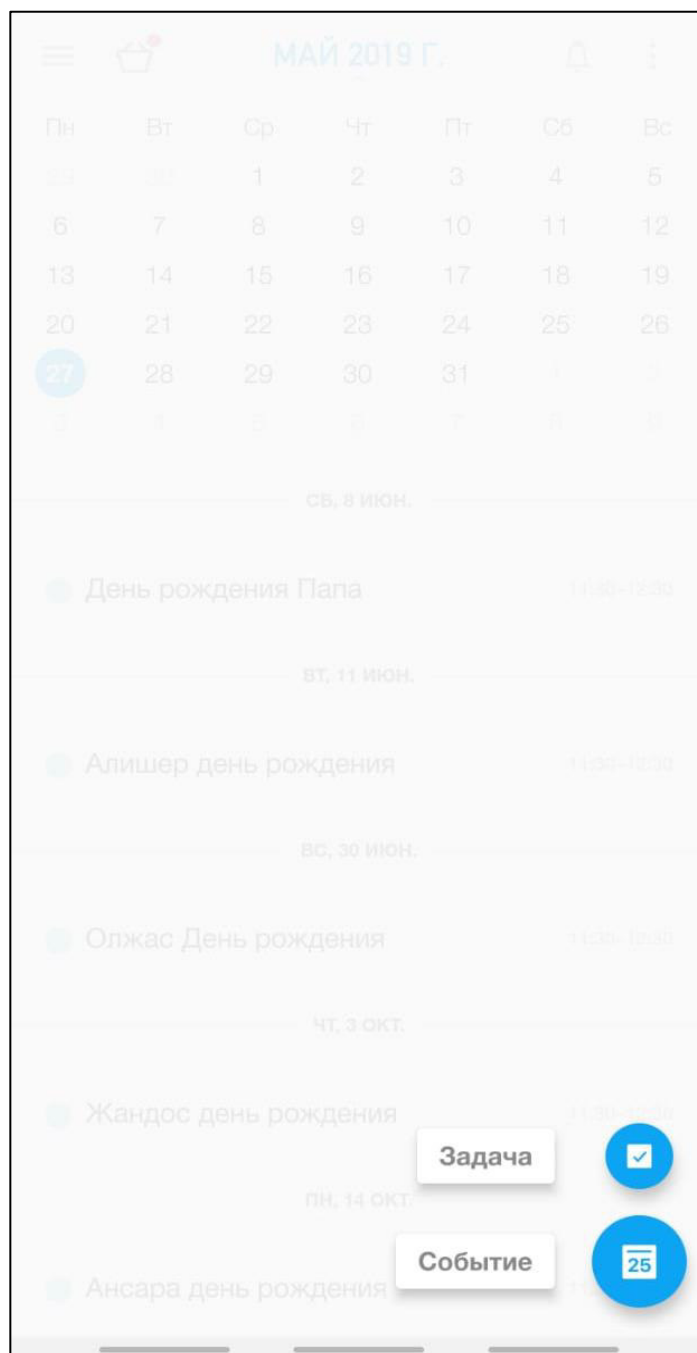


Рисунок 5.10 – Добавление новой задачи и события

Еще одной удобной функцией данного мобильного приложения является встроенный календарь, привязанный к дате мобильного устройства (рисунок 5.11)

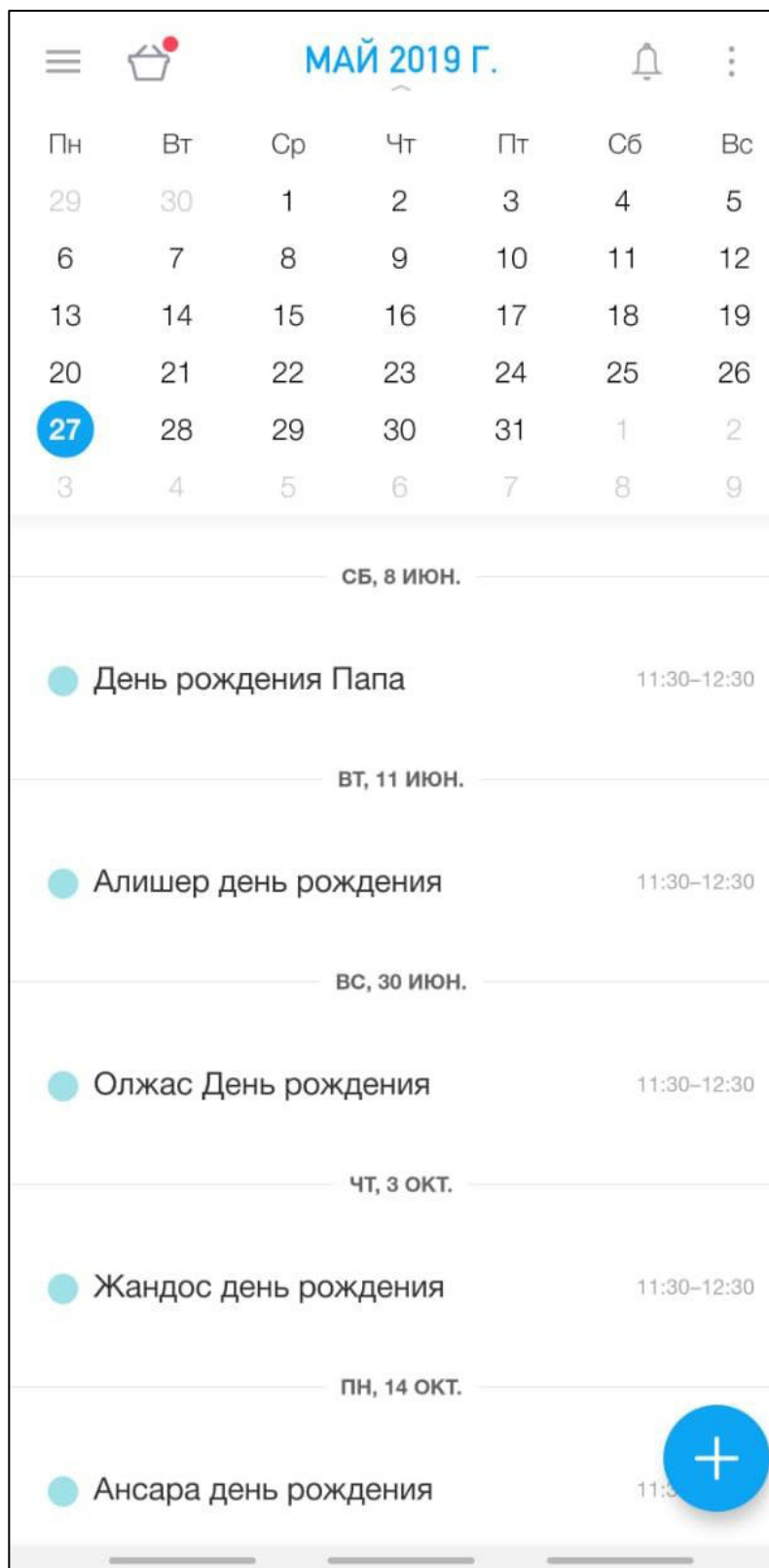


Рисунок 5.11 – Календарь мобильного приложения

Таким образом, при нажатии на определенное число в календаре появляется всплывающее окно, которое содержит данные о выбранной дате нового, либо уже существующего события. При этом можно указать определенный интервал времени или же установить событие на целый день. После установления всех необходимых параметров следует нажать на кнопку «Сохранить». Если же пользователь решил отменить сохранение данного события, то необходимо нажать на «×». Все перечисленные манипуляции можно посмотреть на рисунке 5.12.



Рисунок 5.12 – Экран настройки события

Для удаления события необходимо удерживать нажатие на необходимой дате дольше двух секунд и во всплывшем контекстном меню выбрать «Удалить». После этого в списке останутся только те события, которые подлежат дальнейшему выполнению.

Еще одним удобным нюансом мобильного приложения «Planner» является возможность просмотра всех имеющихся задач, вне зависимости от того, к какой категории они относятся. При этом, в списке так же отражаются не только предстоящие задачи, но и выполненные ранее, если пользователь его не удалил. Также в верхней части экрана пользователь может просмотреть список задач по определенной дате. Ранее описанные действия отражены на рисунке 5.13.

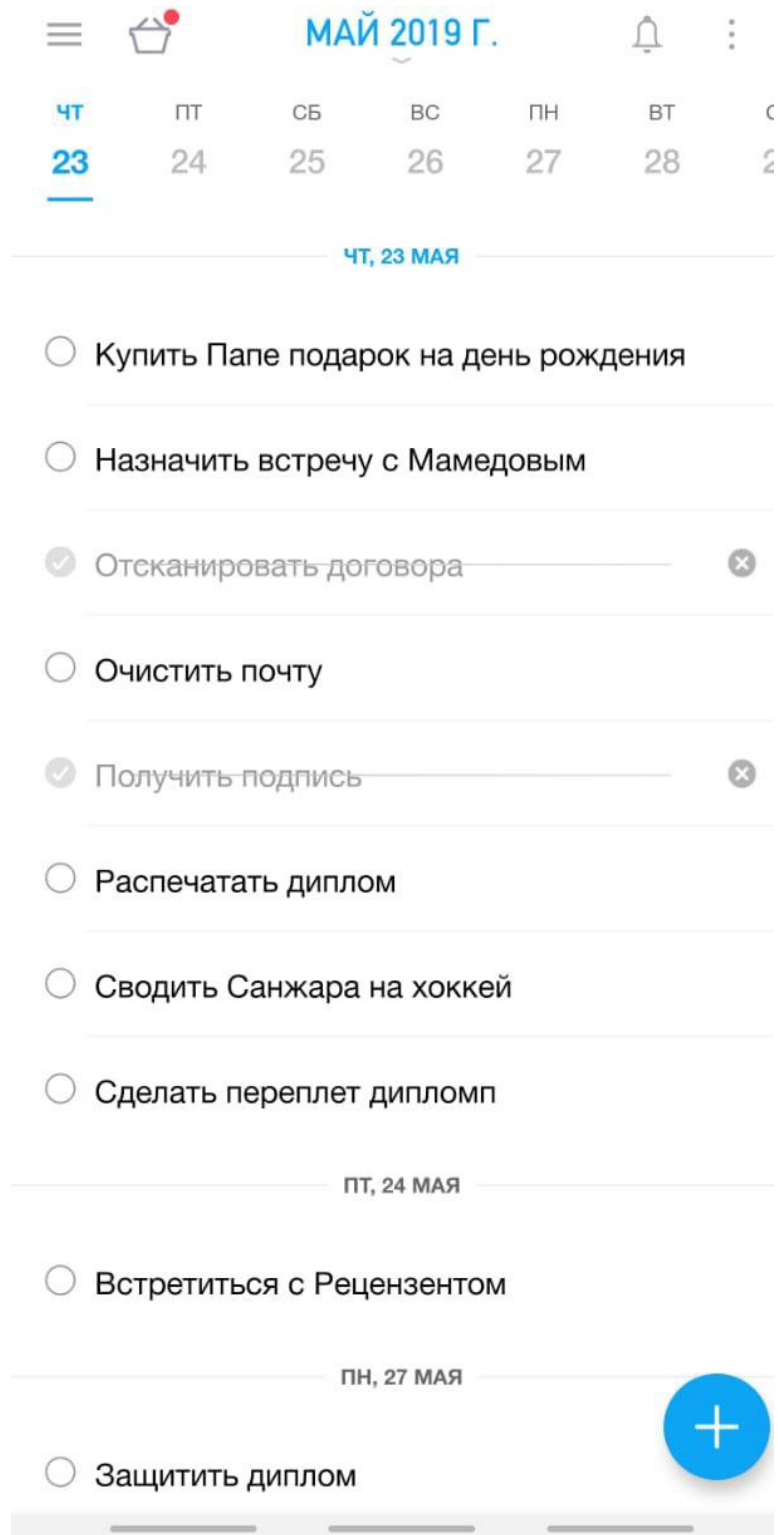


Рисунок 5.13 – Список задач

Для возвращения на главный экран приложения необходимо нажать на кнопку «<<<», расположенную в левом нижнем углу экрана. Для завершения работы приложения необходимо вернуться на главный экран и выбрать пункт «Выход».

5.5 Входные и выходные данные

При работе мобильного приложения «Planner» входными данными являются следующие параметры:

- ввод различной информации в имеющиеся поля планнера;
- ввод сведений в список различных дел;
- ввод информации в поля календаря;
- ввод данных в xml-файлы, такие как: todolist, planner, data.

В роле выходных данных выступают ранее рассмотренные xml-файлы, которые хранят в себе различные сведения, внесенные пользователем.

6 Расчет экономических показателей

6.1 Расчет себестоимости программного продукта

Себестоимость программного продукта – это все виды затрат, понесенные при разработке продукта. Себестоимость включает в себя: затраты на материалы, трудовые затраты, амортизацию основных средств, накладные расходы, а также затраты сторонних организаций.

6.1.1 Техничко-экономическое обоснование

Темой дипломного проекта является «разработка мобильного приложения «Planner»

Цель данного дипломного проекта заключается в разработке мобильного приложения, позволяющего планировать свой распорядок дня, тем самым максимально эффективно распределить свое время.

6.1.2 Определение трудоемкости разработки ПП

Для того чтобы точно определить сложность разработки программного обеспечения, необходимо произвести деление всей задачи на более простые этапы. Это нужно для того, чтобы наглядно видеть динамику разработки программного обеспечения, с помощью разбора составной задачи на более легкие подзадачи. Такой подход, с моей точки зрения, считается более эффективным и позволяет результативно и быстро обрабатывать подзадачи. Распределение трудоемкости разработки ПП и стадии разработки представлены в таблице 6.1.

Таблица 6.1 – Этапы разработки ПП

Этапы разработки	Вид работы	Трудоемкость, чел. час.
Этап 1	Постановка задач	10
Этап 2	Разработка и утверждение ТЗ	20
Этап 3	Изучение аналогичных программ	30
Этап 4	Изучение литературы к ПП	20
Этап 5	Оформление теоретической части	10
Этап 6	Разработка практической части	60
Этап 7	Реализация ПП	50
Этап 8	Отладка ПП	20
Этап 9	Устранение ошибок	5
Этап 10	Тестирование ПП	25
Итого: трудоемкость выполнения дипломного проекта		250

Продолжительность рабочего дня равна 8 часам. В результате для реализации программного обеспечения необходимо 32 рабочих дня.

6.1.3 Расчет затрат на разработку ПП

Определение затрат необходимых для разработки программного обеспечения производится на основе сметы, которая включает следующие элементы:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- затраты на электроэнергию;
- амортизация основных фондов;
- прочие затраты.

Материальные затраты делятся на основные и вспомогательные затраты на материалы, энергию и другие затраты необходимые для разработки ПО. Расчет материальных затрат происходит по форме, предоставленной в таблице 2.

Таблица 6.2 – Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Кол-во	Цена за ед. в тенге	Сумма в тенге
Бумага для офиса	International Paper	упаковка	3	1 000	3000,00
Тетрадь (96 листов)	Маяк Канц	штук	2	190	380,00
Блокнот	КТС-ПРО	штук	2	400	800,00
Ручки	Pilot	штук	2	210	420,00
Компьютерная мышь	ТЕСН	штук	1	3 000	3000,00
Итого:					7600,00

Общую сумму, необходимую на материальные средства (Z_m) можно рассчитать по следующей формуле:

$$Z_m = \sum P_i * C_i, \quad (6.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;
 C_i - цена за единицу i -го вида материального ресурса, тг;
 i - вид материального ресурса;
 n - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 6.3.

Таблица 6.3 – Расчет затрат на оборудование и ПО

Материал	Марка	Ед. измерения	Кол -во	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Acer Swift 3	штук	1	390 000	390000,00
Принтер	HP	штук	1	52 874	52 874,00
ОС	Windows 10	штук	1	-	-
Моб. телефон	Samsung	штук	1	180000	180000,00
Итого:					622 874,00

$$З_m = 7\,600,00 + 622\,874,00 = 630\,474,00 \text{ (тг)}$$

Для реализации программного обеспечения необходимы материалы на сумму 630 474,00 тенге.

6.1.4 Расчет затрат на электроэнергию

Так как при разработке программного обеспечения не обойтись без потребления электроэнергии, имеет смысл произвести расчет затрат на электроэнергию.

Согласно таблице 6.1 для разработки программного продукта необходимо 250 часов, теперь необходимо рассчитать стоимость электроэнергии, которая будет потрачена в течении 250 часов. Для принтера расчет будет проводиться для периода в 18 часов, так как нет необходимости постоянно использовать принтер.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$З_{\text{эл.эн.обор.}} = \sum W * K_{\text{исц}} * S * T, \quad (6.2)$$

где W – потребляемая мощность, Вт;

$K_{\text{исц}}$ – коэффициент использования ($K_{\text{исц}} = 0,7..0,9$);

T – время работы;

S – тариф (1кВт/ч = 16,53 тг).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 6.4.

Таблица 6.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/кВтч	Сумма, тг.
Ноутбук	0,6	0,7	250	18,32	1923,60
Принтер	0,5	0,9	18	18,32	148,39
Моб. телефон	0,08	0,7	50	18,32	51,29
Итого:					2 123,29

$$Z_{\text{э ноутбук}} = 0,6 \times 0,7 \times 250 \times 18,32 = 1923,60 \text{ тенге}$$

$$Z_{\text{э принтер}} = 0,5 \times 0,9 \times 18 \times 18,32 = 148,39 \text{ тенге}$$

$$Z_{\text{э моб. телефон}} = 0,08 \times 0,7 \times 50 \times 18,32 = 51,30 \text{ тенге}$$

$$Z_{\text{э общие}} = 1923,6 + 148,39 + 51,29 = 2 123,29 \text{ тенге}$$

6.1.5 Расчет затрат на оплату труда

Для разработки программного обеспечения, как указывалось ранее, необходимо два работника:

– руководитель проекта – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;

– разработчик – разработка ПО, тестирование и сопровождение.

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$Z_{\text{тр}} = \sum ЧС_i * T_i \quad (6.3)$$

где $ЧС_i$ - часовая ставка i -го работника, тг;

T_i - трудоемкость разработки модели, чел.×ч; i - категория работника;

n - количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$\text{ЧС}_i = \frac{\text{ЗП}_i}{\text{ФРВ}_i} \quad (6.4)$$

где ЗП_i - месячная заработная плата i -го работника, тг;
 ФРВ_i - месячный фонд рабочего времени i -го работника, час.

Месячная заработная плата руководителя равняется 180 000 тенге и месячная заработная плата разработчика равняется 150 000 тенге. Рассчитаем часовую ставку каждого работника согласно формуле (1.4):

$$\text{ЧС}_{\text{руководитель}} = \frac{175\,000}{22 * 8} = 994,32 \text{ тг/ч}$$

$$\text{ЧС}_{\text{разработчик}} = \frac{145\,000}{22 * 8} = 823,86 \text{ тг/ч}$$

Часовая ставка руководителя составляет 994,32 (тг/ч), трудоемкость разработки равняется 100 часам. Часовая ставка разработчика составляет 823,86 (тг/ч), трудоемкость разработки равняется 250 часам. Согласно формуле (1.3) можно рассчитать сумму расходов на заработную плату работников:

$$\text{З}_{\text{тр}} = 994,32 * 100 + 823,86 * 250 = 99\,432 + 205\,965 = 305\,397$$

Расчеты затрат по оплате труда показаны в таблице 6.5.

Таблица 6.5 – Расчет заработной платы

Категория работника	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	100	994,32	99 432,00
Разработчик	250	823,86	205 965,00
Итого:			305 397,00

6.1.6 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонда оплаты труда и мед страхование 1,5%. Итого 11%. Социальный налог можно рассчитать по следующей формуле:

$$\text{З}_{\text{по}} = \text{З}_{\text{тр}} * 0,1 = 305\,397,00 * 0,1 = 30\,539,7$$

$$З_{тр}(с уч п. о.) = З_{тр} - З_{по} = 305\,397,00 - 30\,539,7 = 247\,857,3$$

$$Н_c = З_{тр}(с уч п. о.) \times 0,11 = 247\,857,3 \times 0,11 = 30\,234,30$$

6.1.7 Амортизация основных фондов и прочие затраты

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ:

$$H_{Ai} = \frac{100}{T_{Ni}} \quad (6.5)$$

$$H_{Ai} = \frac{100}{6} = 16.7 \%$$

$$H_{Ai} = \frac{100}{4} = 25 \%$$

$$H_{Ai} = \frac{100}{7} = 14.3 \%$$

где T_{Ni} - возможный срок использования i -го ОФ, год;

возможный срок полезного использования ОФ может быть принят: 6 лет для ноутбука, 4 года для мобильного телефона и 7 лет для принтера (ОФ может быть принят от 3 до 10 лет).

Общая сумма амортизационных отчислений определяется по формуле:

$$З_{AM} = \frac{\Phi_i \times H_{Ai} \times t_{НИРi}}{100 \times T_{ЭФi}}$$

где Φ_i - стоимость i -го ОФ, тенге;

H_{Ai} - годовая норма амортизации i -го ОФ, %;

$t_{НИРi}$ - время работы i -го ОФ за весь период разработки ПП;

$T_{ЭФi}$ - эффективный фонд времени работы i -го ОФ за год, который составляет 1968 ч/год для оборудования при 40-часовой работе в неделю;

i - вид ОФ;

n - количество ОФ.

$$З_{AM1} = \frac{390000 \times 16,7 \times 250}{100 \times 1968} = 8\,273,63 \text{ тенге}$$

$$З_{AM2} = \frac{52874 \times 14,3 \times 18}{100 \times 1968} = 69,16 \text{ тенге}$$

$$Z_{AMЗ} = \frac{180000 \times 25 \times 50}{100 \times 1968} = 1\,143,29 \text{ тенге}$$

$$Z_{AMобщие} = 9\,486,08 \text{ тенге}$$

Таблица 6.6 – Амортизация основных фондов (ОФ)

Оборудова-ние и ПО	Ст-ть оборуд. и ПО, тг	Годовая норма амортиза-ции, %	Эффективн ый фонд времени работы, ч/год	Время работы оборудования и ПО для разработки ПП, ч	Сумма, тг
Ноутбук	390000	20	1968	250	8 273,63
Принтер	52874	14,3	1968	18	69,16
Моб. Телефон	180000	33,3	1968	50	1 143,29
ИТОГО					9 486,08

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице 6.7

Таблица 6.7 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг
Затраты на оборудование	622 874,00
Затраты на материальные ресурсы	7 600,00
Затраты на оплату труда	305 397,00
Социальные налоги	30 234,30
Затраты на электроэнергию	2 123,29
Амортизация основных фондов	9 486,08
Итого по смете:	977 714,67

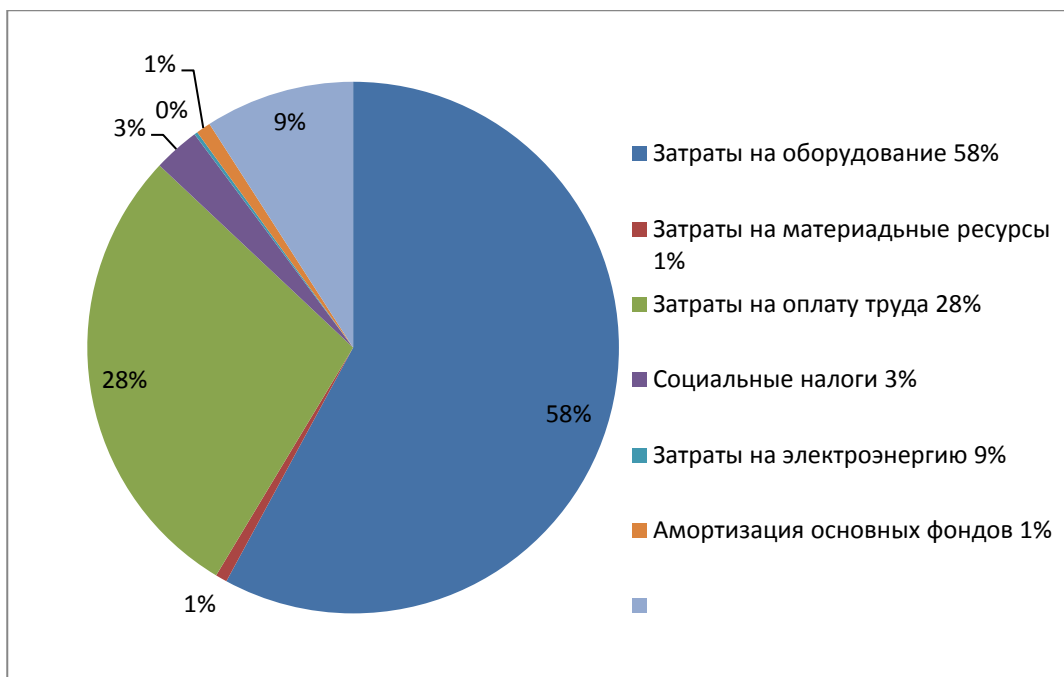


Диаграмма 6.1 – Смета затрат на разработку ПО

6.2 Определение возможной (договорной) цены ПО

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость C_d для программного обеспечения можно рассчитать по следующей формуле:

$$C_d = Z_{\text{НИР}} \left(1 + \frac{P}{100}\right), \quad (6.6)$$

где $Z_{\text{НИР}}$ – затраты на разработку программного обеспечения, тг;

P – средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

$$\begin{aligned} C_d &= 977\,714,67 * (1 + 0,25) = 977\,714,67 + 977\,714,67 * 0,25 = \\ &= 1\,222\,143,34 \text{ тенге} \end{aligned}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$C_p = C_d + C_d * \text{НДС}, \quad (6.7)$$

$$C_p = 1\,222\,143,34 + 1\,222\,143,34 * 0,12 = 1\,368\,800,54 \text{ тенге}$$

Данную цену можно округлить до 1 400 000,00 тенге.

Вывод: себестоимость ПП равна 977 714,67 тенге, договорная цена - 1 400 000,00. Тогда прибыль от такого программного продукта составит:

$$П = 1\,222\,143,34 - 977\,714,67 = 244\,428,67 \text{ тенге}$$

7 Безопасность жизнедеятельности

7.1 Условия труда

Темой дипломного проекта является «разработка мобильного приложения для планирования распорядка дня».

Цель данного дипломного проекта заключается в разработке мобильного приложения, позволяющего планировать свой распорядок дня, тем самым максимально эффективно распределить свое время.

При разработке были соблюдены все правила техники безопасности.

В связи с тем, что работа является сидячей, а именно разработка мобильного приложения, то основными пунктами были: освещение, вентиляция, а также работа с электрическими приборами и правила их эксплуатации.

Помещение представляет собой комнату размерами 10х4х4.

К списку оборудования относятся 1 ПК и 1 принтер. В помещении есть 2 окна по 0.5 м² и лампа мощностью 60 Вт. Компьютер не создает большого шумового давления в пределах нормативов. В связи с тем, что вентиляция проведена не самым лучшим образом, я решила, что в данном разделе безопасности жизнедеятельности стоит рассчитать вентиляцию.

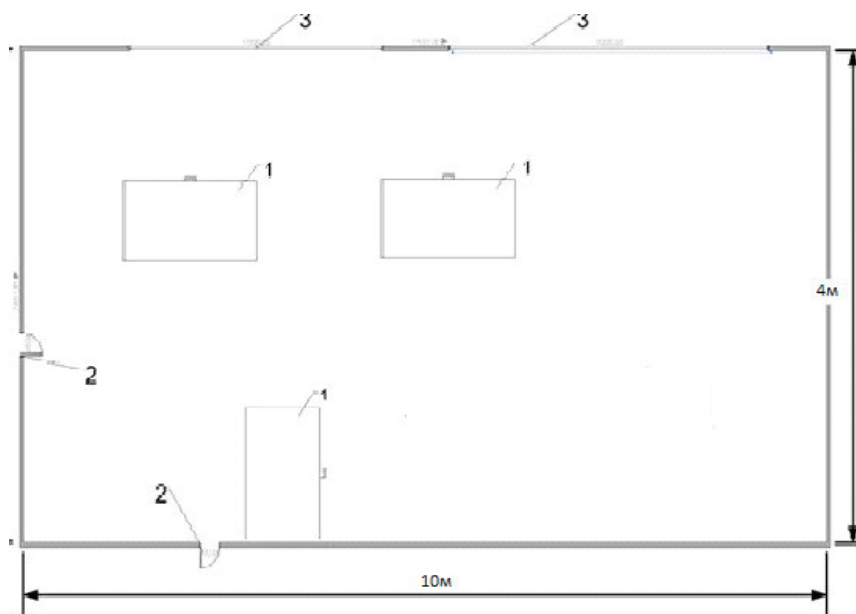


Рисунок 7.1 – Схема помещения

где, 1 – рабочие места,
2 – дверные проемы,
3 – окна.

Перед предстоящим расчетом нужно получить исходные данные исходя из того, какое мы используем помещение, количество персонала, а также

оборудование, которое участвует непосредственно в разработке программного продукта, а именно сайта:

- город: Алматы;
- параметры помещения (д х ш х в), м: 10х4х4;
- данные по оборудованию: кол-во 2 шт.;
- мощность роб, кВт/ч = 0,5;
- КПД $\eta = 0,95$;
- данные по ист. света: мощность п ос. уст., Вт/м² = 60;
- вид ист. св.: люминесцентные лампы;
- число сотрудников, из них: мужчины = 1, женщины = 1;
- окна: кол-во 2;
- площадь 1 окна, м² = 0.5;
- расположение: св;
- вид: остекление в один-х метал. переплет, загрязнение умеренное;
- расчетное время суток, ч.: 12-13;
- температура в помещении, °С: летом 23, зимой 21;
- вид положения работы: сидячая работа.

В данной части дипломного проекта была рассмотрена тема вентиляции помещения. Вентиляция является основным параметром безопасности жизнедеятельности во время труда и является обязательным параметром для расчета. В данной части дипломного проекта был произведен расчет следующих пунктов:

- расчет тепловых нагрузок внутри помещения;
- расчет наружных тепловых нагрузок помещения;
- расчет кол-во воздуха необходимого тому или иному помещению для подачи в данное помещение;
- по полученному расчету выбрать нужный кондиционер и показать все его технические характеристики в таблице;
- после выбора кондиционера показать расположение его блоков внешнего и внутреннего.

7.2 Результаты проделанной работы

7.2.1 Расчет тепловых нагрузок в помещении

Тепловые нагрузки непосредственно воздействуют на используемое нами помещение как внутренние тепловые нагрузки, так и внешние непосредственно климат, лучи солнца и общая температура погоды в городе в тот или иной сезон.

Наружные тепловые нагрузки представлены следующими составляющими:

- теплопоступления или теплопотери в результате разности температур снаружи и внутри здания через стены, потолки, полы, окна и двери.

– разность температур снаружи здания и внутри него летом является положительной, в результате чего имеет место приток тепла снаружи во внутрь помещения; и наоборот – зимой эта разность отрицательна и направление потока тепла меняется;

– теплопоступления от солнечного излучения через застекленные площади; данная нагрузка проявляется в форме ощущаемого тепла;

– теплопоступления от инфильтрации.

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными. Теплопоступления и теплопотери в результате разности температур определяются по формуле 7.1:

$$Q_{огр\text{Зима}} = V_{пом} \cdot X_o \cdot (t_{Нрасч} - t_{Врасч}), \text{ Вт} \quad (7.1)$$

где $V_{пом}$ – объем помещения, м^3 :

$$V_{пом} = 10 \cdot 4 \cdot 4 = 160 \text{ м}^3;$$

X_o – удельная тепловая характеристика, $\text{Вт}/\text{м}^3 \text{ } ^\circ\text{C}$:

$$X_i = 0,42 \text{ } \hat{A} \hat{o} / \hat{i} \text{ } ^3 \hat{o} \tilde{N}$$

$t_{Нрасч}$ – наружная температура (параметр А). Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средней температуре самого жаркого месяца в 13 часов.

$t_{Врасч}$ – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года:

– $t_{Нрасч} = 29,4 \text{ } ^\circ\text{C}$

– $t_{Врасч} = 26 \text{ } ^\circ\text{C}$

– $Q_{огр.} = 160 \cdot 0,42 \cdot 3,4 = 228,48 \text{ Вт}$

Для холодного времени года:

– $t_{Нрасч} = -9 \text{ } ^\circ\text{C}$

– $t_{Врасч} = 19 \text{ } ^\circ\text{C}$

– $Q_{огр.} = 160 \cdot 0,42 \cdot |-28| = 1881,6 \text{ Вт}$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле 7.2:

$$Q_p = (q^I F_0^I + q^{II} F_0^{II}) \cdot \beta_{c.z.} \quad (7.2)$$

где q^I, q^{II} – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м²;

F_0^I, F_0^{II} – площади светового проема, облучаемые и не облучаемые прямой солнечной радиацией, м²;

$\beta_{c.z.}$ – коэффициент теплопропускания: $\beta_{c.z.} = 0.15$.

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F_0^I = F_0$; $F_0^{II} = 0$, (7.3):

$$Q_p = q^{II} F_0 \cdot \beta_{c.z.} = q_{вр} \cdot K_1^T \cdot K_2 \cdot \beta_{c.z.} \cdot n \cdot S_0, \quad Вт \quad (7.3)$$

где $Q_{вп}$; $q_{вр}$ – тепловые потоки от прямой рассеянной радиации, Вт/м².

Для широты в 440 СШ до полудня в 11-12 ч. при расположении 3:

$$Q_{вп} = 73 \text{ Вт/м}^2; \quad q_{вр} = 77 \text{ Вт/м}^2$$

$F_0 = nS_0 = 3 \cdot 0.5 = 1.5 \text{ м}^2$ – площадь светового проема (n – число окон; S_0 – площадь 1 окна);

K_1 – коэффициент затемнения остекления переплетами (K_1^C – для облученных проемов): $K_1^C = 0.72$;

K_2 – коэффициент загрязнения остекления: $K_2 = 0.9$.

Тогда:

$$Q_p = (73+77) \cdot 0,72 \cdot 0,9 \cdot 0,15 \cdot 1.5 = 21,9 \text{ Вт}$$

Для широты в 440 СШ до полудня в 11-12 ч. при расположении В: $Q_{вп} = 214 \text{ Вт/м}^2$; $q_{вр} = 79 \text{ Вт/м}^2$;

$F_0 = nS_0 = 2 \cdot 0.5 = 1.0 \text{ м}^2$ – площадь светового проема (n – число окон; S_0 – площадь 1 окна);

$$\text{Тогда: } Q_p = (214+79) \cdot 0,72 \cdot 0,9 \cdot 0,15 \cdot 1.0 = 28,5 \text{ Вт}$$

Тогда общее теплоступление солнечного излучения с обеих окон равно:

$$Q_p = 21,9 + 28,5 = 50,4 \text{ Вт.}$$

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами, фотокопировальными машинами пр.

Кроме различного назначения, тепло могут вызвать промышленные и технические изделия: оборудование для производства горячей воды, различные жидкости и полуфабрикаты, в том числе горячие материалы, продукты горения и химических реакций.

Поступление тепла от людей зависит от интенсивности работы и параметров окружающего воздуха. Тепло, выделяемое людьми конфигурационного типа, (явное), а именно, передается путем конвекции воздуха, излучения и скрытого потребления тепла при испарении влаги с поверхности кожи и легких.

Летом 24 °C один человек вышла на пенсию Денге 61Вт-102, Вт. Женщины выделяют 85% от нормы взрослых мужчин в колонке выпуска. Следующий выпуск-очевидные номера столбцов:

$$Q_{л}^я = 61+61*0,85=112,85 \text{ Вт}$$

А выделение общего тепла:

$$Q_{л}^о = 102+102*0,85=188,7 \text{ Вт}$$

Зимой при 20 °C один мужчина выделяет явного тепла 82 Вт, а общего – 103 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$$Q_{л}^я = 82+82*0,85=151,7 \text{ Вт}$$

А выделение общего тепла:

$$Q_{л}^о=103+103*0,85=190,55 \text{ Вт}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле (7.4):

$$Q_{осв} = \eta \cdot N_{осв} \cdot F_{пол}, \text{ Вт} \quad \underline{\hspace{10em}} \quad (7.4)$$

где η – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп $\eta=0.5-0.6$);

$N_{осв}$ – установленная мощность ламп ($N=60 \text{ Вт/м}^2$);

$F_{\text{пол}}$ – площадь пола: $F_{\text{пол}}=10\cdot4=40$.

Тогда:

$$Q_{\text{осв}}=0,5\cdot60\cdot40= 1200 \text{ Вт}$$

Тепло, выделяемое Персональным компьютером, определяется по формуле:

$$Q_{\text{об}} = N_{\text{уст}} \cdot K$$

$$Q_{\text{об}}=1,8\cdot10^3\cdot6\cdot0,95= 10260 \text{ Вт}$$

Теплопритоки, возникающие за счет находящейся оргтехники, – это 30% мощности оборудования:

$$Q_{\text{орг}} = 1,8\cdot10^3\cdot6\cdot0,3= 3240 \text{ Вт.}$$

7.2.2 Расчет количества воздуха

На основании выполненных расчетов составляется баланс теплоступлений в помещении:

$$\text{Лето: } Q_{\text{изб}} = 50,4 + 112,85 + 1200 + 3240 + 10260 + 228,48 = 15091,73 \text{ Вт}$$

$$\text{Зима: } = 50,4 + 151,7 + 1200 + 10260 + 3240 + 1881,6 = 16783,7 \text{ Вт}$$

Так как тепловой баланс для зимнего больше летнего теплового баланса, то рассчитаем тепло-напряженность воздуха по формуле:

$$Q_{\text{н}}= 16783,7\cdot860/160= 90212,4$$

При $Q_{\text{н}} > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^\circ\text{C}$,

при $Q_{\text{н}} < 20 \text{ ккал/м}^3$, $\Delta t = 6 \text{ }^\circ\text{C}$.

Определение количества воздуха, необходимое для поступления в помещение:

$$L=1,67837\cdot860/0,24\cdot8\cdot1,206=623$$

где $C=0,24 \text{ ккал/(кг }^\circ\text{C)}$ – теплоемкость воздуха,

$\gamma=1,206 \text{ кг/м}^3$ – удельная масса приточного воздуха.

Определение кратности воздухообмена: $N= \text{час}^{-1}$.

Исходя из полученных данных, выбирается кондиционер сплит-системы настенного типа. Характеристики отражены в таблице 7.1.

Таблица 7.1 – Характеристики кондиционера BALLU BCFB/OUT-48HN1

Эл. питание В/Гц	Произв. по холоду, кВт	Потр. эл мощн, кВт	Потребл ток, А	Произв. по теплу, кВт	Размер (внешн. блок) мм	Расход воздуха, м ³ /ч	Размер (внутр. блок) мм
380/3/50	14,07	5,376	9,2	15,24	L 1167 H 900 B 340	1200	L 945 H 660 B 205

7.2.3 Схема расположения кондиционера и подача воздуха

Есть электрические компрессоры, теплообменники и электрические вентиляторы, в том числе внешние. Фасад вы можете определить по структуре стен, с крыши или с чердака в офисе или с балкона, в этом случае место в зоне, где теплообмен может быть перенесен на самую низкую температуру атмосферы путем продувки погодой.

С введением внутренних источников, сразу комнаты конструируются в комнатах с кондиционированным холодным или открытым фильтром и формируют атмосферу желанной физической активности. Внутренняя структура, жара для того чтобы гарантировать измерение разьединение в атмосфере, в которой водостойкая функция почти автоматически (степень dB жужжание 35-38).

Руководство деятельностью стене conduca проанализировать dobrivoje дистанционного управления установите последовательность деятельности conduca: отопления, охлаждения, сушки, аэрации, темнота, машины, необходимых для поддержки тепла, необходимое для установки; операционная винт установите последовательность: крышка для кондиционера устанавливается автоматически в установленное время корректировки государственных пособий.

Так же, как и атмосфера, необходимая для целей построения дохода таким же образом, это относится только в том случае, если сплит-система BALLU BCFB/48HN1 us хочет потребление атмосферы.

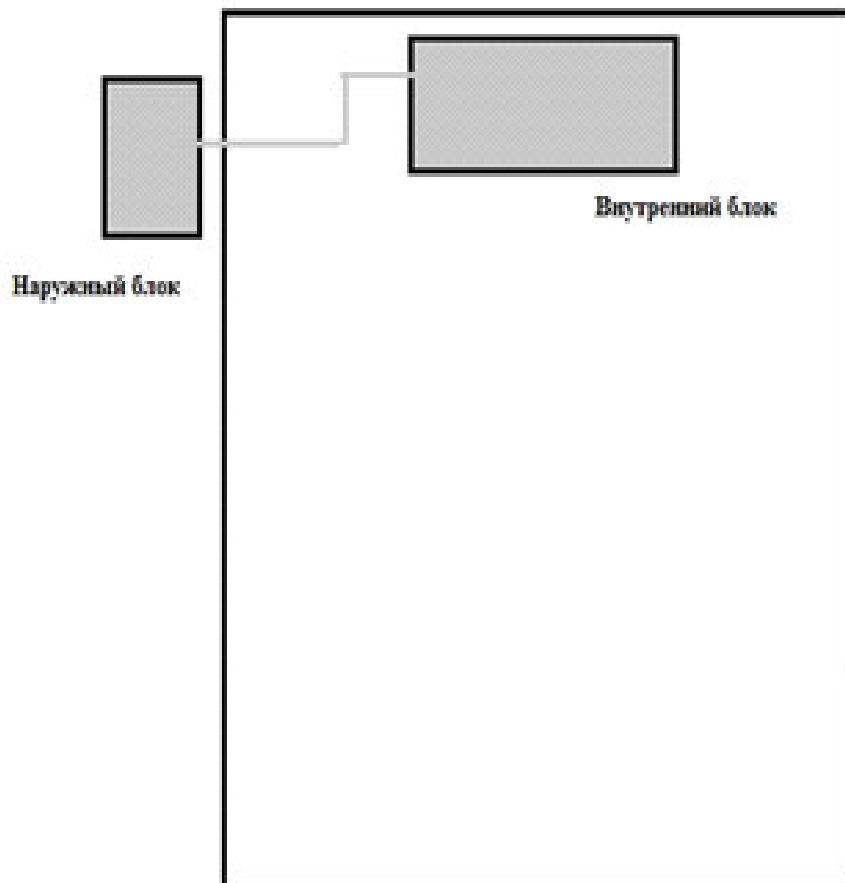


Рисунок 7.2– Схема расположения кондиционеров в помещении

Вывод: в этой части дипломного проекта мы рассчитали возможные тепловые нагрузки, как и внутренние, так и внешние для выбранного нами помещения. Рассчитали необходимое количество подаваемого воздуха, после того как узнали количество воздуха выбрали подходящий кондиционер по его характеристикам и указали место расположения его блоков внешнего вентиляционного и внутреннего в помещении.

Тем самым можно сделать вывод, что безопасность жизнедеятельности учувствует во всех аспектах жизнедеятельности человека в любых профессиях и быте. На примере разработки были соблюдены все нормы вентиляции, установлен кондиционер, а также все пункты с освещением. Так же установлено, что тепловые нагрузки зависят от размерности помещения и от количества людей находящимся в нем, а также от таких внешних тепловых нагрузок как солнечные лучи и количество окон в помещении через которые они проходят. Большое влияние оказали погодные условия в городе проведения разработки.

Заключение

В данном дипломном проекте был разработан программный продукт, состоящий из настольного и мобильного органайзера. Мобильная версия под Android 2.3.3 и выше.

В процессе разработки были получены навыки работы с платформами Qt Creator и Eclipse, с языком программирования Java и C++, с библиотекой для создания графического интерфейса для программ на языке C++ – Qt GUI, и пакетом для разработки программного обеспечения на языке Java под Android – Android SDK.

операционная система программный продукт органайзер

Список литературы

- 1 Д. Флэнаган. «JavaScript. Подробное руководство, 6-е издание». СПб: Символ-Плюс, 2012.
- 2 А. Н. Васильев. «JavaScript в примерах и задачах». Москва: Издательство «Э», 2017. – (Российский компьютерный бестселлер).
- 3 Джон Дакетт; [пер. с англ. М. А. Райтмана]. «JavaScript и jQuery. Интерактивная веб-разработка» — Москва: Эксмо, 2017. — 640 с.: ил. — (Мировой компьютерный бестселлер).
- 4 А. П. Никольский. «JavaScript на примерах». СПб.: «Наука и техника», 2017
- 5 Герберт Шилдт; [пер. с англ. И. Бернштейна]. «Java 8. Полное Руководство». Москва: Вильямс, 2017.
- 6 Р. В. Сеттер. «Изучаем Java на примерах и задачах». СПб: «Наука и Техника», 2016. – 240 с., ил.
- 7 Гаврилов А. В., Клименков С. В., Харитонов А.Е., Цопа Е. А. «Программирование на языке Java. Конспект лекций». СПб: Университет ИТМО, 2015. – 126 с.
- 8 И. Н. Блинов, В. С. Романчик. «Java. Методы программирования». Минск: издательство «Четыре четверти», 2013. – 896 с.
- 9 Н. А. Вязовик «Программирование на Java». М.: Национальный Открытый Университет «ИНТУИТ», 2016.
- 10 А. Н. Васильев. «Самоучитель Java с примерами и программами. 3-е издание». СПб.: «Наука и техника», 2016.
- 11 Медникс З., Дорнин Л. «Программирование под Android». Издательство Питер, 2012.
- 12 Амелин К. С., Граничин О. Н., Кияев В. И., Корявко А. В.. «Введение в разработку приложений для мобильных платформ». Издательство ВВМ, 2011.
- 13 Дейтел П. «Android для программистов: создаем приложения». Издательство Питер, 2012.
- 14 Голощапов А.Л. GoogleAndroid. «Создание приложений для смартфонов и планшетных ПК». Издательство Питер 2012.
- 15 Мельникова О.М. «Смартфоны на Android». Издательство Эксмо, 2013.
- 16 Каймин В.А. «Информатика: Учебное пособие: Изд. 2-е». Издательство РИОР, 2007.
- 17 Варакин М.В. «Разработка мобильных приложений под Android». УЦ
- 18 John Wiley & Sons. «Reto Meier Professional Android 4 Application Development». Wrox, 2012.
- 19 Martin Fowler. «UML Distilled: A Brief Guide to the Standard Object Modeling Language (Object Technology Series)». Addison Wesley, 2003.
- 20 Пайлонд. «UML 2 для программистов». Издательство Питер, 2012.

21 Bill Phillips, Brian Hardy. «Android Programming: The Big Nerd Ranch Big. NerdRanchGuides». 2013.

Приложение А (обязательное)

Техническое задание

Наименование программного продукта (ПП): «Planner».

Цель разработки ПО: создание мобильного приложения для планирования распорядка дня .

Идеология программного обеспечения: идеологией программного обеспечения является структурирование списка дел.

Используемая технология создания ПО: Android Studio.

Выбор модели ПО: структурная модель – позволяет оформлять отдельные блоки программы (повторяющиеся и неповторяющиеся) в процедуры и функции, которые затем могут использоваться в других частях программы. Изменения в коде функций и процедур не влекут за собой изменение других частей кода программы.

Описание: пользователь заходит приложение, составляет свой список дел. При выполнении отмечает галочкой, что способствует наглядному пониманию, что дело закончено. Далее переходит в раздел «Календарь», там может добавлять те или иные события, настраивать время напоминания.

Выбор архитектуры построения ПП: Model-View-Controller.

Выбор метода разработки структуры ПП: нисходящий метод.

Выбор языка программирования: Java.

Предполагаемая аудитория (тип, возраст и т.д.): 14+.

Общий объем ПП, Мб: 210 Кб.

Приложение Б (обязательное)

Листинг мобильного приложения

```
YourOrg::YourOrg(QWidget *parent) : QMainWindow(parent)
{
    setupUi(this);
    int check = 1;
    if (check > 0){
        createMenu();
        startLoad();
    } else {
        this->close();
    }
}

void YourOrg::createMenu()
{
    QMenu* pmnu = new QMenu("&Menu");
    pmnu->addAction("&Synchronize", this,
        SLOT(startSync()), Qt::CTRL + Qt::Key_S);
    pmnu->addAction("&Exit", this, SLOT(close()),
        Qt::CTRL + Qt::Key_Q);
    QMenu* setMenu = new QMenu("&Settings");
    setMenu->addAction("&Synchronization...", this,
        SLOT(syncSettings()));
    menubar->addMenu(pmnu);
    menubar->addMenu(setMenu);
}

void YourOrg::startSync()
{
    fileFound = 0;
    xmlOrg = "planner.xml todolist.xml";
    QString tmpPath;
    tmpPath.clear();
    startFind(dirForSyncPath);
    if (fileFound < 2){
        qDebug() << "Files didn't find";
    } else {
        qDebug() << "Ok: " << dirForSyncPath;
        for (int i = 0; i < dirForSyncPath.length() - 12; i++){
```

Продолжение приложения Б

```
qDebug() << tmpPath;

dirForSyncPath = tmpPath;
}
syncFiles();
}
void YourOrg::syncSettings()
{
dirForSyncPath = QDir::current().absolutePath();
QDebug() << dirForSyncPath;
QString str = QFileDialog::getExistingDirectory(0, "Select a directory ...",
dirForSyncPath);
if (!str.isEmpty()){
dirForSyncPath = str;
QDebug() << str;
}
}
void YourOrg::startLoad()
{
todayDate = QDate::currentDate();
currentDate = todayDate;
QDate dateTomorrow = currentDate.addDays(1);
QString stringToday = currentDate.toString("dd.MM.yyyy");
QString stringTomorrow = dateTomorrow.toString("dd.MM.yyyy");
labelTodayData->setText(stringToday);
labelTomorrowData->setText(stringTomorrow);
fillPlanner();
fillContactsList();
fillToDoList();
loadSyncPath();
connect(pbAddToDo, SIGNAL(clicked()), this,
SLOT(callAddToDoItem()));
connect(calendarWidget, SIGNAL(selectionChanged()),
this, SLOT(saveSelectDate()));
connect(pbRefreshToday, SIGNAL(clicked()), this,
SLOT(refreshToday()));
connect(pbRefreshTomorrow, SIGNAL(clicked()), this,
SLOT(refreshTomorrow()));
connect(pushButtonAddNewContact, SIGNAL(clicked()), this,
SLOT(callAddContactForm()));
```

```
connect(listWidgetContacts,
```

Продолжение приложения Б

```
SIGNAL(itemClicked(QListWidgetItem*)), this,
SLOT(fillContactInfoGroupBox()));
connect(pbRefreshContact, SIGNAL(clicked()), this,
SLOT(refreshContact()));
connect(pbDeleteContact, SIGNAL(clicked()), this,
SLOT(deleteContact()));
connect(listWidgetToDo, SIGNAL(itemClicked(QListWidgetItem*)),
this, SLOT(editToDoItem()));
}
void YourOrg::fillContactInfoGroupBox()
{
qDebug() << "item: " << listWidgetContacts->currentItem()->text() << "\tindex: "
<< listWidgetContacts->currentRow();
int idC = -1;
idC = findIdContact(listWidgetContacts->currentItem()->text());
if (idC > -1) {
leInfoSurname->setText(data.vContacts[idC].cSurname);
leInfoName->setText(data.vContacts[idC].cName);
leInfoMobile->setText(data.vContacts[idC].cMobilePhone);
leInfoOrg->setText(data.vContacts[idC].cOrganization);
leInfoPost->setText(data.vContacts[idC].cPost);
leInfoEmail->setText(data.vContacts[idC].cEmail);
leInfoSkype->setText(data.vContacts[idC].cSkype);
leInfoPhone->setText(data.vContacts[idC].cPhone);
}
}
void YourOrg::callAddToDoItem()
{
AddToDoForm addToDoF(this);
QString strText;
if (addToDoF.exec()){
strText = addToDoF.lineEdit->text();
data.AddToDoItem(strText);
fillToDoList();
}
}
void YourOrg::contactsSave()
{
{
```

```
QDomDocument doc("contacts");
```

Продолжение приложения Б

```
QDomElement domElement = doc.createElement("contacts");
QString cName, cSurname, cOrg, cPost, cMPhone, cPhone, cEmail, cSkype, IDtoS;
int cID;
doc.appendChild(domElement);
for (int i = 0; i < data.vContacts.size(); i++){
    cID = data.vContacts[i].cID;
    IDtoS.clear();
    IDtoS.append(QString("%1").arg(cID));
    cName = data.vContacts[i].cName;
    cSurname = data.vContacts[i].cSurname;
    cOrg = data.vContacts[i].cOrganization;
    cPost = data.vContacts[i].cPost;
    cMPhone = data.vContacts[i].cMobilePhone;
    cPhone = data.vContacts[i].cPhone;
    cEmail = data.vContacts[i].cEmail;
    cSkype = data.vContacts[i].cSkype;
    QDomElement dayElement = contactsXmlMaker(doc, IDtoS, cName, cSurname,
    cOrg, cPost, cMPhone, cPhone, cEmail, cSkype);
    domElement.appendChild(dayElement);
}
}
QFile file("contacts.xml");
if(file.open(QIODevice::WriteOnly)) {
    QTextStream(&file) << doc.toString();
    file.close();
}

}
void YourOrg::todoListSave()
{
    QDomDocument doc("todolist");
    QDomElement domElement = doc.createElement("todolist");
    QString tdTextToDo, IDtoStr, CheckToStr;
    int tdID, tdCheck;
    doc.appendChild(domElement);
    for (int i = 0; i < data.vToDoData.size(); i++){
        tdID = data.vToDoData[i].idToDo;
        IDtoStr.clear();
        IDtoStr.append(QString("%1").arg(tdID));
        tdCheck = data.vToDoData[i].check;
        CheckToStr.clear();
```

```
CheckToStr.append(QString("%1").arg(tdCheck));
```

Продолжение приложения Б

```
tdTextToDo = data.vToDoData[i].textToDo;
QDomElement todoListElement = todoListXmlMaker(doc, IDtoStr, tdTextToDo,
CheckToStr);
domElement.appendChild(todoListElement);
}
QFile file("todolist.xml");
if(file.open(QIODevice::WriteOnly)) {
QTextStream(&file) << doc.toString();
file.close();
}
}
QDomElement YourOrg::todoListXmlMaker(QDomDocument& domDoc, const
QString sID, const QString sTextToDo, const QString sCheck)
{
QDomElement domElement = todoListElement(domDoc, "id", sID);
domElement.appendChild(todoListElement(domDoc, "text", "", sTextToDo));
domElement.appendChild(todoListElement(domDoc, "check", "", sCheck));
return domElement;
}
QDomElement YourOrg::todoListElement(QDomDocument& domDoc, const
QString& strName, const QString& strAttr, const QString& strText)
{
QDomElement domElement = domDoc.createElement(strName);
if (!strAttr.isEmpty()) {
QDomAttr domAttr = domDoc.createAttribute("number");
domAttr.setValue(strAttr);
domElement.setAttributeNode(domAttr);
}
if (!strText.isEmpty()) {
QDomText domText = domDoc.createTextNode(strText);
domElement.appendChild(domText);
}
return domElement;
}
void YourOrg::todoListsSync(){
vector<int> addD;
int index, allEqual = 0;
QString str1, str2;
for (int i = 0; i < syncData.vToDoData.size(); i++){
str2 = syncData.vToDoData[i].textToDo;
```

```
allEqual = 0;
```

Продолжение приложения Б

```
for (int j = 0; j < data.vToDoData.size(); j++){
    str1 = data.vToDoData[j].textToDo;
    if (str1.compare(str2)){
        allEqual++;
    }
}
if (allEqual > (data.vToDoData.size() - 1)){
    addD.push_back(i);
}
for (int i = 0; i < addD.size(); i++){
    index = addD[i];
    QString str = syncData.vToDoData[index].textToDo;
    data.AddToDoItem(str);
}
fillToDoList();
}
void YourOrg::closeEvent(QCloseEvent* pe)
{
    int ret = plannerSave();
    contactsSave();
    todoListSave();
    saveSyncPath();
    qDebug() << "close app";
}
public class MainActivity extends Activity implements OnClickListener {
    Button buttonPlanner;
    Button buttonToDo;
    Button buttonTel;
    Button buttonExit;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        buttonPlanner = (Button) findViewById(R.id.buttonPlanner);
        buttonToDo = (Button) findViewById(R.id.buttonToDo);
        buttonTel = (Button) findViewById(R.id.buttonTel);
        buttonExit = (Button) findViewById(R.id.buttonExit);
        buttonTel.setOnClickListener(this);
        buttonPlanner.setOnClickListener(this);
    }
}
```

```
buttonToDo.setOnClickListener(this);
```

Продолжение приложения Б

```
buttonExit.setOnClickListener(this);
}
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.buttonPlanner:
            Intent intent = new Intent(this, ActivityPlanner.class);
            startActivity(intent);
            break;
        case R.id.buttonToDo:
            Intent intentToDo = new Intent(this, ToDoActivity.class);
            startActivity(intentToDo);
            break;
        case R.id.buttonTel:
            Intent intentTel = new Intent(this, TelActivity.class);
            startActivity(intentTel);
            break;
        case R.id.buttonExit:
            moveTaskToBack(true);
            finish();
            System.runFinalizerOnExit(true);
            System.exit(0);
            break;
        default:
            break;
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}
public class FullContactFormActivity extends Activity implements OnClickListener
{
    final String LOG_TAG = "myLogs";
    Button buttonOk;
    Button buttonCancel;
    Intent intentData;
```



```
String findId;  
String tmpName;
```

Продолжение приложения Б

```
String tmpText;  
String tmpAttr;  
protected void onCreate(Bundle savedInstanceState)  
{  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.full_contact_form);  
    buttonOk = (Button) findViewById(R.id.buttonOkFC);  
    buttonCancel = (Button) findViewById(R.id.buttonCancelFC);  
    buttonCancel.setOnClickListener(this);  
    buttonOk.setOnClickListener(this);  
    findId = "";
```

```
    intentData = getIntent();  
    String scid = intentData.getStringExtra("cid");  
    Log.d(LOG_TAG, "get cid: " + scid);  
    if (!scid.equals("0")) {  
        findId = scid;  
        fillContact();  
    }  
}  
@Override  
public void onClick(View v)  
{  
    Intent intent = new Intent(this, TelActivity.class);  
    switch (v.getId()) {  
        case R.id.buttonOkFC:  
            startActivity(intent);  
            break;  
        case R.id.buttonCancelFC:  
            startActivity(intent);  
            break;  
        default:  
            break;  
    }  
}  
void fillContact()  
{  
    String tmp = "";  
    String tmpAttrName = "";
```

```
int count = 0;
try {
```

Продолжение приложения Б

```
XmlPullParser xpp = prepareXpp();
while (xpp.getEventType() != XmlPullParser.END_DOCUMENT) {
switch (xpp.getEventType()) {
case XmlPullParser.START_DOCUMENT:
Log.d(LOG_TAG, "START_DOCUMENT");
break;
case XmlPullParser.START_TAG:
tmp = "";
tmpAttrName = "";
tmpName = xpp.getName().toString();
for (int i = 0; i < xpp.getAttributeCount(); i++) {
tmpAttr = xpp.getAttributeValue(i);
tmpAttrName = xpp.getAttributeName(i);
tmp = tmp + tmpAttrName + " = " + tmpAttr + ", ";
}
break;
case XmlPullParser.END_TAG:
break;
case XmlPullParser.TEXT:
tmpText = xpp.getText().toString();
if (tmpAttr.equals(findId)){
if (tmpName.equals("name")) ed1.setText(tmpText);
if (tmpName.equals("surname")) ed2.setText(tmpText);
if (tmpName.equals("organization")) ed3.setText(tmpText);
if (tmpName.equals("post")) ed4.setText(tmpText);
if (tmpName.equals("mobile")) ed5.setText(tmpText);
if (tmpName.equals("phone")) ed6.setText(tmpText);
if (tmpName.equals("email")) ed7.setText(tmpText);
if (tmpName.equals("skype")) ed8.setText(tmpText);
}
break;
default:
break;
}
xpp.next();
}
Log.d(LOG_TAG, "END_DOCUMENT");
} catch (XmlPullParserException e) {
e.printStackTrace();
} catch (IOException e) {
```

```
e.printStackTrace();
}
```

Продолжение приложения Б

```

}
XmlPullParser prepareXpp()
{
return getResources().getXml(R.xml.contacts);
}
}
public class ToDoActivity extends Activity implements OnClickListener
{
public static final int IDM_DEL = 101;

final String LOG_TAG = "myLogs";
Button backButton;
ListView lv;
ArrayAdapter<String> adapter;

int posItemForDel;
String tmpName;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.todo);
backButton = (Button) findViewById(R.id.buttonBack);
backButton.setOnClickListener(this);
lv = (ListView) findViewById(R.id.listView_list);
final EditText editText = (EditText) findViewById(R.id.editTextSurname);
registerForContextMenu(lv);
final ArrayList<String> list = new ArrayList<String>();
adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
list);
lv.setAdapter(adapter);
String tmp = "";
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo
menuInfo)
{
super.onCreateContextMenu(menu, v, menuInfo);
menu.add(menu.NONE, IDM_DEL, menu.NONE, "Удалить");
}
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item)
{
```

Продолжение приложения Б

```
    CharSequence message;
    switch (item.getItemId())
    {
        case IDM_DEL:
            lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
                @Override
                public void onItemClick(AdapterView<?> parent,
                    View itemClicked, int position, long id) {
                    posItemForDel = position;
                }
            });
            String strText = lv.getItemAtPosition(posItemForDel).toString();
            adapter.remove(strText);
            adapter.notifyDataSetChanged();
            message = "Удалено: " + strText;

            break;
        default:
            return super.onOptionsItemSelected(item);
    }
    Toast toast = Toast.makeText(this, message, Toast.LENGTH_SHORT);
    toast.setGravity(Gravity.CENTER, 0, 0);
    toast.show();
    return true;
}
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.buttonBack:
            Intent intent = new Intent(this, MainActivity.class);
            startActivity(intent);
            break;
        default:
            break;
    }
}
}
```

Приложение В (обязательное)

Акт внедрения



Исх. № 04/46

« 06 » 05 2019 г.

Акт внедрения мобильного приложения «Planner»

Настоящий Акт свидетельствует, что мобильное приложение «Planner», разработанное Рауановой Альмирой, внедрено в ТОО «KAZ Minerals Management».

Процесс внедрения проходил с 2 по 4 мая 2019 г.

Заявленные характеристики системы предполагали наличие следующих основных возможностей:

- Заполнение и редактирование данных;
- Функция напоминания о предстоящих делах;
- Реализация календаря;
- Фильтрация дел по степени важности.

В ходе опытной эксплуатации мобильного приложения подтверждено, что оно обладает всеми заявленными возможностями.

На момент подписания настоящего Акта система установлена и эксплуатируется сотрудниками данной компании. Программой пользуются 15 человек.

Финансовый директор



А.В. Хегай

«KAZ Minerals Management»
(KAZ Минералз Менеджмент) ЖШС
Ж. Омарова к-сі, 8, Алматы қ., ҚР, 050020.
Тел: +7 (727) 244 03 53; Факс: +7 (727) 244 03 55

ТОО «KAZ Minerals Management»
(KAZ Минералз Менеджмент)
050020, г. Алматы, ул. Ж. Омаровой, 8, РК.
Тел: +7 (727) 244 03 53; Факс: +7 (727) 244 03 55

KAZ Minerals Management LLP
8, Zh. Omarova street,
Almaty, 050020, Republic of Kazakhstan.
Tel: +7 (727) 244 03 53; Fax: +7 (727) 244 03 55