

Министерство образования Новосибирской области
ГБПОУ НСО «Новосибирский авиационный технический колледж
имени Б.С. Галушака»

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «КАЛЬКУЛЯТОР
СНА»**

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных
систем

МДК01.03 Разработка мобильных приложений

НАТКиГ.210100.043.000ПЗ

Выполнил:

студент группы ПР-315

Архипов К.С.

2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	4
1.1 Описание предметной области	4
1.2 Образ клиента	4
1.3 Сценарии	4
1.4 Сбор и анализ прототипов	5
2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ	7
2.1 UI/UX дизайн приложения	7
2.2 Выбор технологии, языка и среды программирования	12
3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ	13
3.1 Разработка мультимедийного контента	13
3.2 Описание используемых плагинов	14
3.3 Описание разработанных процедур и функций	15
4 ТЕСТИРОВАНИЕ	18
4.1 Протокол тестирования дизайна приложения	18
4.2 Протокол тестирования функционала приложения	20
ЗАКЛЮЧЕНИЕ	21
БИБЛИОГРАФИЯ	22
Приложение А	23

					НАТКиГ.210100.43.000ПЗ									
Изм.	Лист	№ докум	Подпись	Дата										
Разраб		Архипов			РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «КАЛЬКУЛЯТОР СНА»					Литера	Лист	Листов		
Пров		Климова								у		2	26	
Н. Контр		Тышкевич								ПР-315				
Утв		Тышкевич												

ВВЕДЕНИЕ

В настоящее время, проблемы со сном знакомы почти каждому человеку, тем более студенту (причиной может служить работа или учеба). Приложения для контроля сна способны помочь людям справиться с бессонницей, проблемами с пробуждением, а также восстановить режим. Поэтому, вопросы, касающиеся правильного сна, являются очень актуальными.

Целью курсового проекта является разработка мобильного приложения – калькулятор сна «Sleepy» (далее по тексту – «Приложение»).

Для достижения цели поставлены следующие задачи:

1. изучить структуру сна человека;
2. разработать макет приложения;
3. определить основной функционал;
4. выбрать основные библиотеки для работы приложения;
5. разработать мобильное приложение в соответствии с макетом и функционалом.

Объектом исследования можно назвать структуру сна человека.

Предметом является контроль процессов засыпания и пробуждения с помощью мобильного устройства.

Практическая значимость работы заключается в помощи людям, у которых имеются проблемы со сном.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Описание предметной области

На сегодняшний день почти каждый человек сталкивался с такими проблемами со сном, как долгое время пробуждения или засыпания. Например, по данным исследования Всероссийского центра изучения общественного мнения от 13 марта 2020 года около 50% опрошенных имеют проблемы со сном.

Для того, чтобы исправить проблемы со сном, можно рассчитать время для засыпания или пробуждения. Для расчета необходимо знать в какое время нужно заснуть или проснуться, длительность цикла сна человека, а также время, которое человек тратит на засыпание.

В целях экономии времени, а также упрощения расчета, данный процесс можно автоматизировать.

1.2 Образ клиента

Данное приложение подойдет для любого человека, который имеет проблемы со сном (например, плохое самочувствие после пробуждения) и хочет их исправить. Поскольку в приложении присутствует как русский язык, так и английский, то пользоваться им могут и англоговорящие граждане. Пользователем может быть как женщина, так и мужчина. Ограничений по возрасту нет.

1.3 Сценарии

Студент Валерий учится на 2 курсе в колледже. Он решил восстановить режим сна. Для этого ему необходимо ложиться каждый день в одно и то же время. Но он не может правильно подобрать время, на которое ему необходимо поставить будильник. Из-за этого, после пробуждения, он чувствует себя уставшим и невыспавшимся.

					НАТКиГ.210300.43.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		4

Преподаватель Сергей преподает уроки математики. Он допоздна проверял работы студентов. Ему необходимо встать ровно в 8:00, но поскольку он собирается лечь спать позднее, чем обычно, то он может не выспаться. Для того, чтобы Сергей смог хорошо выспаться ему нужно рассчитать оптимальное время, в которое нужно лечь спать, а также поставить будильник на 8:00 утра.

1.4 Сбор и анализ прототипов

На данный момент существует немного приложений, которые занимаются расчетом времени сна. Из имеющихся в Google Play рассмотрим «Sleep Time» и «Sleep Calculator». «Sleepy» - данное разрабатываемое приложение. Рассматриваемые приложения имеют функционал для расчета времени пробуждения и засыпания. «Sleep Calculator» имеет функции отслеживания сна и сбора статистики.

Сравнение по основным критериям данных приложений представлено в таблице 1.

Таблица 1 – Сравнение приложений

Параметры	Sleep Calculator	Sleep Time
Стоимость	Бесплатно	Бесплатно
Язык	Русский, английский	Русский
Настройка времени засыпания	Есть	Есть
Настройка длительности цикла сна	Отсутствует	Есть
Настройка количества отображаемых циклов	Отсутствует	Есть

Продолжение таблицы 1 – Сравнение приложений

Выбор темы интерфейса	Отсутствует	Отсутствует
Настройка формата времени	Есть	Есть
Будильник	Есть	Есть (иногда не срабатывает)
Возможность использования системного будильника	Отсутствует	Отсутствует
Настройка громкости будильника (из приложения)	Отсутствует	Отсутствует
Выбор мелодии будильника	Есть	Отсутствует

В результате сравнения можно сделать вывод о том, что в разрабатываемом приложении необходимо реализовать удобный калькулятор расчетов времени для сна, а также все необходимые настройки для расчетов. Дополнительно необходимо реализовать корректно работающий будильник.

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

2.1 UI/UX дизайн приложения

2.1.1 Цветовая схема приложения и логотип

Проектирование дизайна и прототипа приложения производилось в графическом редакторе «Figma».

Определены две основные цветовые схемы. Первая цветовая схема выполнена в более ярких фиолетовых тонах и представлена на рисунке 1.



Рисунок 1 – Светлая цветовая схема

Вторая палитра цветов более темных коричневых оттенков (Рисунок 2). Разделение на темную и светлую тему в приложении необходимо, так как, например, пользователю может быть не приятно использовать светлую тему ночью.

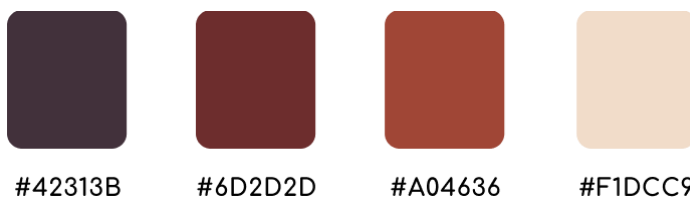


Рисунок 2 – Темная цветовая схема приложения

На основе темной цветовой схемы разработан логотип приложения (Рисунок 3). На переднем плане логотипа в центре находится луна и облака, поскольку именно они ассоциируются у человека с темным ночным временем суток и со сном.



Рисунок 3 – Логотип приложения

2.1.2 Дизайн приложения

С помощью выбранных цветовых палитр разработан полный дизайн, состоящий из 6 экранов. На рисунке 4 представлен дизайн «Обучающего экрана» с использованием светлой палитры цветов.

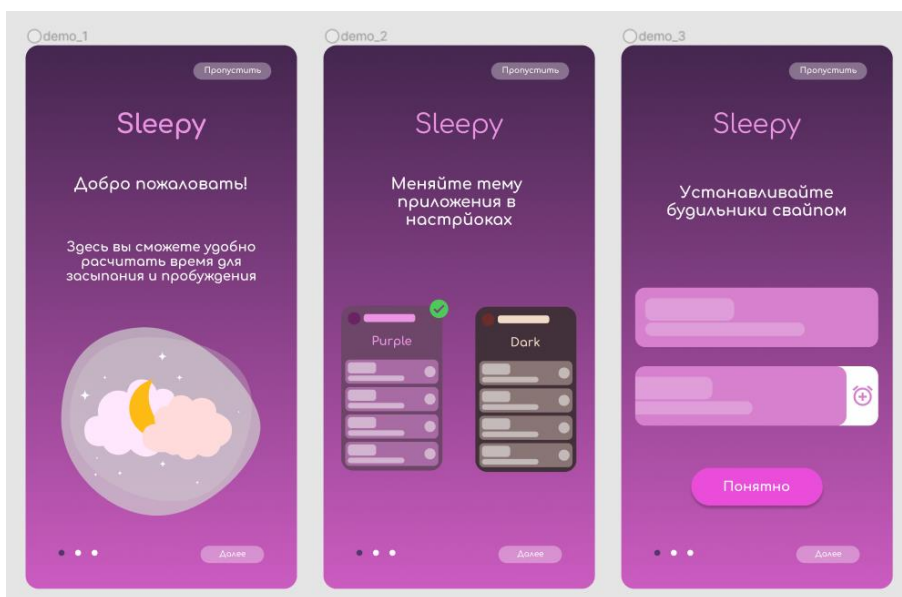


Рисунок 4 – «Обучающий экран» в светлой теме

На данном экране представлены картинки с кратким описанием – руководством по использованию приложения.

На рисунке 5 изображены дизайны для двух схожих экранов – «Когда лечь спать» и «Когда проснуться». На данных экранах присутствует виджет для выбора времени, а также список с рассчитанным временем.

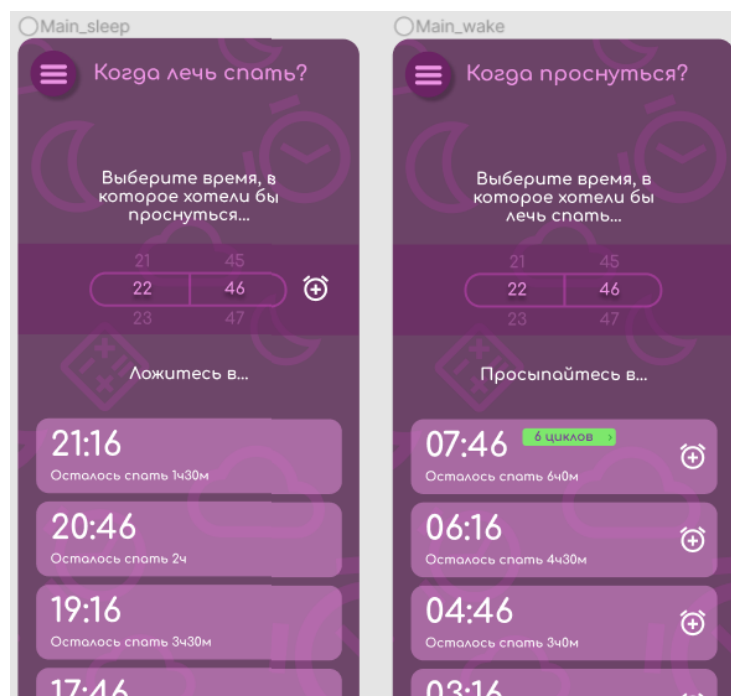


Рисунок 5 – Экраны «Когда лечь спать» и «Когда проснуться» в светлой теме

Экран «Будильник» на рисунке 6 состоит из списка установленных будильников и кнопки установки нового будильника в нижней части экрана.

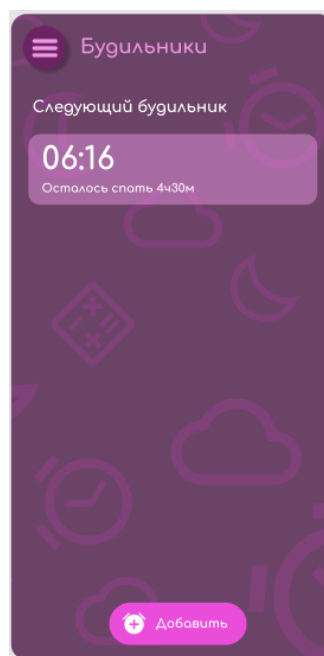


Рисунок 6 – Экран «Будильник» в светлой теме

На рисунке 7 представлена часть дизайна экрана «Настройки». Макет разделен на 3 логические части: настройки интерфейса; основные настройки; настройки будильника.

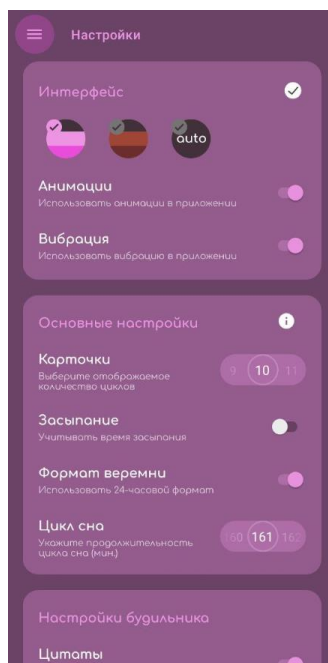


Рисунок 7 – Экран «Настройки» в светлой теме

Навигация по рассмотренным выше экранам осуществляется при помощи бокового меню, макет которого изображен на рисунке 8. Меню содержит кнопки для навигации на экраны: «Когда лечь спать»; «Когда проснуться»; «Будильник»; «Настройки».

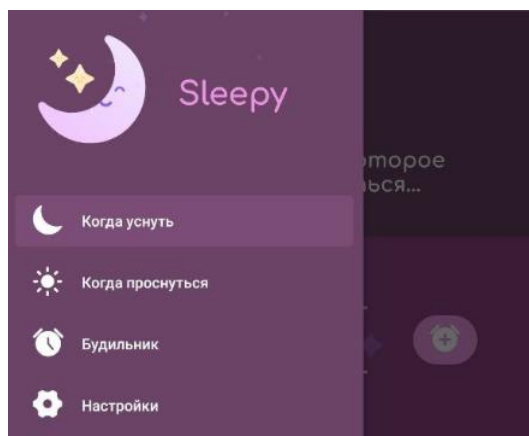


Рисунок 8 – Дизайн бокового меню приложения

Дизайн разработанных экранов в темной теме представлен на рисунке 9.

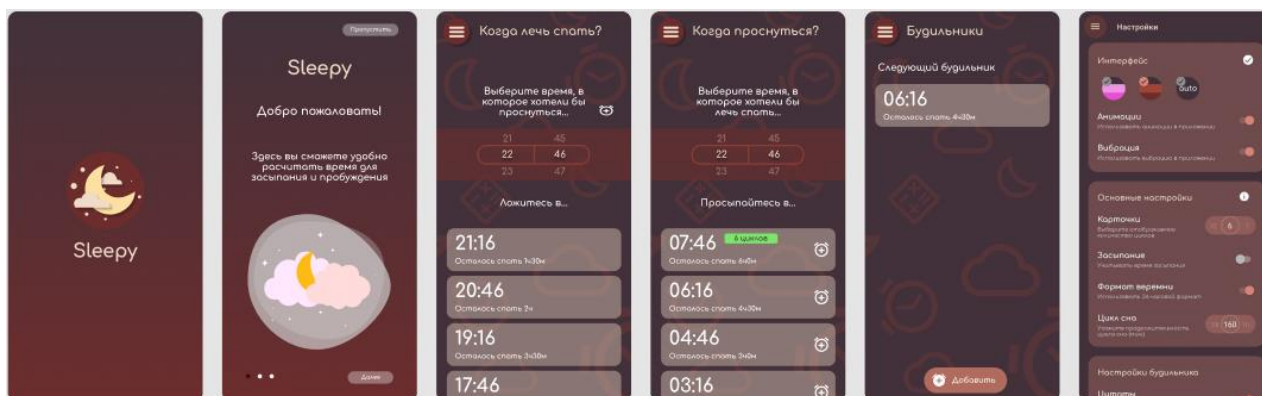


Рисунок 9 – Дизайн приложения в темной теме

2.1.3 Диаграмма передвижения пользователя

На основе созданных макетов экранов разработан прототип работы приложения в Figma. Прототип построен таким образом, чтобы пользователь смог достигнуть своей цели при использовании приложения как можно быстрее. Диаграмма передвижения пользователя представлена на рисунке 10.

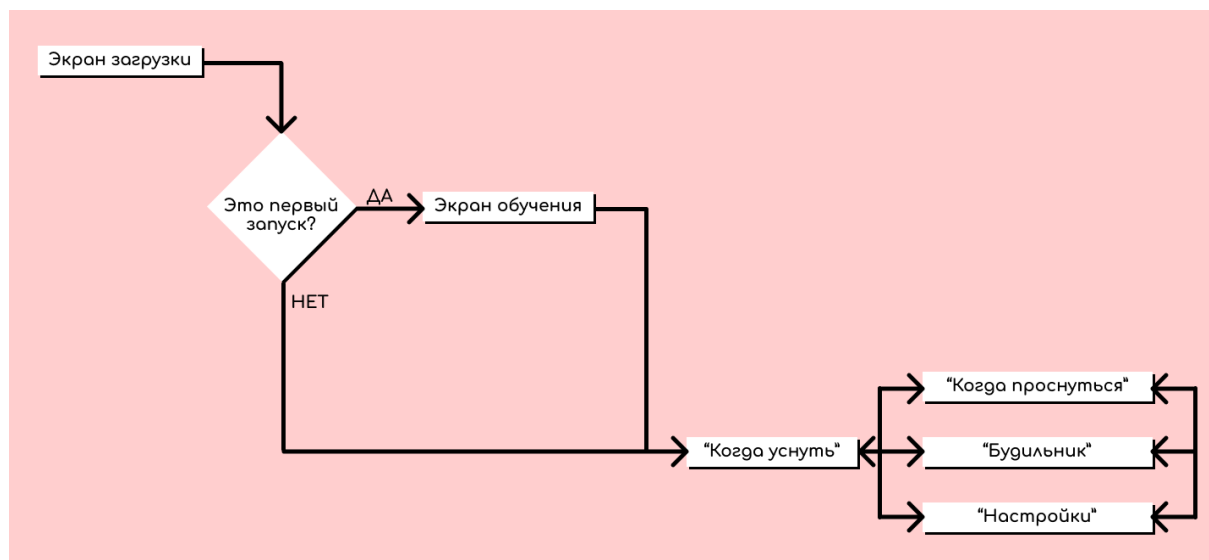


Рисунок 10 – Диаграмма передвижения пользователя

2.2 Выбор технологии, языка и среды программирования

Поскольку калькулятор сна будет разрабатываться для ОС Android, используется среда разработки Android Studio, так как она обладает всем необходимым функционалом для разработки мобильных нативных приложений.

В качестве основного языка программирования для разработки приложения выбран Kotlin, поскольку данный язык обладает всем необходимым функционалом для решения данной задачи.

					НАТКиГ.210300.43.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		12

3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

3.1 Разработка мультимедийного контента

Весь мультимедийный контент разработан с помощью языка разметки XML. Верстка выполнена полностью в соответствии с ранее разработанным дизайном.

Все иконки и другие визуальные ресурсы импортированы из Figma в проект в виде XML-файлов (рисунок 11). Данный подход снижает общий вес арк-файла приложения и проекта, также отсутствуют потери в качестве мультимедийного контента. Данный тип ресурсов необходимо хранить в директории «drawable»

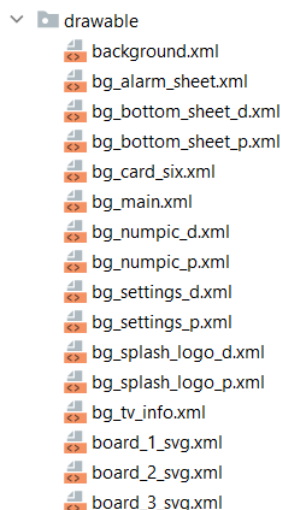


Рисунок 11 – Используемые иконки и визуальные ресурсы

Поскольку в приложении используются две цветовые схемы, то их необходимо занести в файл colors.xml (рисунок 12).

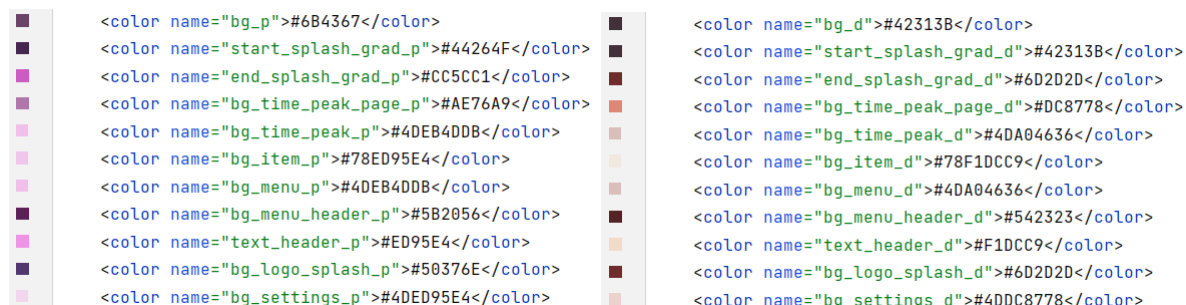


Рисунок 12 – Цвета, используемые в приложении

Ссылки на эти цвета используются для описания цветов темы приложения. Палитра для светлой темы описана в файле themes.xml, для темной в themes.xml (night). На рисунке 13 представлены светлая и темная тема приложения.

```
<!--СВЕТЛАЯ ТЕМА-->
<item name="colorPrimary">@color/text_header_p</item>
<item name="colorPrimaryVariant">@color/bg_p</item>
<item name="colorOnPrimary">@color/bg_p</item>
<item name="android:textColorPrimary">@color/white</item>
<item name="android:windowActivityTransitions">true</item>
<item name="colorSecondary">@color/text_header_p</item>
<item name="colorSecondaryVariant">@color/checked_variant</item>
<item name="colorOnSecondary">@color/text_header_p</item>
<item name="drawerArrowStyle">@style/DrawerArrowStyle</item>
<item name="android:textColorSecondary">@color/white</item>

<!--ТЕМНАЯ ТЕМА-->
<item name="colorPrimary">@color/bg_time_peak_page_d</item>
<item name="colorPrimaryVariant">@color/bg_d</item>
<item name="colorOnPrimary">@color/bg_d</item>
<item name="android:textColorSecondary">@color/white</item>
<item name="android:windowActivityTransitions">true</item>
<item name="colorSecondary">@color/bg_time_peak_page_d</item>
<item name="colorSecondaryVariant">@color/checked_variant</item>
<item name="colorOnSecondary">@color/text_header_d</item>
<item name="drawerArrowStyle">@style/DrawerArrowStyle</item>
<item name="android:textColorPrimary">@color/white</item>
```

Рисунок 13 – Светлая и темная тема приложения

3.2 Описание используемых плагинов

В проекте используются библиотеки с различными компонентами и элементами компоновки. Список библиотек с их описанием представлен в таблице 2.

Таблица 2 – Список используемых плагинов

Разработчик/наименование	Описание
Airbnb/Lottie	Популярная библиотека для использования json-анимаций в приложении
ShawnLin013/NumberPicker	Библиотека, которая предоставляет простой и настраиваемый виджет для выбора числа NumberPicker
chthai64/SwipeRevealLayout	Данный плагин упрощает работу с разработкой функции свайпа для элементов списка
google/flexbox-layout	Официальная библиотека от Google, которая предоставляет удобный и многофункциональный контейнер для компоновки элементов на экране

3.3 Описание разработанных процедур и функций

После верстки всех экранов реализованы функции для навигации с помощью бокового меню.

Функционал для экрана «Когда уснуть» содержит такие функции, как расчет оптимального времени засыпания, сброс времени на специальном виджете для выбора времени TimePicker и установка будильника. Рассчитанное время выводится в виде списка в специальный, предназначенный для этого компонент RecyclerView.

Функции установки и отмены будильника вынесены в отдельный класс. Вся работа с будильником происходит при помощи класса, обеспечивающего доступ к системным службам оповещения AlarmManager. Показ экрана

будильника при срабатывании осуществлен с помощью PendingIntent (действие при срабатывании будильника). Чтобы у будильника было какое-либо звуковое сопровождение, используется класс, который способен воспроизводить звуковые файлы MediaPlayer.

Для экрана «Когда проснуться» реализованы функции расчета оптимального времени для пробуждения и сброса времени внутри TimePicker. Рассчитанное время выводится в виде списка в RecyclerView, адаптер для которого содержит функцию для установки будильника по свайпу элемента.

Экран «Будильник» содержит функции добавления нового будильника, генерации списка будильников, отмены будильника. При нажатии кнопки добавления будильника, запускается функция для показа MaterialTimePicker, с помощью которого пользователь может выбрать необходимое время.

Настройка всего приложения ведется с помощью постоянного хранилища для настроек – SharedPreferences (данные хранятся в виде пар ключ/значение). Для экрана «Настройки» написаны необходимые функции для работы с SharedPreferences: функция для добавления, редактирования и удаления настроек. Добавление производится моментально после выбора пользователем каких-либо значений. С помощью компонента Switch в настройки записывается значение типа Boolean с соответствующим ключом. При помощи элемента NumberPicker (элемент для выбора числа) в SharedPreferences записываются значение типа Int и ключ настройки.

Экран обучения содержит функции для генерации слайдов с полезными советами по использованию приложения. Слайды можно пролистывать и просматривать. Если пользователю не требуется обучение, то он может его пропустить.

В приложении присутствует виброотклик на некоторые действия пользователя (например, если пользователь нажмет на кнопку, то устройство будет вибрировать). Функции вибрации описаны в отдельном классе и

реализованы с помощью системного сервиса вибрации. Пользователь может отключить данную функцию в настройках приложения.

Функции анимации в приложении (например, анимация перехода между экранами) занесены в отдельный класс. Анимацию можно отключить в настройках приложения.

					НАТКиГ.210300.43.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		17

4 ТЕСТИРОВАНИЕ

4.1 Протокол тестирования дизайна приложения

Тестирование дизайна проводится на устройстве Honor 8x (API 29) с разрешением экрана 2340×1080 и BQ-5059 (API 24) с разрешением экрана 1280×720. Необходимо проверить корректное отображение элементов приложения на устройстве Honor 8X с минимальным (рисунок 14) и наибольшим системным масштабом интерфейса (рисунок 15). Аналогичная проверка отображения проведена на устройстве BQ-5059. Результаты представлены на рисунках 16 и 17.

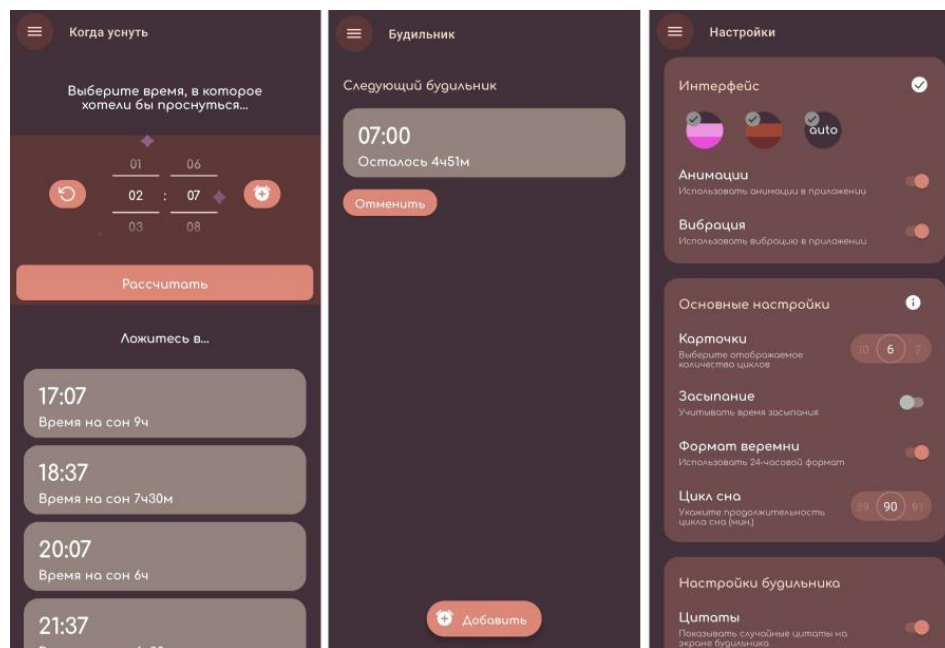


Рисунок 14 – Тестирование дизайна с минимальным масштабом на Honor 8X

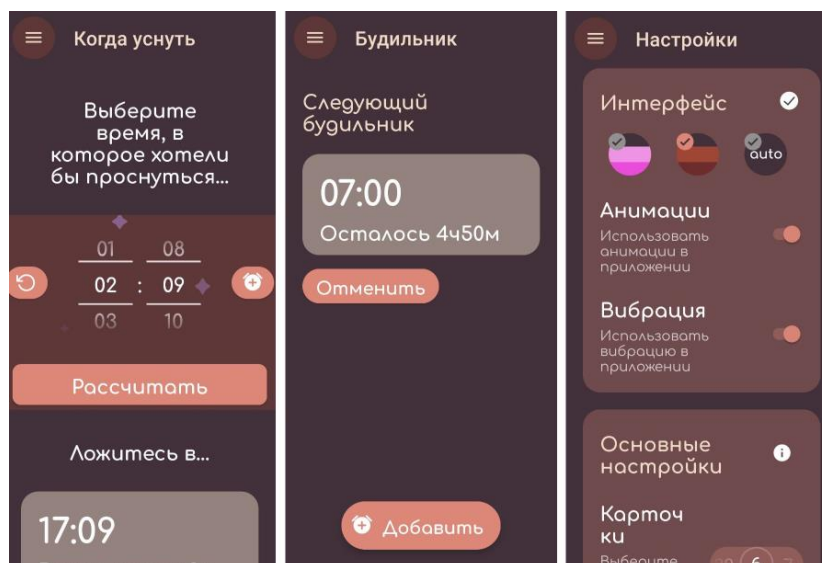


Рисунок 15 – Тестирование дизайна с наибольшим масштабом на Honor 8X

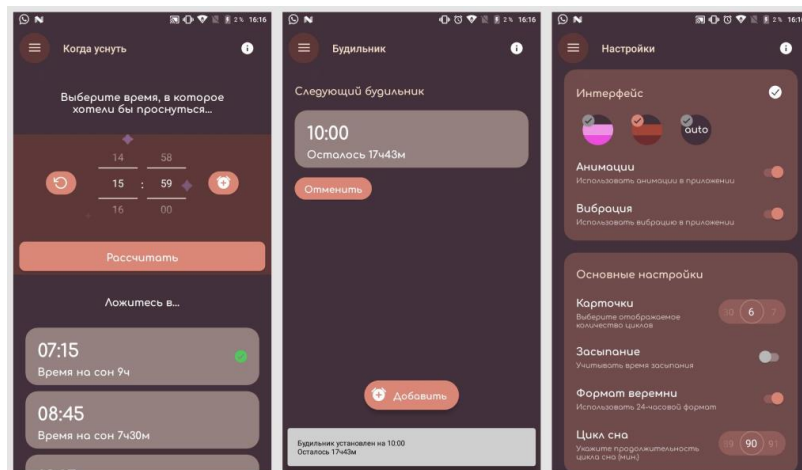


Рисунок 16 – Тестирование дизайна с наименьшим масштабом на BQ-5059

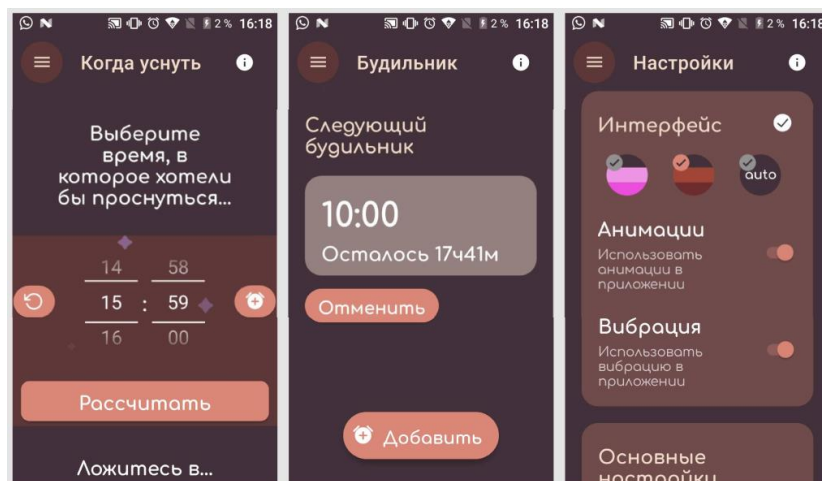


Рисунок 17 – Тестирование дизайна с наибольшим масштабом на BQ-5059

Как видно на рисунках выше, элементы на экране внутри приложения отображаются корректно вне зависимости от системного масштаба интерфейса и разрешения экрана.

4.2 Протокол тестирования функционала приложения

В целях проверки правильности работы некоторых функций, разрабатываются Unit-тесты. Ниже представлена реализация тестов функций класса, предназначенного для работы со временем.

```
@Test
fun calcRemainingTimeMinute() {
    //тестирование функции расчета оставшегося времени
    val calendar = Calendar.getInstance()
    calendar.add(Calendar.DATE, 1)
    Assert.assertEquals(
        TimeUtils.calcRemainingTimeMinute(
            date = Date(calendar.timeInMillis),
            context = mock(Context::class.java)
        ),
        "24null"
    )
}

@Test
fun getFormatTime() {
    //тестирование функции форматирования минут в строку
    val calendar = Calendar.getInstance()
    calendar.set(2022, 5, 8, 14, 15, 12)
    Assert.assertEquals(
        TimeUtils.getFormatTime(
            minute = calendar.get(Calendar.MINUTE).toLong(),
            context = mock(Context::class.java)
        ), "15null"
    )
}
```

Аналогичные тесты проведены для функций работы с будильником.

Все реализованные тесты для функционала приложения выполняются корректно.

ЗАКЛЮЧЕНИЕ

В ходе изучения предметной области выявлено, что проблема, связанная с плохим пробуждением, для многих людей является актуальной. Также преобладающим параметром при расчете оптимального времени в калькуляторе сна является экономия времени. Именно на это акцентировано внимание при разработке мобильного приложения.

Основные экраны приложения спроектированы таким образом, чтобы не нагружать пользователя лишней информацией. Весь интерфейс реализован таким образом, чтобы пользователь достиг своей цели, выполнив минимальное количество действий. В ходе разработки добавлены некоторые вспомогательные полезные функции, такие как: установка будильника, настройка приложения.

Для реализации возможности изменения настроек приложения используется внутреннее хранилище данных SharedPreference.

В целях проверки корректности работы основных функций приложения проведены Unit-тесты. Все разработанные функции успешно прошли тестирование.

Таким образом поставленная цель курсовой работы достигнута.

БИБЛИОГРАФИЯ

1 ВЦИОМ [Электронный ресурс]: Исследования в области сна у Россиян – Режим доступа: <https://wciom.ru/analytical-reviews/analiticheskii-obzor/spim-i-vidim-kak-spyat-v-rossii>

2 Developers Android [Электронный ресурс]: Полная документация по Android – Режим доступа: <https://developer.android.com/>

3 Kotlinlang [Электронный ресурс]: Документация по языку программирования Kotlin – Режим доступа: <https://kotlinlang.ru/>

4 Material Design [Электронный ресурс]: Рекомендации от Google по построению дизайна приложения, верстке и использованию тех или иных компонентов – Режим доступа: <https://material.io/components?platform=android>

5 Metanit [Электронный ресурс]: Документация по языку программирования Kotlin – Режим доступа: <https://metanit.com/kotlin/tutorial/>

6 StackOverflow [Электронный ресурс]: Система вопросов и ответов о программировании – Режим доступа: <https://stackoverflow.com/>

7 Wikipedia [Электронный ресурс]: Статья про сон – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BD>

Приложение А

Техническое задание

ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку мобильного приложения «Разработка мобильного приложения «Калькулятор сна», используемого для расчета оптимального времени для пробуждения и засыпания.

Наименование программы: «Sleepy». Далее по тексту Приложение.

Краткая характеристика области применения: мобильное приложение предоставляет возможность быстрого расчета оптимального времени для пробуждения и засыпания, а также установки будильника.

1 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основание для проведения разработки является Приказ № Уч-041/4 от 17.03.2022 года.

Наименование темы разработки – «Разработка мобильного приложения «Калькулятор сна».

2 НАЗНАЧЕНИЯ ДЛЯ РАЗРАБОТКИ

Функциональным назначением приложения является обеспечение удобного и эффективного интерфейса для пользователя, а также автоматизации всех необходимых расчетов.

3 ТРЕБОВАНИЕ К ПРИЛОЖЕНИЮ

3.1 Требования к функциональным характеристикам

					НАТКиГ.210300.43.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		23

Система должна обеспечивать возможность выполнения нижеперечисленных функций, описанных в таблице А.1.

Таблица А.1 – Выполняемые функции приложения

Номер	Функция
1	Расчет оптимального времени для пробуждения
2	Расчет оптимального времени для засыпания
3	Установка будильника
4	Добавление, редактирование и очистка пользовательский настроек приложения
5	Смена темы приложения

3.2 Требования к надёжности

Требования к надёжности не предоставляются

3.3 Условия эксплуатации

Пользователь должен иметь практические навыки использования мобильного устройства под управлением операционной системы Android.

3.4 Требования к составу и параметрам технических средств

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android не ниже версии 7.0.

3.5 Требования к информационной и программной совместимости

Разработка приложения ведется на языке программирования Kotlin.

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android с версией SDK не ниже 24.

3.6 Требования к защите информации

Требования не представляются.

3.7 Требования к маркировке и упаковке

Требования не представляются.

3.8 Специальные требования

Специальные требования не предоставляются.

4 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации должен включать в себя:

- техническое задание;
- пояснительная записка.

5 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

6 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Таблица А.2 – Стадии разработки

Этапы разработки КП	Срок выполнения	Отчетность
Определение цели и задач, объекта и предмета исследования	26.03.2022	Пояснительная записка
Описание предметной области	02.04.2022	Пояснительная записка

Продолжение таблицы А.2 – Стадии разработки

Выбор технологии, языка и среды программирования	09.04.2022	Пояснительная записка
Оформление технического задания	16.04.2022	Пояснительная записка
Проектирование UI/UX дизайна	23.04.2022	Пояснительная записка
Разработка мобильного приложения	30.04.2022	Пояснительная записка
Отладка и тестирование приложения	14.05.2022	Пояснительная записка
Оформление документации	21.05.2022	
Защита	26.05.2022	

7 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Виды испытаний – защита проекта.

Общие требования к приёмке:

- техническое задание;
- пояснительная записка;
- бланк задания;
- программный продукт.