



министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
БАКАЛАВРА НА ТЕМУ:
Программное приложение для подбора одежды в
соответствии с погодой

Студент

ИУ6-84Б

(Группа)

(Подпись, дата)

С.П. Пантелеев

(И.О. Фамилия)

Руководитель

(Подпись, дата)

М.В. Широкова

(И.О. Фамилия)

Нормоконтролер

(Подпись, дата)

А.М. Суоров

(И.О. Фамилия)

2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ6

 А.В. Пролетарский

« 4 » декабря 2022 г.

З А Д А Н И Е
на выполнение выпускной квалификационной работы бакалавра

Студент группы ИУ6-84Б

Пантелеев Сергей Павлович

(Фамилия, имя, отчество)

Тема квалификационной работы Программное приложение для подбора одежды в соответствии с погодой

Источник тематики (НИР кафедры, заказ организаций и т.п.)

НИР кафедры

Тема квалификационной работы утверждена распоряжением по факультету ИУ № 03.02.01-04.03/20 от « 9 » декабря 2021 г.

Часть I. Исследовательская

Выполнение анализа существующих систем подбора одежды в зависимости от погодных условий. Выявление принципов работы. Определение основных классификаций одежды. Исследование вариантов подбора одежды в зависимости от погоды. Определение основных функций разрабатываемого приложения

Часть 2. Конструкторская

Выбор языка и среды разработки. Проектирование архитектуры системы. Определение способов хранения информации. Проектирование и разработка интерфейса. Реализация диалогов с пользователем. Разработка алгоритма подбора одежды в зависимости от параметров. Реализация логики работы системы. Выполнение тестирования приложения

Часть 3. Технологическая

Разработка технологии тестирования приложения

Оформление квалификационной работы:

Расчетно-пояснительная записка на 55–65 листах формата А4.


Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

1. Схема структурная информационной системы – 1 лист формата А1;
2. Диаграмма вариантов использования – 1 лист формата А1;
3. Формы интерфейса – 1 лист формата А1;
4. Граф состояния интерфейса – 1 лист формата А1;
5. Схема алгоритма подбора одежды в зависимости от погодных условий – 1 лист формата А1;
6. Диаграмма классов уровня реализации – 1 лист формата А1;


Дата выдачи задания « 1 » сентября 2021 г.

В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до « 1 » июня 2022 г.

Руководитель квалификационной работы

 02.03.2022 М.В. Широкова
(Подпись, дата) (И.О. Фамилия)

Студент

 02.03.2022 С.П. Пантелеев
(Подпись, дата) (И.О. Фамилия)

Примечание: 1. Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

УТВЕРЖДАЮ

КАФЕДРА Компьютерные системы и сети

Заведующий кафедрой ИУ6

ГРУППА ИУ6-84Б

А.В. Пролетарский

« 3 » *дека* 2022 г.

КАЛЕНДАРНЫЙ ПЛАН

выполнения выпускной квалификационной работы бакалавра

студента: Пантелеев Сергей Павлович

(фамилия, имя, отчество)

Тема квалификационной работы Программное приложение для подбора одежды в соответствии с погодой

| № п.п. | Наименование этапов выпускной квалификационной работы | Сроки выполнения этапов | | Отметка о выполнении | |
|-----------|---|---|-------------------|----------------------|-------------------|
| | | план | факт | Должность | ФИО, подпись |
| 1. | Задание на выполнение работы. Формулирование проблемы, цели и задач работы | <u>09.2021</u> <small>Планируемая дата</small> | <i>27.09.21</i> | Руководитель ВКР | М. В. Широкова |
| 2. | 1 часть <u>Исследовательская</u> | <u>12.2021</u> <small>Планируемая дата</small> | <i>20.12.21</i> | Руководитель ВКР | М. В. Широкова |
| 3. | Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач | <u>02.2022</u> <small>Планируемая дата</small> | <i>03.03.2022</i> | Заведующий кафедрой | А.В. Пролетарский |
| 4. | 2 часть <u>Конструкторская</u> | <u>05.2022</u> <small>Планируемая дата</small> | | Руководитель ВКР | М. В. Широкова |
| 5. | 3 часть <u>Технологическая</u> | <u>05.2022</u> <small>Планируемая дата</small> | | Руководитель ВКР | М. В. Широкова |
| 6. | 1-я редакция работы | <u>05.2022</u> <small>Планируемая дата</small> | | Руководитель ВКР | М. В. Широкова |
| 7. | Подготовка доклада и презентации | <u>06.2022</u> <small>Планируемая дата</small> | | Руководитель ВКР | М. В. Широкова |
| 8. | Заключение руководителя | <u>06.2022</u> <small>Планируемая дата</small> | | Руководитель ВКР | М. В. Широкова |
| 9. | Нормоконтроль | <u>06.2022</u> <small>Планируемая дата</small> | | Нормоконтролер | |
| 10. | Внешняя рецензия | <u>06.2022</u> <small>Планируемая дата</small> | | Руководитель ВКР | М. В. Широкова |
| 11. | Защита работы на ГЭК | <u>06.2022</u> <small>Планируемая дата</small> | | Руководитель ВКР | М. В. Широкова |

Студент *Пантелеев* 2.03.2022 *Сергей Павлович* Руководитель работы *Широкова* 2.03.2022 *М.В. Широкова*
(подпись, дата) (И.О. Фамилия) (подпись, дата) (И.О. Фамилия)

АННОТАЦИЯ

Расчетно-пояснительная записка выпускной квалификационной работы бакалавра посвящена процессу проектирования и разработки приложения по подбору одежды в соответствии с погодой. Для реализации приложения был проведен анализ систем, которые помогают пользователям подбирать одежду в соответствии с погодой. Помимо этого, был произведен анализ типов существующей одежды и особенностей восприятия температуры, а также особенности подбора одежды в зависимости от температуры. В результате исследования предметной области были выявлены требования к программному приложению, которое позволяет пользователю получить информацию об одежде, которую можно одеть в соответствии с погодными условиями.

На основе проведенных исследований было спроектировано и реализовано программное приложение для подбора одежды в соответствии с погодой.

ANNOTATION

Calculation and explanation summary of graduation qualification work of the bachelor is devoted to process of designing and development of application for choosing clothes in accordance with weather. For application realization analysis of systems which help users to chose clothes in accordance with weather is conducted. Also types of clothes, feature of perception of temperature and peculiarities of choosing clothes in accordance with weather were analyzed. As a result of research in a subject domain requirement for program application were identified.

Based on conducted research program application for choosing clothes in accordance with weather was developed.

РЕФЕРАТ

Расчетно-пояснительная записка 112 с., 28 рис., 10 табл., 36 источн., 5 прил.

ПОДБОР ОДЕЖДЫ, ГАРДЕРОБ, ПОГОДА, ТЕПЛООЩУЩЕНИЕ, ТЕМПЕРАТУРА

Объектом разработки является программное приложение для подбора одежды в соответствии с погодой.

Целью данной работы является приложение, которое предоставит пользователю список подобранной одежды, основываясь на гардеробе пользователя, его теплоощущении и температуре на улице.

В ходе выполнения работы были решены следующие задачи:

- исследование предметной области;
- сравнение аналогов;
- проектирование и разработка диаграмм;
- реализация программного приложения.

В результате разработки было спроектировано, реализовано и протестировано приложение, предлагающее пользователю одежду, которая подобрана в зависимости от температурных условий и теплоощущения пользователя. Пользователями системы будут являться люди, которые имеют трудности с решением, что одеть при выходе на улицу.

СОДЕРЖАНИЕ

| | |
|--|----|
| ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ..... | 6 |
| ВВЕДЕНИЕ..... | 7 |
| 1 Анализ предметной области и уточнение требований к программному продукту..... | 8 |
| 1.1 Анализ предметной области и формирование требований к системе..... | 8 |
| 1.2 Анализ существующих систем подбора одежды в зависимости от погоды... | 8 |
| 1.3 Анализ классификаций одежды, видов одежды и особенностей ощущения температуры человеком..... | 15 |
| 1.4 Анализ существующих рекомендаций по подбору одежды в зависимости от погоды..... | 18 |
| 1.5 Результаты исследования..... | 20 |
| 2 Разработка приложения по подбору одежды в зависимости от погоды..... | 22 |
| 2.1 Анализ требований и выбор основных проектных решений..... | 22 |
| 2.1.1 Выбор технологии и языка программирования..... | 22 |
| 2.1.2 Выбор архитектуры и подхода разработки | 22 |
| 2.2 Разработка диаграммы вариантов использования..... | 24 |
| 2.3 Построение концептуальной модели предметной области..... | 26 |
| 2.4 Построение схемы структурной информационной системы..... | 29 |
| 2.5 Разработка интерфейса приложения..... | 29 |
| 2.5.1 Построение диаграммы состояний интерфейса..... | 29 |
| 2.5.2 Разработка форм интерфейса..... | 32 |
| 2.5.3 Разработка графов диалогов..... | 38 |
| 2.6 Выбор методов хранения данных приложения..... | 39 |
| 2.7 Разработка алгоритма подбора одежды..... | 41 |
| 2.8 Разработка диаграммы уровня реализации..... | 45 |
| 2.9 Разработка диаграммы компоновки..... | 47 |
| 2.10 Комплексное тестирование программного продукта..... | 49 |
| 2.10.1 Функциональное тестирование..... | 49 |
| 2.10.2 Автоматическое тестирование..... | 52 |

| | |
|--|-----|
| 2.10.3 Оценочное тестирование..... | 53 |
| 3 Разработка технологии тестирования приложения..... | 55 |
| ЗАКЛЮЧЕНИЕ..... | 58 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 59 |
| Приложение А. Техническое задание..... | 63 |
| Приложение Б. Руководство пользователя..... | 72 |
| Приложение В. Графический материал..... | 89 |
| Приложение Г. Фрагмент исходного кода..... | 96 |
| Приложение Д. Таблицы тестов функционального тестирования..... | 102 |

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Android – операционная система на ядре Linux

APK файл – итоговый исполняемый файл в системе android

MacOS и iOS – операционные системы, которые стоят в продукции от корпорации Apple

Windows – операционная система для компьютеров от корпорации Microsoft

Баг – недоработка в программном коде

БД – база данных

Интерфейс – графическое представление приложения пользователю

ООП – объектно-ориентированное программирование

ПК – персональный компьютер

ПП – программный продукт

Теплоощущение – то, как в целом организм реагирует на температуру, например, ему часто холодно, или наоборот, постоянно жарко

ТЗ – техническое задание

Фреймворк – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта

ВВЕДЕНИЕ

В настоящее время существует проблема по подбору одежды в соответствии с погодой на улице. Видя погоду на улице, не все интуитивно могут определить температуру, из-за этого многие пользуются сервисами прогноза погоды [1]. Однако, даже зная прогноз погоды, не каждый может определить, что одеть, для того чтобы, например, не заболеть, когда холодно, или наоборот, чтобы не было жарко, когда тепло. Разрабатываемое приложение позволит решить проблему по подбору одежды в зависимости от погоды путем выдачи рекомендаций пользователю, основываясь на его теплоощущении, пола, температуре на улице и имеющейся у пользователя одежды.

На данный момент существует множество аналогов приложений для подбора одежды. Однако, многие из них имеют только англоязычный интерфейс и у них отсутствуют возможности добавления собственной одежды и выбора теплоощущения. А также не все предоставляют выбор пола и автоматический подбор одежды.

Главными плюсами разрабатываемого приложения являются наличие как шаблонной одежды, так и возможность добавление новой одежды пользователем самостоятельно и русифицированный интерфейс. Также предусмотрена возможность выбора показателя теплоощущения. Помимо этого, в приложении имеется возможность выбора пола пользователя, который учитывается при подборе одежды.

Поэтому разработка приложения по подбору одежды в соответствии с погодой является актуальной.

1 Анализ предметной области и уточнение требований к программному продукту

1.1 Анализ предметной области и формирование требований к системе

При подборе одежды в соответствии с погодными условиями можно опираться на несколько параметров, таких как температура на улице и гардероб.

Температуру возможно определить по местоположению устройства пользователя, поэтому необходимо обеспечить доступ системе к местонахождению устройства, что позволит определить показатель температуры на улице.

Для подбора одежды приложению помимо температуры необходимо владеть информацией о доступной одежде пользователя. Однако, трудно учесть все существующие виды одежды, поэтому приложение должно содержать в себе шаблонную одежду, которая есть у многих пользователей. С помощью представленного гардероба приложение сможет подбирать одежду для пользователя в соответствии с погодой. При этом, если в шаблонном гардеробе отсутствует какой-то вид одежды, необходимо предусмотреть возможность добавления пользователем своего вида одежды.

Исходя из вышесказанного можно сделать вывод, что минимальными требованиями к системе являются:

- определение местоположения,
- определение температуры,
- наличие набора шаблонной одежды,
- наличие возможности добавления новой одежды.

1.2 Анализ существующих систем подбора одежды в зависимости от погоды

Для выявления достоинств и недостатков существующих приложений, необходимо провести их сравнительный анализ, на основе которого будут сформулированы основные функции разрабатываемого приложения.

Исходя из технического задания было решено рассматривать приложения, разработанные на ОС android.

Одними из самых популярных приложений в playMarket, количество скачиваний которых превышает 10000 раз, являются [2]:

- “Getwardrobe – гардероб, стилист, одежда, мода” от 3Botinka [3],
- “combyne – Outfit ideas & outfit creation” от combine GmbH [4],
- “Wear This – Outfits by Weather” от This Technologies [5],
- “Weather Awear – Dress for the weather” от io Jeanne [6],
- “Weatherproof – Weather & Clothes” от Dr. Alexaner Rieger [7],
- “What To Wear Today” от Diecke Apps [8].

Рассмотрим приведенные выше аналоги.

Getwardrobe – гардероб, стилист, одежда, мода

Приложение “Getwardrobe – гардероб, стилист, одежда, мода” позволяет создать собственный виртуальный гардероб, фотографируя одежду и формируя из неё образы. Интерфейс приложения можно увидеть на рисунке 1.

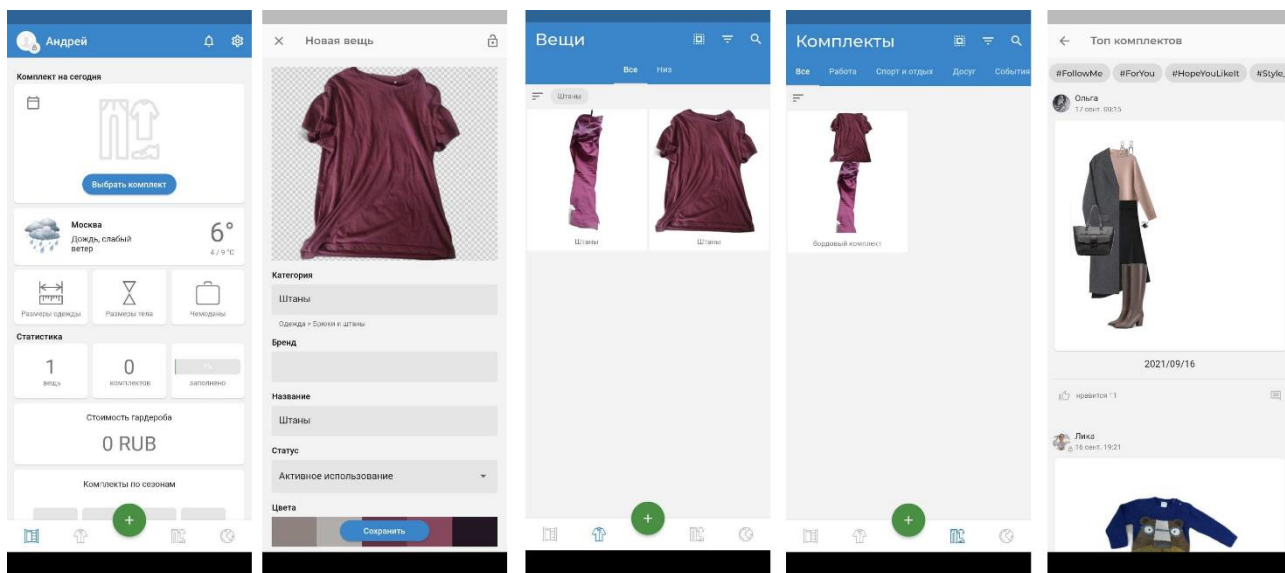


Рисунок 1 – Интерфейсы приложения “Getwardrobe – гардероб, стилист, одежда, мода”

Приложение имеет следующие функции:

- создание собственных образов,
- добавление одежды,
- просмотр чужих образов,
- деление на виды деятельности.

Из минусов можно выделить:

- отсутствие заранее загруженной шаблонной одежды,
- отсутствие выбора гендера,
- много платного контента.

Wear This – Outfits by Weather

Приложение “Wear This – Outfits by Weather” предлагает одежду в соответствии с температурой на улице. ПП определяет температуру на ближайшие 24 часа и предлагает несколько вариаций того, что можно одеть. Картинки одежды интуитивно понятны. Интерфейс приложения можно увидеть на рисунке 2.

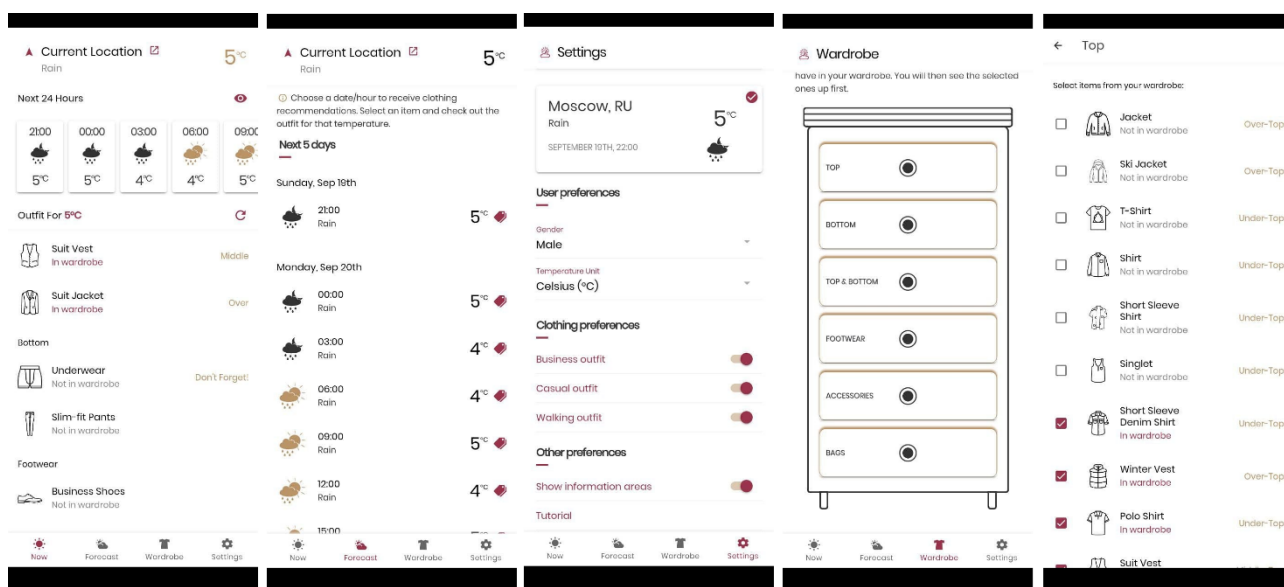


Рисунок 2 – Интерфейсы приложения “Wear This – Outfits by Weather”

Приложение имеет следующие функции:

- одежда показана текстом и картинками,
- погода определяется на ближайшие 5 дней,
- большой гардероб заранее загруженной одежды,
- выбор шкалы температуры,
- можно выбирать одежду, которая будет рекомендоваться,
- выбор деятельности (обычный, рабочий и прогулочный).

Из недостатков можно выделить отсутствие возможности добавления собственной одежды и смены языка интерфейса.

Combyne – Outfit ideas & outfit creation

Приложение “Combyne – Outfit ideas & outfit creation” дает возможность создать собственную библиотеку образов, смотреть и обсуждать чужие “луки”. Помимо этого, в программе есть чат и возможность подписаться на другого человека, что делает ПП похожим на социальную сеть с кругом интересом в виде одежды. Интерфейс приложения можно увидеть на рисунке 3.

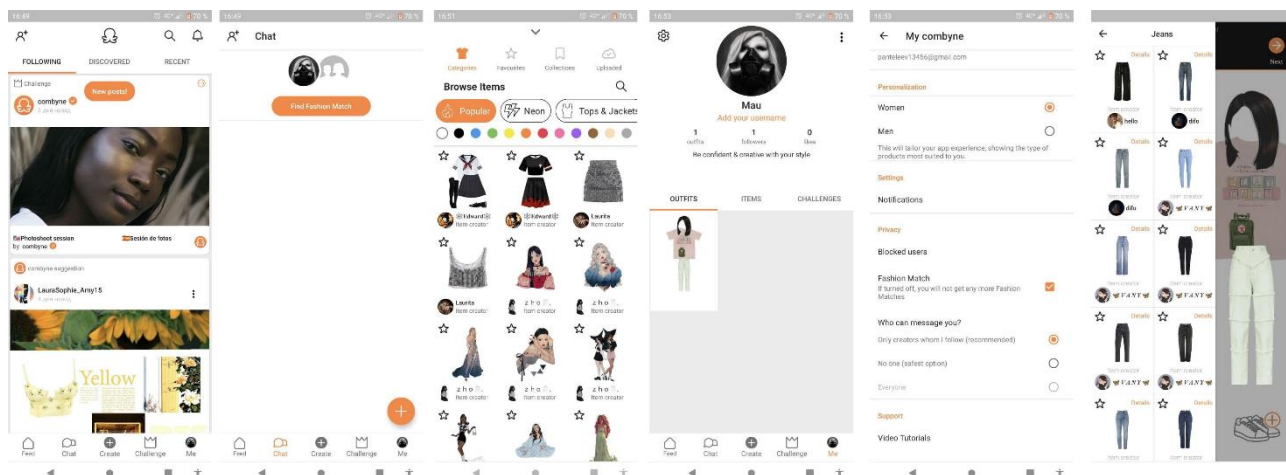


Рисунок 3 – Интерфейсы приложения “Combyne – Outfit ideas & outfit creation”

Программа имеет следующие функции (перечислены не абсолютно все, а главное или отличительные черты):

- просмотр чужих образов,
- создание собственных образов,
- возможность добавлять собственную одежду,
- большой ассортимент заранее заложенной одежды,
- создание подборок совместно с другими пользователями,
- выбор гендера,
- чат для общения.

Weather Awear – Dress for the weather

Приложение “Weather Awear – Dress for the weather” определяет, что одеть в соответствии с температурой на улице. Интерфейс приложения можно увидеть на рисунке 4.

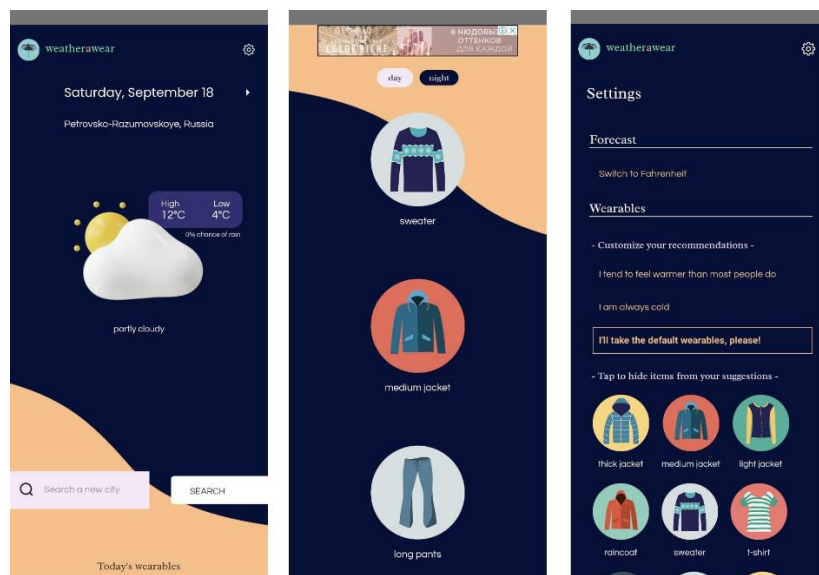


Рисунок 4 – Интерфейсы приложения “Weather Awear – Dress for the weather”

Программа определяет максимальную и минимальную температуру в течение дня и на основе этого подбирает одежду. Приложение имеет следующие функции:

- предлагает одежду на день и на ночь,
- одежда показана текстом и картинками,
- показывает погоду на ближайшие два дня,
- можно выбрать шкалу температуры между цельсием и фаренгейтом,
- можно выбирать “теплоощущение” своего тела (пример: “мне всегда холодно”),
- можно скрывать некоторую одежду, чтобы она больше не появлялась.

Weatherproof – Weather & Clothes

Приложение “Weatherproof – Weather & Clothes?” предлагает, что одеть в соответствии с погодными условиями. Интерфейс приложения можно увидеть на рисунке 5. Программа определяет погоду на ближайший день, например, если сейчас вечер, то будет температура до следующего вечера: текущая и утренняя, дневная и вечерняя следующего дня.

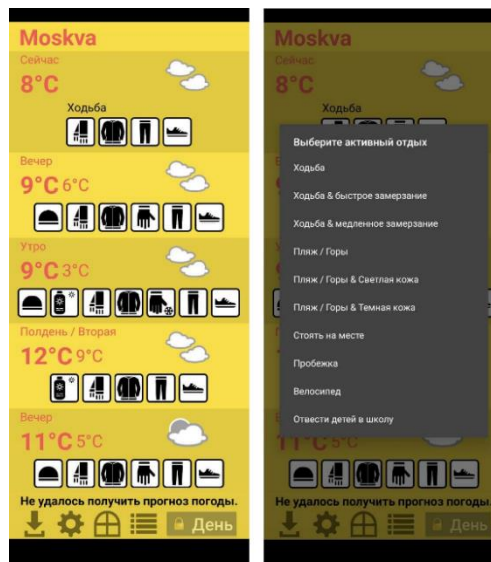


Рисунок 5 – Интерфейсы приложения “Weatherproof – Weather & Clothes”

Приложение имеет следующие функции:

- показывает, что одеть на разные периоды дня,
- одежда показана картинками,
- подбор одежды под разные виды деятельности.

Из минусов можно выделить:

- картинок одежды мало (и они очень абстрактные) и не всегда понятно, что это,
- много платного функционала.

What To Wear Today

Приложение “What To Wear Today” определяет, что одеть в соответствии с погодными условиями. Интерфейс приложения можно увидеть на рисунке 6.

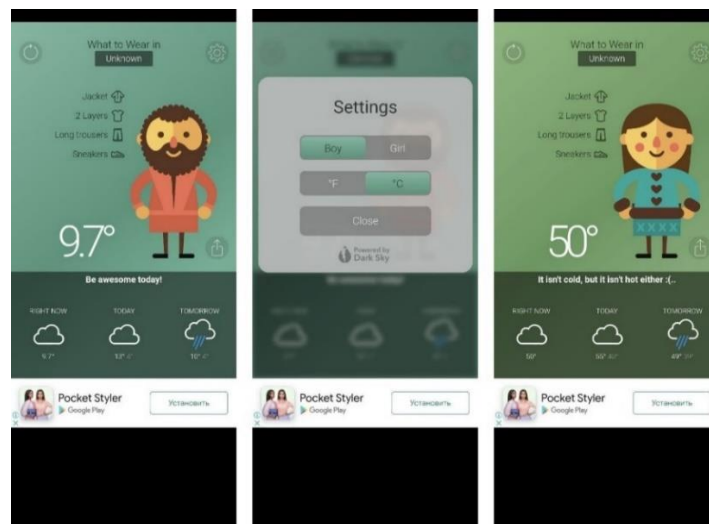


Рисунок 6 – Интерфейс приложения “What To Wear Today”

Программа определяет текущую, среднюю за день и завтрашнюю температуру на улице в соответствии с местоположением телефона. Имеет следующие функции:

- выбор гендера,
- выбор шкалы температуры,
- показывает, что надеть текстом и картинкой.

Из минусов можно выделить:

- не подстраивается под формат экрана,
- однотипный подбор одежды,
- визуально интуитивно не понятно.

Проанализировав приведенные приложения, можно сделать вывод, что программ по подбору одежды не пару штук и они качественно сделаны. Но не все учитывают фактор погоды, а просто предлагают создать свою виртуальную гардеробную. Взяв этот аспект, что приложение при подборе одежды должно опираться на погоду, выделим критерии качества:

- правильная геолокация и погода – так как приложение будет подбирать одежду по погоде, необходимо, чтобы оно правильно определяло геолокацию пользователя и температуру на улице;
- выбор гендера – пользователю будет предлагаться только соответствующая гендеру одежда, что упростит использование;
- теплорегуляция тела – делает подбор одежды более гибкий, так как у людей может быть разное теплоощущение и кому-то может быть все время холоднее или наоборот, жарче, чем другим;
- шаблонный гардероб – чтобы пользователь мог сразу посмотреть, что ему предложит приложение, не добавляя всю свою одежду в виртуальный гардероб;
- добавление собственной одежды – одежды сейчас очень много и не вся может быть учтена разработчиком, поэтому пользователь может добавить что-то свое, чего нет в БД программы;

- автоматический подбор одежды – пользователю не нужно формировать свои образы. Программа сама предложит, что лучше одеть по погоде;
- русифицированный интерфейс – чтобы все русскоязычные пользователи могли комфортно пользоваться приложением.

По выделенным критериям проведен сравнительный анализ приложений, который продемонстрирован в таблице 1.

Таблица 1 – Сравнительный анализ приложений по подбору одежды

| Критерий \ Приложение | Правильная геолокация и погода | Выбор гендера | Тепло-регуляция тела | Шаблонный гардероб | Добавление собственной одежды | Автоматический подбор одежды | Русификация |
|---|--------------------------------|---------------|----------------------|--------------------|-------------------------------|------------------------------|-------------|
| Getwardrobe – гардероб, стилист, одежда, мода | + | - | - | - | + | - | + |
| Wear This – Outfits by Weather | + | + | - | + | - | + | - |
| Combyne – Outfit ideas & outfit creation | - | + | - | + | - | - | - |
| Weather Awear – Dress for the weather | + | - | + | + | - | + | - |
| Weatherproof – Weather & Clothes | + | - | - | - | - | + | + |
| What to wear? | + | + | - | - | - | + | - |

Проведя сравнение приложений по выделенным критериям, можно сделать вывод, что из рассмотренных приложений нет таких, которые учитывали бы все параметры по подбору одежды в соответствии с погодой. Также не все программы переведены на русский, что может вызвать проблемы у русскоязычных пользователей.

1.3 Анализ классификаций одежды, видов одежды и особенностей ощущения температуры человеком

В современном мире множество стилей одежды, но их все можно разделить на конкретные категории по частям тела:

- голова – головные уборы,
- торс – майки, кофты, куртки, жилетки,
- ноги – штаны, юбки, обувь.

На голову могут быть одеты только аксессуары (например, очки) и головные уборы, но на ощущение температуры влияют только головные уборы, поэтому рассматриваться будут только головные уборы.

На торс можно одеть много различной одежды (например майка, футболка, пиджак, плащ и т.д.), которые делятся на три уровня:

- 1) майки, футболки (одевается на голый торс);
- 2) кофты, свитера (теплая одежда);
- 3) пальто, куртка (верхняя одежда).

На ноги можно одеть только штаны (шорты, юбки и т.д.) и обувь, поэтому можно выделить две классификации:

- штаны и юбки,
- обувь.

Для отбора шаблонной одежды были рассмотрены несколько статей [9][10][11][12][13] и интернет-магазинов [14][15][16], на основе которых была определена самая распространённая одежда.

Виды одежды, классифицирующиеся как головные уборы, представлены на рисунке 7.

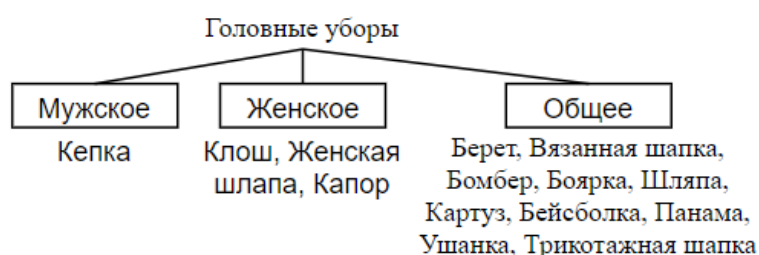


Рисунок 7 – Головные уборы

Виды одежды, классифицирующиеся как верхняя одежда, представлены на рисунке 8.

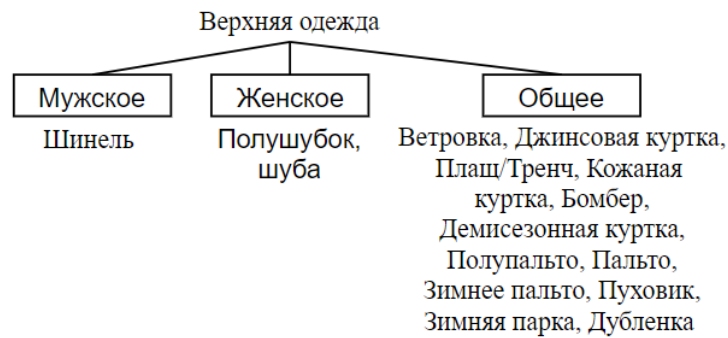


Рисунок 8 – Верхняя одежда

Виды одежды, классифицирующиеся как кофты и свитера, представлены на рисунке 9.

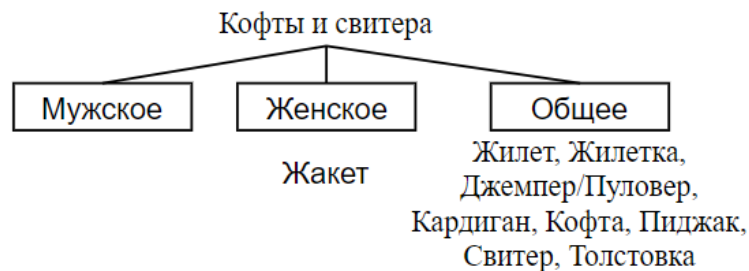


Рисунок 9 – Кофты и свитера

Виды одежды, классифицирующиеся как майки и футболки, представлены на рисунке 10.



Рисунок 10 – Майки и футболки

Виды одежды, классифицирующиеся как штаны и юбки, представлены на рисунке 11.

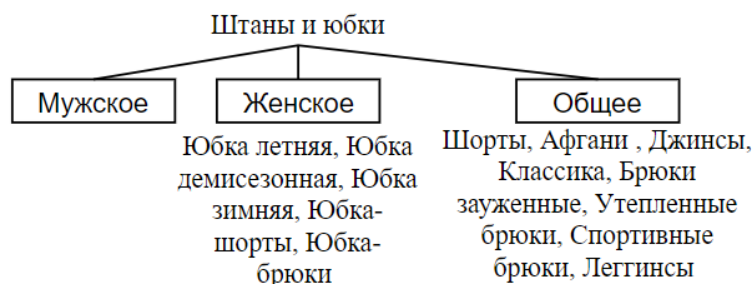


Рисунок 11 – Штаны и юбки

Виды одежды, классифицирующиеся как обувь, представлены на рисунке 12.

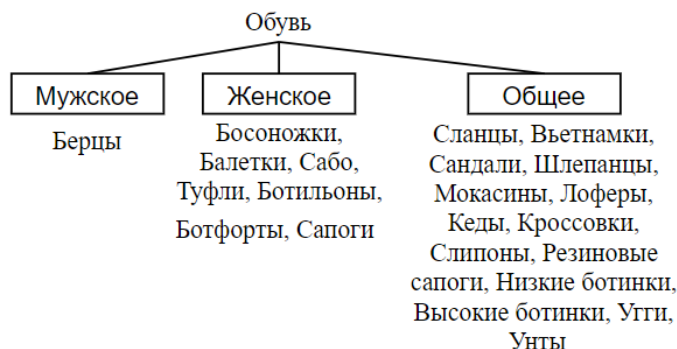


Рисунок 12 – Обувь

Не все люди ощущают температуру одинаково. Например, кто-то ходит в шапке до поздней весны, а кто-то ходит без шапки и расстегнутый зимой. Необходимо учитывать предпочтения людей по их температурным ощущениям. В ходе анализа были выявлены три настройки температуры:

- всегда холодно – температурный порог у человека выше, чем показатель на одежде;
- нормально – когда для человека температурный показатель одежды соответствует действительности;
- всегда жарко – температурный порог у человека ниже, чем показатель на одежде.

1.4 Анализ существующих рекомендаций по подбору одежды в зависимости от погоды

Для определения температурных показателей одежды были исследованы статьи, которые рассказывают о том, для какой температуры подходит та или иная одежда [17][18][19].

Основываясь на рассмотренных статьях и температурных показателях одежды, рассмотренных ранее, была составлена таблица 2, в которой для каждой одежды указывается её температурный диапазон.

Таблица 2 – Температурные показатели для одежды

| Головные уборы | | |
|--------------------------|---|--|
| Мужское | Женское | Общее |
| Кепка (+5 < N) | Клош (-5 – +5), Женская шляпа (+5 < N), Капор (-10 – 0) | Берет (0 – +8), Вязанная шапка (-10 – -5), Бомбер (N < -10), Боярка (N < -10), Шляпа (+8 < N), Картуз (+10 < N), Бейсболка (+8 < N), Панама (+20 < N), Ушанка (N < -10), Трикотажная шапка (0 – +5) |
| Верхняя одежда | | |
| Мужское | Женское | Общее |
| Шинель (-20 – +5) | Полушубок (-15 – 0), Шуба (N < -5) | Ветровка (+12 – +18), Джинсовая куртка (+15 – +22), Плащ/Тренч (+15 – +20), Кожаная куртка (+5 – +18), Бомбер (+5 – +15), Демисезонная куртка (+5 – +16), Полупальто (+8 – +15), Пальто (+5 – +15), Зимнее пальто (-15 – +5), Пуховик (N < -5), Зимняя парка (-20 – 0), Дубленка (N < 0) |
| Кофты, свитера и прочее | | |
| Мужское | Женское | Общее |
| | Жакет (0 – +18) | Жилет (-15 – +15), Жилетка (-15 – +15), Джемпер/Пуловер (N < +18), Кардиган (N < +15), Кофта (N < +18), Пиджак (N < +20), Свитер (N < +10), Толстовка (N < +15) |
| Майки, футболки и прочее | | |
| Мужское | Женское | Общее |
| Логслив (любая) | Блузка (любая), Сарафан (+25 < N), Туника (+15 – +30), Платье летнее (+18 < N), Платье вечернее\коктейльное (любая), Платье демисезонное\зимнее (-15 – +15), Платье-толстовка (-15 – +15) | Майка (любая), Футболка (любая), Поло (любая), Рубашка (любая), Водолазка (-15 – +5) |
| Штаны, юбки и прочее | | |
| Мужское | Мужское | Общее |
| | Юбка летняя (+20 < N), Юбка демисезонная (+5 – +20), Юбка зимняя (-20 – +5), Юбка-шорты (+5 – +25), Юбка-брюки (+5 – +20) | Шорты (+24 < N), Афгани (+16 – +26), Джинсы (любая), Классика (любая), Брюки зауженные (0 – N), Утепленные брюки (N < -5), Спортивные брюки (любая), Леггинсы (+10 – +20) |

Продолжение таблицы 2

| Обувь | | |
|--------------------|--|---|
| Мужское | Женское | Общее |
| Берцы (-30 – +10), | Босоножки (+24 < N), Балетки (+20 < N), Сабо (+20 < N), Туфли (+15 < N), Ботильоны (+2 – +10), Ботфорты (-8 – +4), Сапоги (N < 0), | Сланцы (+24 < N), Вьетнамки (+24 < N), Сандали (+20 < N), Шлепанцы (+20 < N), Мокасины (+15 < +20), Лоферы (+12 < +20), Кеды (+5 < N), Кроссовки (0 < N), Слипсы (+15 < N), Резиновые сапоги (+5 < N), Низкие ботинки (+5 – +20), Высокие ботинки (-25 – +3), Угги (-30 – -5), Унты (N < -10) |

Буква “N” в таблице обозначает любую температуру, которая меньше или больше (в зависимости от знака) указанной.

1.5 Результаты исследования

Перед разработкой приложения по подбору одежды необходимо описать его основные принципы работы. На основе проведенного анализа существующих приложений, выделив их достоинства и недостатки, опишем концепцию ПП.

Разрабатываемое приложение предназначено для подбора одежды в соответствие с температурой (погодой) на улице.

При запуске приложение должно определять местонахождение пользователя и показывать температуру.

Далее приложение предлагает, что можно одеть в соответствии с погодой из шаблона обычной одежды, то есть той, которая является наиболее распространённой.

Также пользователю необходимо предоставить возможность подстроить приложение под себя, а именно:

- определить свой гендер, в связи с которым изменится ассортимент предлагаемой одежды;
- теплоощущения собственного тела, то есть пользователю постоянно холодно или наоборот, ему часто жарко;

— выбор шкалы измерения температуры – градус цельсия, кельвины или фаренгейт.

Необходимо предоставить выбор имеющейся одежды из той, что уже заложена в приложение. Это необходимо для того, чтобы программа не выдавала варианты надеть то, чего у пользователя нет в реальном гардеробе.

Ассортимент предлагаемой одежды должен быть обширен, но пользователь может иметь вид одежды, которого нет в БД. Следовательно, необходимо учесть вариант добавления вещи в БД приложения. При добавлении пользователь должен указать рекомендованную температуры.

Возможна перетасовка одежды, которую приложение предложило одеть.

Пример. Если температура “+20”, возможны варианты:

- Кепка + футболка + шорты,
- Рубашка + майка + джинсы,
- Рубашка + футболка + брюки.

Варианты могут варьироваться от выбранной пользователем имеющейся у него одежды.

Структурируя все вышеперечисленное, приложение будет иметь следующие функции:

- определение местоположения и температуры,
- выбор теплоощущения тела,
- выбор шкалы температуры,
- выбор шаблонной одежды,
- подбор одежды на день, в зависимости от погоды,
- возможность лично расширять гардероб,
- предложение нескольких вариантов одежды на день,
- русифицированный интерфейс.

2 Разработка приложения по подбору одежды в зависимости от погоды

2.1 Анализ требований и выбор основных проектных решений

2.1.1 Выбор технологии и языка программирования

В качестве технологии программирования было выбрано объектно-ориентированное программирование. ООП является основой всех современных приложений и имеет удобное и практическое применение. При использовании этого метода вся программа разбивается на объекты, с каждым из которых работа происходит по отдельности, что позволяет в будущем расширять программный продукт путем добавления новых объектов.

Для написания мобильных приложений существует множество языков программирования. Наиболее популярные – это Java, Kotlin [20] и Swift [21], которые специализируются на мобильной разработке. А также есть кроссплатформенные языки, которые тоже можно использовать, это Rust, Dart, Python (в связке с Kivy и BeeWare), Javascript (в связке с React Native) и C# (в связке с Xamarin) [22].

Приложение разрабатывается под мобильное устройство, следовательно, выбор языка будет осуществлять из тех, что специализируются на мобильной разработке, так как для них больше соответствующих библиотек и документации.

Для разрабатываемого приложения был выбран язык Java, так как он является наиболее подходящим вариантом. В отличие от Kotlin, который еще достаточно молодой язык, Java имеет больше документации и другой обучающей информации. Swift же не был выбран, так как этот язык используется для разработки приложения под систему MacOS и iOS, то есть только для устройств от корпорации Apple.

2.1.2 Выбор архитектуры и подхода разработки

Выбор архитектуры приложения важен при создании приложения. От этого будет зависеть то, как элементы кода (объекты, классы и т.д.) внутри приложения будут взаимодействовать между собой.

Для мобильной android разработки существуют несколько типов архитектур, но выделим и рассмотрим только основные. Большинство android приложений используют одну из следующих архитектур [23]:

- MVC,
- MVP,
- MVVM,
- MVI.

MVVM часто используется в больших приложениях, а её использование в разрабатываемом приложении только усложнит разработку.

MVC не выбрана, так как этот тип архитектуры вызывает большее количество зависимостей внутри приложения, что может вызывать “волновой” эффект ошибок при разработке или усовершенствовании приложения.

MVI архитектура имеет модель state, который хранит в себе состояние экрана и каждый раз создается заново, имея число состояний, что не нужно для разрабатываемого приложения. Также MVI достаточно нова, из-за чего по ней меньше документации и инструкций.

Учитывая принципы работы и небольшую масштабность разрабатываемого приложения, наиболее подходящей для разработки приложения по подбору одежды является MVP архитектура. Её визуальное представление показано на рисунке 13.

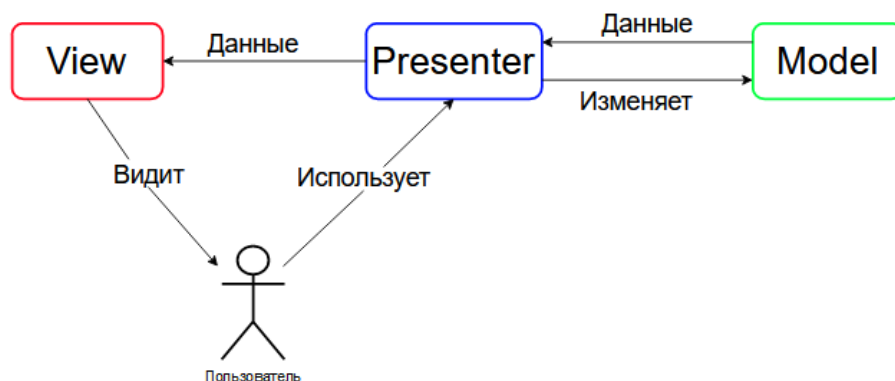


Рисунок 13 – MVP архитектура

MVP – Model-View-Presenter, архитектура описывает, как отделить UI от логики интерфейса (что происходит при взаимодействии с виджетами) и от данных (какие данные отображать на экране). Presenter является промежуточным

компонентом между моделью и представлением. Presenter содержит в себе логику управления представлением, то есть берет данные из модели, обрабатывает её и передает в представление [23][24][25].

В качестве жизненного цикла разработки была выбрана спиральная модель [26]. Этот метод позволяет в конце каждого цикла иметь работающий продукт, который можно продемонстрировать. Это дает возможность своевременно оценить и протестировать продукт, чтобы сразу вносить какие-либо правки и исправлять ошибки, не дожидаясь окончания разработки. Нахождение багов на каждом этапе позволяет избежать “волнового” исправления ошибок. А также этот метод позволяет детальнее подойти к каждому этапу разработки по отдельности.

Также при разработке было решено использовать нисходящий подход, реализуя сначала модули верхнего уровня (интерфейс пользователя), а после переходя к модулям нижнего уровня (логика работы программы).

2.2 Разработка диаграммы вариантов использования

При проектировании приложения необходимо знать основные варианты взаимодействия пользователя с системой, для визуального представления была разработана диаграмма вариантов использования.

В результате анализа ТЗ были выявлены следующие варианты взаимодействия пользователя и приложения:

- 1) просмотр предлагаемой одежды,
- 2) обновление предлагаемой одежды,
- 3) изменение гардероба,
 - а) просмотр уже имеющейся одежды,
 - выбор имеющейся одежды,
 - удаление имеющейся одежды,
 - редактирование параметров имеющейся одежды,
 - б) добавление новой одежды,
- 4) редактирования настроек.

Рассмотрим варианты использования, связанные с гардеробом пользователя, показанные в таблицах 3 – 5. Так как этот вариант использования имеет наибольшее количество возможных взаимодействий, то он должен быть наиболее подробно проработан.

Таблица 3 – Вариант использования ***“Изменить гардероб”***

| Действия пользователя | Отклик системы |
|--|--|
| 1. Пользователь в панели навигации нажимает кнопку “Гардероб”. 3. Пользователь выбирает: а) нажимает на одну из категорий одежды. б) нажимает на кнопку “Расширить гардероб”. | 2. Приложение открывает окно “Гардероб”. 4. Приложение открывает окно: а) с одеждой выбранной категории, см. раздел Просмотр уже имеющейся одежды (таблица 2). б) с формой создания новой одежды, см. раздел Добавить новую одежду (таблица 5). |

Таблица 4 – Раздел ***“Просмотр уже имеющейся одежды”***

| Действия пользователя | Отклик системы |
|--|--|
| 1. Пользователь может: а) нажать на шаблонную одежду, отмеченную, как которой нет; б) нажать на шаблонную одежду, отмеченную, как которая есть; в) удалить одежду, ранее добавленную пользователем; г) редактировать одежду, ранее добавленную пользователем; д) нажать кнопку “Добавить” | 2. Приложение: а) на картинке шаблонной одежды появляется галочка и одежда отмечается в БД как имеющаяся и учитывается при подборе; б) на картинке шаблонной одежды снимается галочка и одежда отмечается в БД как отсутствующая и не учитывается при подборе; в) картинка пропадает с экрана и одежда удаляется из БД; г) открывается окно редактирования параметров (температуры, категории, пола и картинки); д) открывается окно добавление новой одежды, см. раздел Добавить новую одежду (таблица 3). |

Таблица 5 – Раздел ***“Добавить новую одежду”***

| Действия пользователя | Отклик системы |
|---|---|
| 1. Пользователь вводит название одежды, максимальную и минимальную рекомендованную температуру. | 3. Выпадает список выбора типов одежды. |

Продолжение таблицы 5

| Действия пользователя | Отклик системы |
|--|--|
| 2. Пользователь нажимает на кнопку “тип одежды”. | 6. Выпадает список выбора для какого пола предназначена одежда. |
| 4. Пользователь выбирает нужный тип одежды. | 9. Открывается окно выбора файлового приложения, откуда будет взята картинка. |
| 5. Пользователь нажимает на кнопку “для кого”. | 11. Открывается выбранное приложение. |
| 7. Пользователь выбирает для какого пола предназначена одежда. | 13. Файловое приложение закрывается. |
| 8. Пользователь нажимает на изображение картинки. | 15. Открывается окно: а) выбора категории одежды, если ранее была нажата кнопку “расширить гардероб”; б) той категории одежды, в которой была нажата кнопка “Добавить” |
| 10. Пользователь нажимает на одно из файловых приложений. | |
| 12. Пользователь выбирает одну из картинок и нажимает на неё. | |
| 14. Пользователь нажимает кнопку “сохранить”. | |

Разработанная диаграмма вариантов использования представлена на рисунке 14 [27].

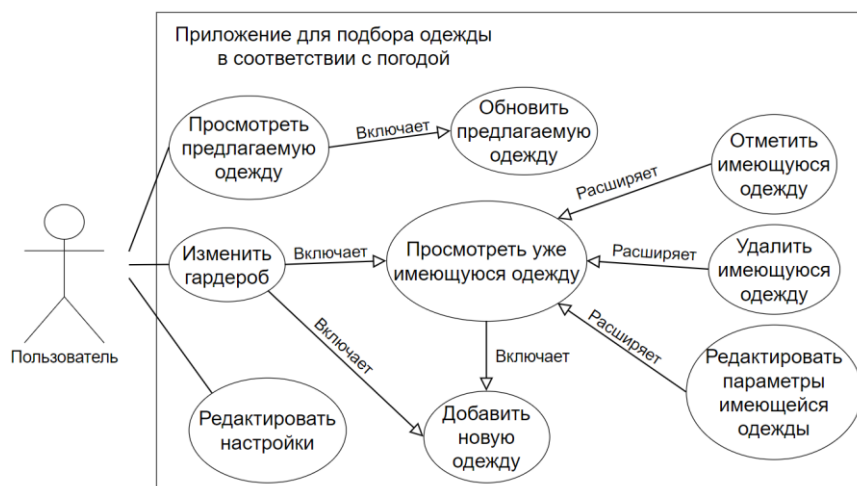


Рисунок 14 – Диаграмма вариантов использования

2.3 Построение концептуальной модели предметной области

При анализе предметной области, технического задания и структуры разрабатываемого ПО были выявлены следующие основные объекты:

- 1) главный экран с подобранной одеждой,
- 2) подобранная одежда,
- 3) данные,
- 4) текущая температура,
- 5) текущее местоположение;
- 6) настройки пользователя:
 - а) температурная шкала,
 - б) пол пользователя,
 - в) теплоощущение пользователя,
- 7) гардероб,
- 8) категории одежды:
 - а) шаблонная одежда,
 - б) пользовательская одежда,
 - в) данные одежды:
 - название,
 - минимальная и максимальная температура,
 - тип (категория),
 - для кого (пол),
 - картинка.

Меню содержит несколько кнопок – главный экран, окно настроек и гардероба. Главный экран выводит подобранную одежду, которая подбирается, основываясь на данных – текущей температуре, которая берется в зависимости от текущего местоположения; пользовательских настроек – какое у пользователя теплоощущение и пол; а также от самих температурных данных, которые хранятся в параметрах как шаблонной, так и пользовательской одежды. Сама одежда хранится в гардеробе, который в свою очередь для удобства разделяет одежду на несколько категорий. На рисунке 15 показано визуальное представление концептуальной модели предметной области.

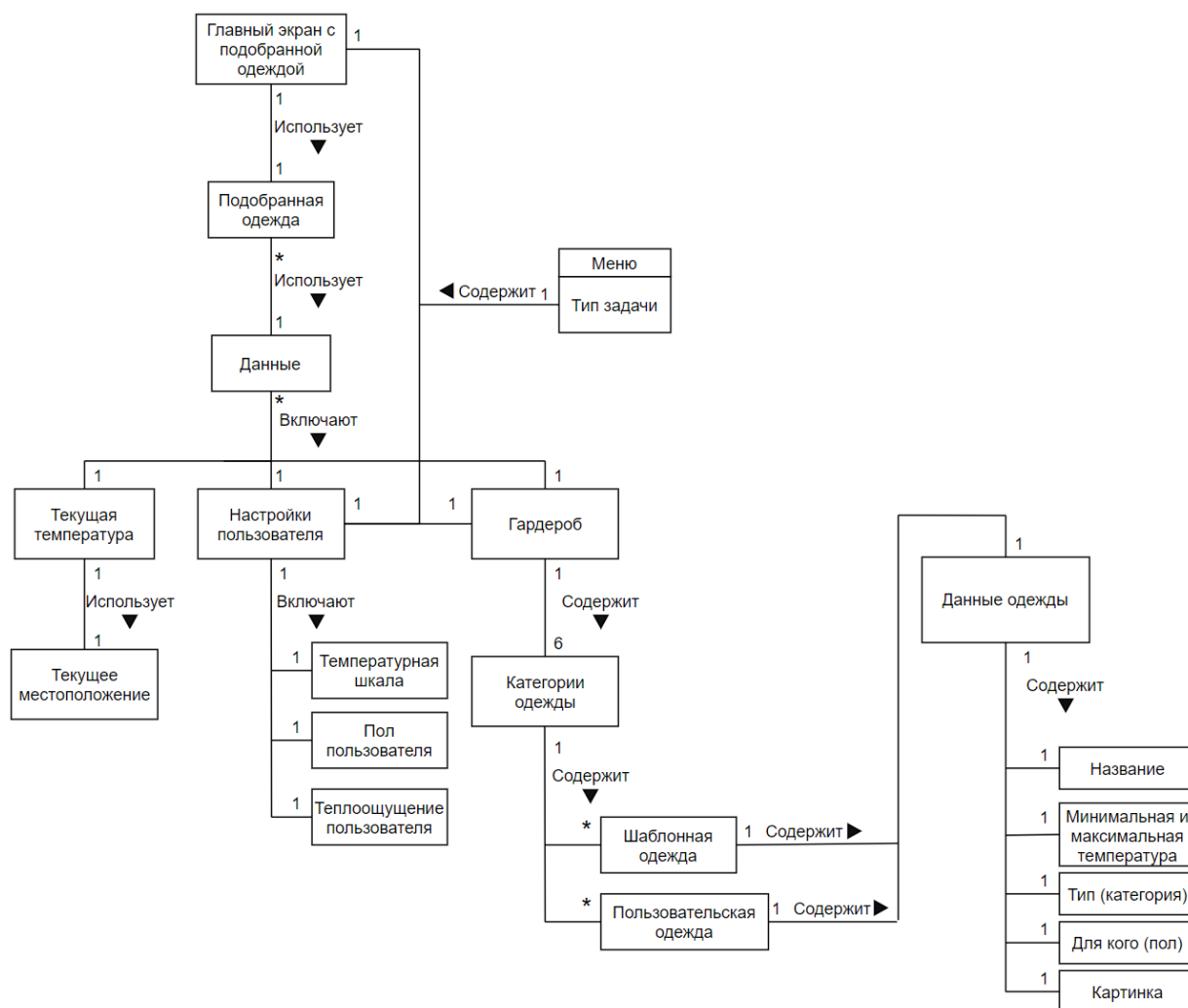


Рисунок 15 – Концептуальная модель предметной области

Основными объектами можно выделить:

- подобранная одежда, это картинки и названия, которые приложение выводит пользователю как вариант что можно сегодня одеть;
- настройки пользователя, в них пользователь настраивает приложение под себя, выбирая такие параметры как, теплоощущения и пол, которые влияют на подбор одежды;
- гардероб, в нем хранится вся одежда, то есть как шаблонная, так и добавленная пользователем;
- данные одежды, этот объект включает в себя параметры одежды, с помощью которых одежда сортируется по категориям и полу, а также основываясь на них, выбирается та одежда, что будет учитываться в подборе.

2.4 Построение схемы структурной информационной системы

Структурная схема программного продукта показывает разделение программы на её главные составляющие. На основе анализа технического задания, в разрабатываемом приложении, которое подбирает одежду по погоде, выявлено 3 подсистемы:

- 1) подсистема подбора одежды: в подсистеме алгоритма подбора одежды происходит подбор одежды для пользователя в соответствии с выбранными параметрами (настроек, текущей температуры и имеющейся одежды);
- 2) подсистема гардероба включает в себя две подсистемы:
 - а) в подсистеме подкатегорий одежды происходит выбор имеющейся одежды;
 - б) в подсистеме добавления новой одежды происходит добавление новой одежды;
- 3) подсистема пользовательских настроек, в которой выбираются теплоощущение, температурная шкала и пол пользователя.

На основе выявленных подсистем была составлена структурная схема ПП, показанная на рисунке 16.

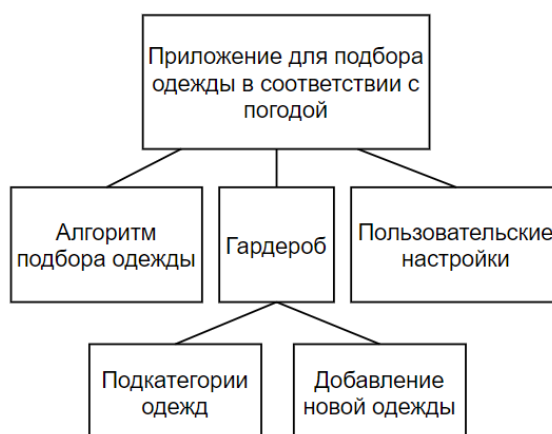


Рисунок 16 – Структурная схема информационной системы

2.5 Разработка интерфейса приложения

2.5.1 Построение диаграммы состояний интерфейса

В следствие анализа предметной области был выбран вид интерфейса со свободной навигацией, то есть пользователь может свободно перемещаться по

окнам приложения и смотреть, что программный продукт может предложить. Так устроены большинство современных приложений, и пользователи привыкли, что могут перемещаться по разным окнам интерфейса.

Чтобы не загромождать экран кнопками перехода между окнами, переключение между основными окнами было сделано с помощью выезжающей боковой панели. Для удобства, свободы и быстроты перемещения между окнами, боковую панель можно открыть в любом окне, сделав свайп с левой границы экрана в право. Разработанный граф состояний интерфейса, для наглядного отображения переходов, представлен на рисунке 17 [28].

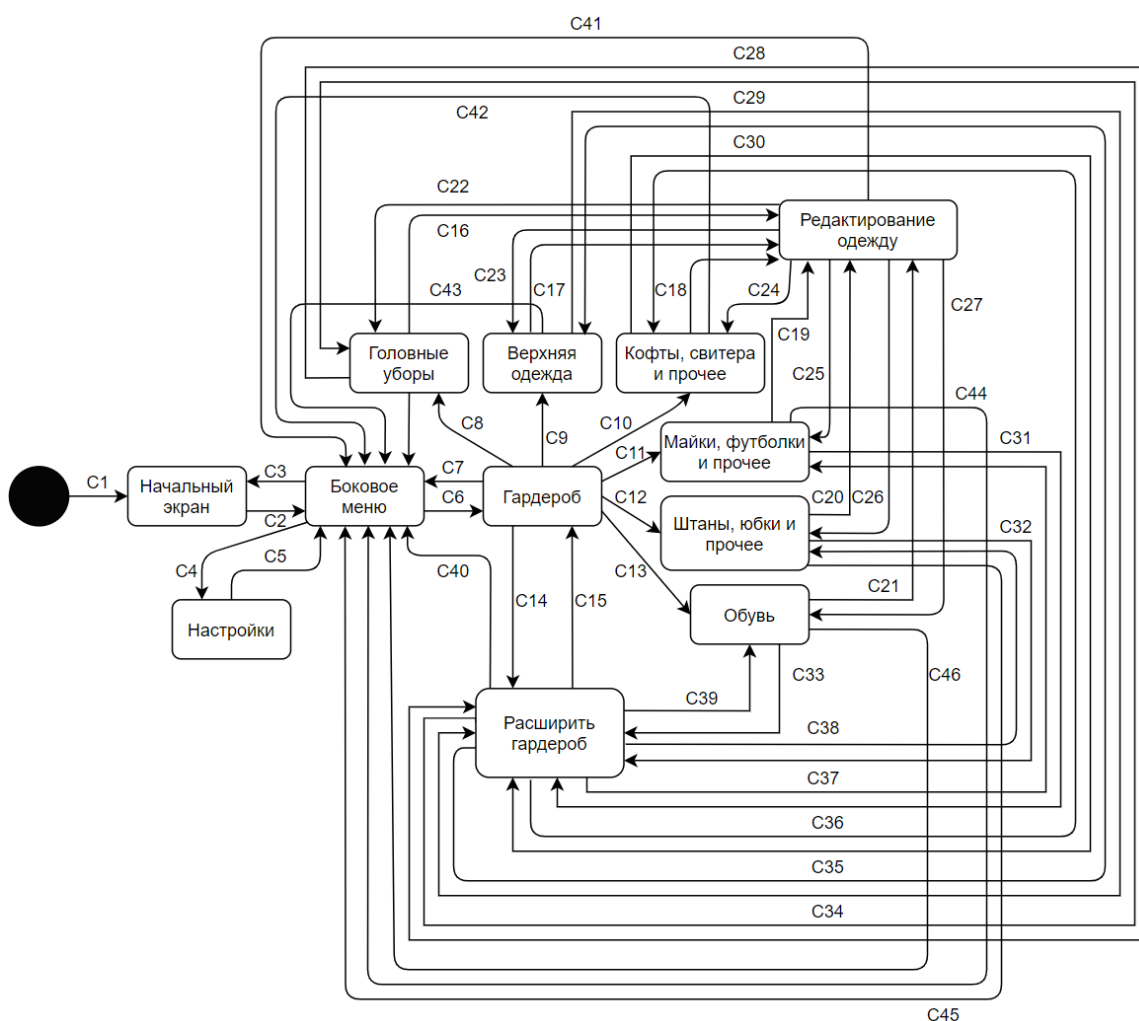


Рисунок 17 – Граф состояний интерфейса

C1 – активация формы при запуске приложения;

C2 – открытие навигационной панели при нажатии на кнопку “Добавить” или при свайпе с левой границы экрана в право;

C3 – по нажатию кнопки “Что надеть?” возвращение на главный экран;

С4 – по нажатию кнопки “Настройки” переход в окно пользовательских настроек;

С5 – открытие навигационной панели при нажатии на “Добавить”;

С6 – по нажатию кнопки “Гардероб” переход в окно категорий гардероба;

С7 – открытие навигационной панели при нажатии на “Добавить”;

С8 – по нажатию кнопки “ГОЛОВНЫЕ УБОРЫ” переход в соответствующую категорию одежды;

С9 – по нажатию кнопки “ВЕРХНЯЯ ОДЕЖДА” переход в соответствующую категорию одежды;

С10 – по нажатию кнопки “КОФТЫ, СВИТЕРА И ПРОЧЕЕ” переход в соответствующую категорию одежды;

С11 – по нажатию кнопки “МАЙКИ, ФУТБОЛКИ И ПРОЧЕЕ” переход в соответствующую категорию одежды;

С12 – по нажатию кнопки “ШТАНЫ, ЮБКИ И ПРОЧЕЕ” переход в соответствующую категорию одежды;

С13 – по нажатию кнопки “ОБУВЬ” переход в соответствующую категорию одежды;

С14 – по нажатию кнопки “РАСШИРИТЬ ГАРДЕРОБ” переход в окно добавления новой одежды;

С15 – по нажатию кнопки “Сохранить” возвращение в окно категорий гардероба;

С16 – С21 – по нажатию кнопки “Редактировать” на добавленной одежде, переход в окно редактирования параметров выбранной одежды;

С22 и С34 – по нажатию кнопки “Сохранить” возвращение в окно категории одежды “Головные уборы”;

С23 и С35 – по нажатию кнопки “Сохранить” возвращение в окно категории одежды “Верхняя одежда”;

С24 и С36 – по нажатию кнопки “Сохранить” возвращение в окно категории одежды “Кофты, свитера и прочее”;

C25 и C37 – по нажатию кнопки “Сохранить” возвращение в окно категории одежды “Майки, футболки и прочее”;

C26 и C38 – по нажатию кнопки “Сохранить” возвращение в окно категории одежды “Штаны, юбки и прочее”;

C27 и C39– по нажатию кнопки “Сохранить” возвращение в окно категории одежды “Обувь”;

C28 – C33 – по нажатию кнопки “Добавить” переход в окно добавления новой одежды;

C40 – C46 – открытие боковой панели при свайпе с левой границы экрана в право.

2.5.2 Разработка форм интерфейса

Приложение имеет прямое взаимодействие с пользователем, поэтому необходимо разработать удобный и интуитивно понятный интерфейс. Разработанные формы интерфейса проекта показаны на рисунках 18 – 23.

Главной задачей приложения является предоставления подобранной одежды в зависимости от погоды. Исходя из этого главным экраном приложения было решено разработать окно, которое включает в себя следующую информацию: местоположение, температура, список предлагаемой одежды. Окно главного экрана показана на рисунке 18.

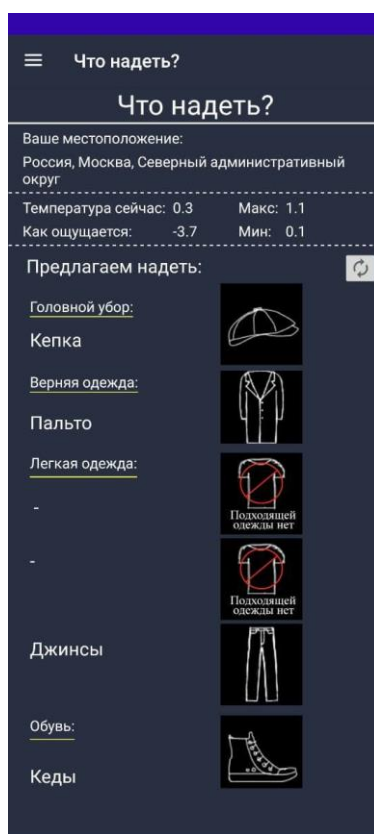


Рисунок 18 – Главного экран

Помимо предлагаемой одежды, рассортированной по категориям, текущего местоположения и температуры, присутствует кнопка обновления (стрелочки), которая предназначена для переподбора одежды. Если подходящей одежды нет, то будет выведено соответствующее изображение, как показано для “Легкой одежды”.

Для навигации по приложению было использовано боковое меню, показанное на рисунке 19. Оно удобно тем, что не загромождает экран кнопками навигации на каждой вкладке, а выносит их в отдельную вкладку, которую можно вызвать по необходимости.

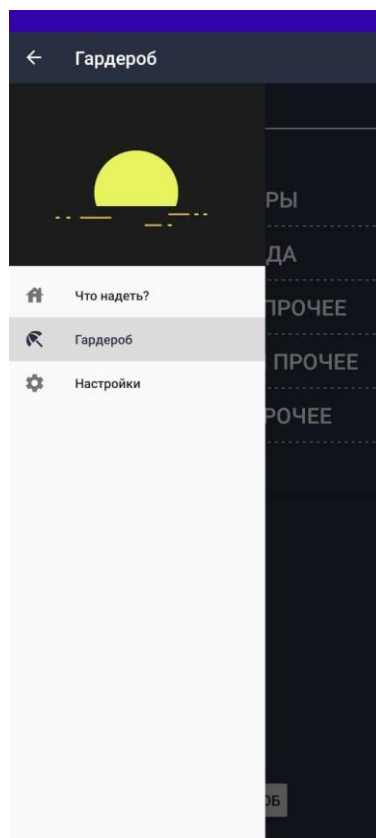


Рисунок 19 – Боковое меню

Через боковое меню происходит переключение между окнами “Что надеть?”, “Гардероб” и “Настройки”. В меню вынесены именно эти кнопки, так как эти окна являются основными экранами приложения. Открывается боковое меню с помощью кнопки “Меню” или при свайпе с левой границы экрана вправо.

Пользователь может настроить подбор одежды под себя в окне “Настройки”, которое показано на рисунке 20.

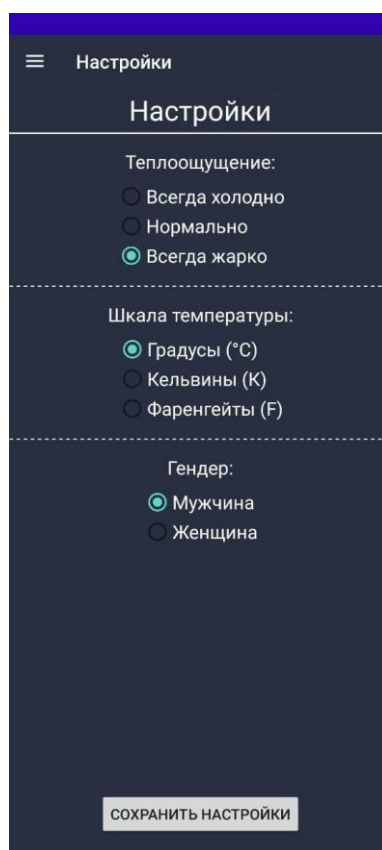


Рисунок 20 – Настройки

В настройках пользователь может выбрать пол и температурную шкалу, которая будет отображаться на главном экране, и теплоощущение:

- всегда холодно, то есть пользователь ощущает холод чаще остальных людей;
- нормально, то есть терморегуляция тела происходит нормально;
- всегда жарко, то есть пользователь ощущает тепло больше остальных, например, пользователю не холодно зимой без шапки.

Одежда рассортирована на категории, и для перехода к каждой из них, приложение имеет общее окно “Гардероб”, показанное на рисунке 21.

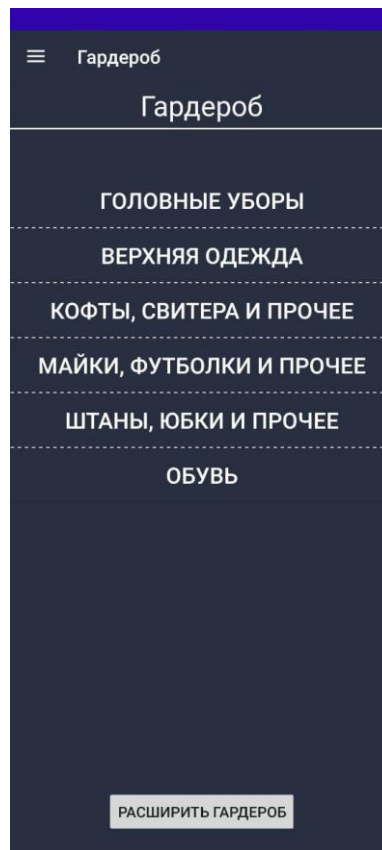


Рисунок 21 – Гардероб

В окне “Гардероб” отображены категории одежд, по которым рассортирована одежда, такими категориями являются: “Головные уборы”, “Верхняя одежда”, “Кофты, свитера и прочее”, “Майки, футболки и прочее”, “Штаны, юбки и прочее” и “Обувь”. Помимо этого, предусмотрена кнопка “Расширить гардероб” для возможности добавления новой одежды.

Для удобства просмотра имеющейся одежды для каждой категории разработано окно, в котором представлены все виды одежды из выбранной категории. Одно из таких окон показано на рисунке 22.

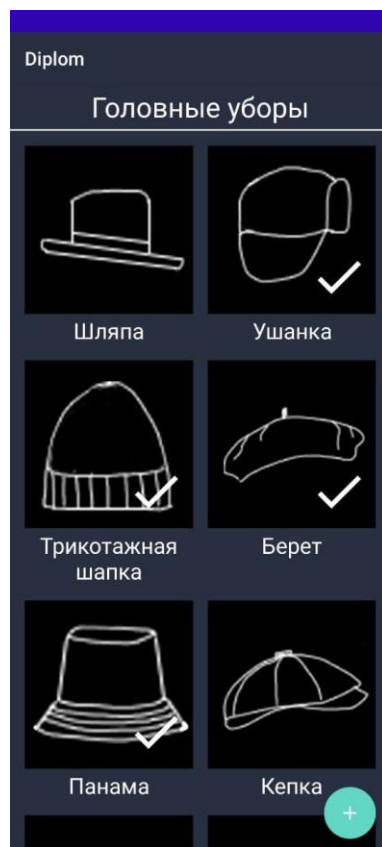


Рисунок 22 – Категория одежды

На рисунке 22 представлена шаблонная одежда категории головные уборы. Для учета той одежды, которая присутствует у пользователя предусмотрен знак в виде галочки. При подборе одежды учитываются лишь те виды одежды, которые имеют данный знак. На картинках одежд, у которых нет галочки, учитываться при подборе не будут. Также для добавления новой одежды была разработана кнопка “Добавить”, находящаяся в правом нижнем углу. Так как основная функция этого окна – это показать всю одежду конкретной категории, поэтому кнопка “Добавить” небольшого размера и размещена в углу, чтобы не оттягивать на себя много внимания.

Помимо добавления шаблонной одежды, для пользователя предусмотрена возможность добавлять новый вид одежды, для этого было разработано окно добавления, показанное на рисунке 23.

Diplom

Добавление одежды

Название:

Рекомендованная температура:

Максимальная:

Минимальная:

Тип одежды:

Для кого:

Рисунок 23 – Добавление одежды

Окно добавления одежды было решено реализовать формой заполнения данных, так как при добавлении одежды пользователю необходимо вводить данные. Этими данными являются: название, картинка, рекомендованная минимальная и максимальная температура и тип одежды и пола (то есть для кого она предназначена). Для сохранения введенных данных предусмотрена кнопка “сохранить”.

2.5.3 Разработка графов диалогов

Разрабатываемое приложение имеет несколько окон, следовательно диалог приложения и пользователя не всегда будет одинаковый. Для удобного взаимодействия пользователя и приложения необходимо продумать как приложение будет реагировать на различные действия пользователя, чтобы это более подробно показать были разработаны несколько графов диалогов:

- общий граф диалога;
- граф диалога добавления новой одежды.

Общий граф диалога показан на рисунке 24, который отображает работу приложения в целом. Сначала пользователь выбирает вкладку гардероб, в которой отмечает шаблонную одежду, и/или добавляет собственную. Далее настраивает параметры настроек под себя. И на основе этих действий пользователю подбирается одежда.

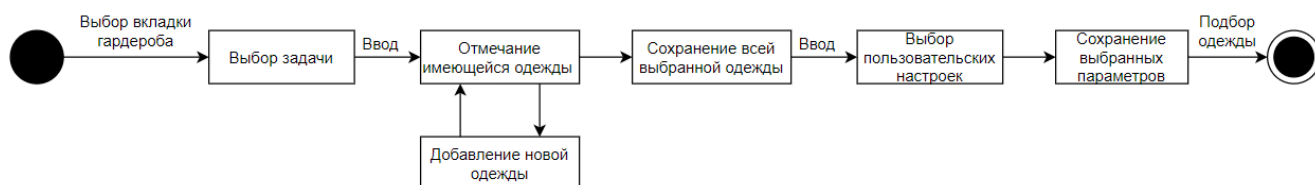


Рисунок 24 – Общий граф диалога

Граф диалога добавления новой одежды показан на рисунке 25, который показывает работу взаимодействия между приложением и пользователем при добавлении новой одежды. Сначала пользователь должен оказаться на одном из окон – гардероб или одна из категорий одежды. После нажать на кнопку добавления новой одежды. Далее ввести все параметры новой одежды и выбрать её картинку, в ином случае высветятся уведомления о невыполнении того или иного действия. В конце при нажатии кнопку “Сохранить” новая одежда добавляется в БД.

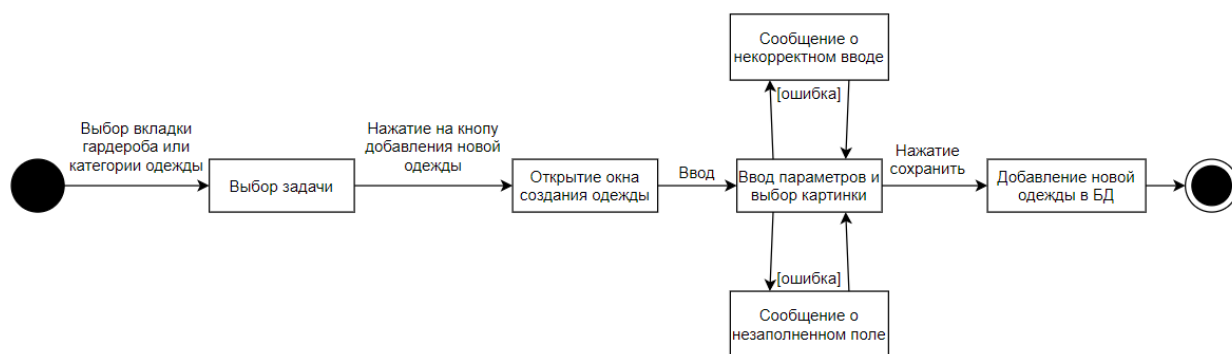


Рисунок 25 – Граф добавления новой одежды

2.6 Выбор методов хранения данных приложения

Приложение содержит в себе данные, которые при выходе из приложения не должны потеряться, поэтому необходимо предусмотреть метод хранения этих данных.

В приложении хранятся следующие данные:

- шаблонная одежда, она должна быть в приложении сразу после установки у каждого пользователя, то есть на всех возможных устройствах, поэтому хранится она должна в памяти приложения;
- пользовательская одежда, она у каждого пользователя своя, поэтому каждая новая добавленная одежда хранится в БД, а при удалении, удаляется из БД;
- пользовательские настройки, они у всех общие, но выбраны могут быть по-разному, поэтому настройки каждого отдельного пользователя сохраняются на каждом отдельном устройстве.

В разработанном ПО используется три метода хранения информации:

- БД Room, которая является более удобной надстройкой SQL [29];
- `sharedpreferences` для небольшого количества информации, которая сохраняется в отдельный файл [30];
- просто массивы, которые хранят параметры шаблонной, то есть заранее заданной, одежды.

Шаблонная одежда, и информация о ней, хранится в массивах типа `List`.

Листинг хранения шаблонной одежды из категории “Верхняя одежда”:

```
private static List<ClothInitial> generateOuterwear() {
    List<ClothInitial> clothes = new ArrayList<>();
    clothes.add(new ClothInitial(51, "Бомбер",
R.drawable.outerwear_bomber, 5, 15, ClothesType.OUTERWEAR,
ClothGender.UNISEX));
    return clothes;
}
```

Это сделано для того, чтобы уменьшить количество обращений и не создавать лишних запросов к БД. Последовательность того, что хранит элемент массива: `id`, название одежды, ссылку на картинку, минимальная температура, максимальная температура, тип, пол. В БД шаблонной одежды хранится только информация о наличии её у пользователя. Если пользователь отмечает одежду, то она добавляется в таблицу и ставится “1” в поле наличия. Когда пользователь

снимает отметку, то одежда удаляется из БД, это сделано, чтобы не хранить лишнюю информацию, которая занимает память.

Данные одежды, добавленной пользователем, полностью хранятся в БД. Таблица пользовательской одежды содержит поля: id, название, название картинки, максимальная и минимальная температура, пол и тип.

Таблицы базы данных шаблонной и пользовательской одежды показаны на рисунке 26.

| ClothCustom | | ClothInitial | |
|-------------|---|--------------|---------------------|
| PK | id int | PK | id int |
| | name String imageFilePath String tempMin int tempMax int gender ClothGender type ClothesType | | isAvailable boolean |

Рисунок 26 – Таблицы базы данных

С помощью `sharedpreferences` хранится информация, заданная в настройках (теплоощущение, температурная шкала и пол). В окне “Настройки”, при нажатии кнопки “Сохранить” происходит сохранение выбранной информации. Подгрузка ранее выбранных параметров происходит в момент запуска приложения.

2.7 Разработка алгоритма подбора одежды

Одной из основных подсистем приложения является алгоритм подбора одежды. Алгоритм необходимо разработать так, чтобы точность его подбора удовлетворяла требованиям, введенным пользователем, следовательно он должен содержать в себе большое количество проверок:

- проверка, что категория одежды соответствует подбираемой категории;
- проверка наличие выбранной одежды у пользователя;
- проверка соответствия одежды полу пользователя;
- проверка, что температурные параметры одежды подходят под реальную температуру.

Обработка шаблонной одежды, и одежды, добавленной пользователем, изначально происходит раздельно. Это сделано для избежание лишних ветвлений при проверках, что может привести к незаметным ошибкам.

Первым этапом идет работа с шаблонной одеждой. Её обработка происходит в несколько шагов.

1. Создается массив шаблонной одежды нужной категории.
2. Поочередно берется каждая вещь из только что созданного массива.
3. Проверка наличия одежды у пользователя в БД.
 - а) Если вещи нет, то возвращение к пункту 2.
 - б) Если вещь есть, то переход к пункту 4
4. Подходит ли одежда по текущей температуре?
 - а) Если нет, то возвращение к пункту 2.
 - б) Если вещь есть, то переход к пункту 5
5. Соответствует ли одежда полу пользователя?
 - а) Если нет, то возвращение к пункту 2.
 - б) Если вещь есть, то переход к пункту 6
6. Добавление в массив подходящей шаблонной одежды.

Следующим этапом идет работа с добавленной пользователем одеждой. Её обработка происходит в несколько шагов.

1. Поочередно берется каждая вещь, добавленная пользователем.
2. Проверка одежды на тип.
 - а) Если тип другой, то возвращение к пункту 1.
 - б) Если тип подошел, то переход к пункту 3.
3. Подходит ли одежда по текущей температуре?
 - а) Если нет, то возвращение к пункту 1.
 - б) Если да, то переход к пункту 4.
4. Соответствует ли одежда полу пользователя?
 - а) Если нет, то возвращение к пункту 1.
 - б) Если да, то переход к пункту 5.
5. Добавление в массив подходящей пользовательской одежды.

После происходит работа с подходящей одеждой. Подбор происходит в несколько шагов.

1. Среди шаблонной и пользовательской одежды есть подходящая?
 - а) Если нет, то возвращение к пункту 2.
 - б) Если есть, то возвращение к пункту 3.
2. Вывод картинки, что подходящей одежды нет.
3. Выбор случайной одежды среди подходящей.

Разработанная схема алгоритма представлена на рисунке 27.

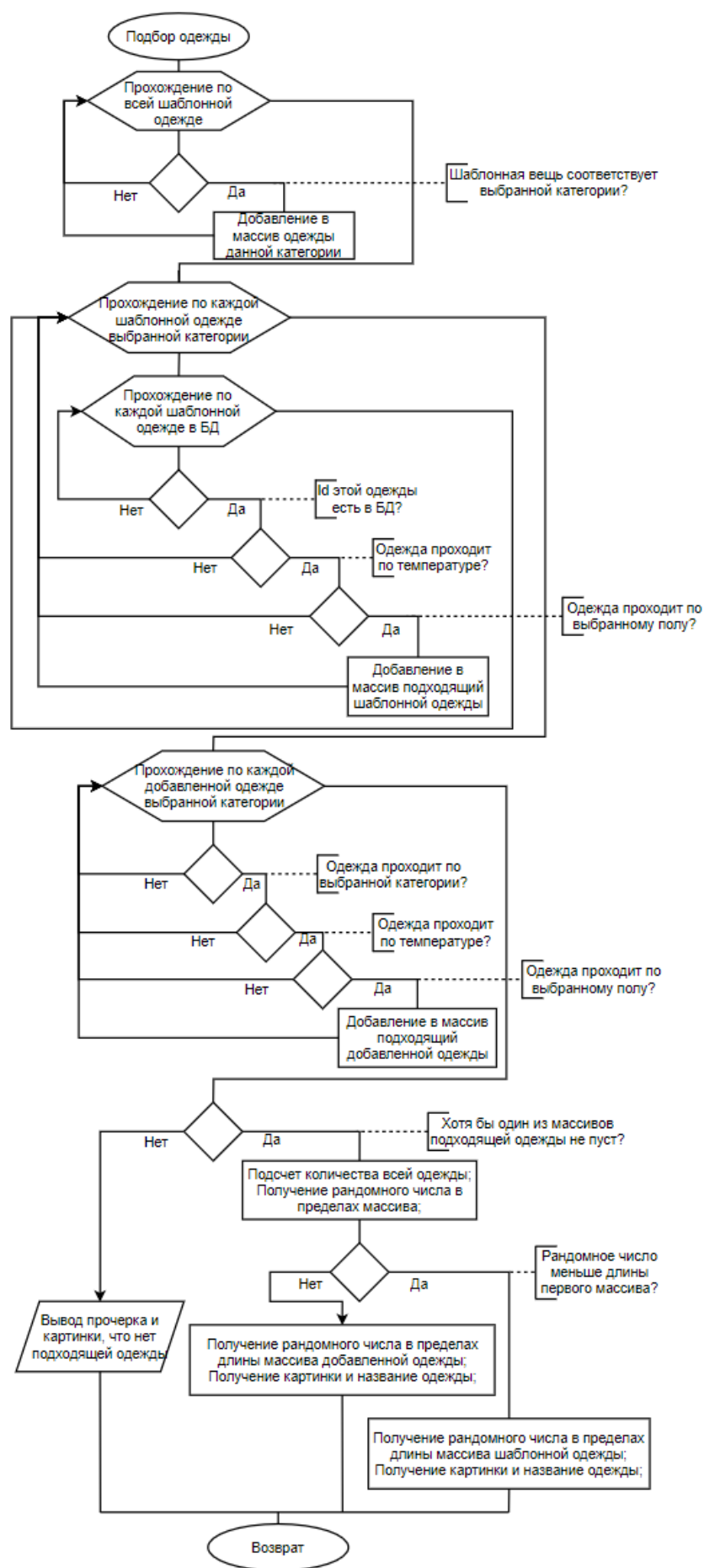


Рисунок 27 – Алгоритм подбора одежды

2.8 Разработка диаграммы уровня реализации

После окончания разработки ПО была разработана диаграмма классов уровня реализации, чтобы представить созданные классы с их полями и методами. Диаграмма классов уровня реализации показана в приложении В.

В процессе написания кода были созданы классы, которые можно разделить по категориям того, за что они отвечают.

За определение местоположение устройства пользователя отвечают классы:

- ApiClient – хранит API сайта openweather и создает запрос;
- WeatherData – преобразует полученные данные из float формата в String;
- Main – указаны параметры, которые необходимо будет сохранить при принятии JSON запроса;
- WeatherRequest – принимает запрос информации о погоде;
- ApiInterface – интерфейс, хранящий в себе ссылку на пользователя с названием местоположения.

За базу данных отвечают следующие классы:

- ClothCustomEntity – описана таблица пользовательской одежды;
- ClothInitialEntity – описана таблица шаблонной одежды;
- ClothCustomDao – прописаны SQL запросы на получение всей одежды, поиска по id, по типу, а также добавление, изменение и удаление (из БД). Всё для пользовательской одежды;
- ClothInitialDao – прописаны SQL запросы на получение всей одежды, поиска по id, по типу, а также добавление и удаление (стоит галочка или нет). Всё для шаблонной одежды;
- ClothesAppDatabase – создание сущности базы данных и привязка к каждой из таблиц конкретных методов;
- Converters – преобразует данные из enum в int, когда идет запрос к БД, и наоборот, когда информация поступает из БД;
- RoomWrapper – создание базы данных, если её нет, или её возвращение при сторонних запросах.

Класс `SharedPrefsWrapper` отвечает за сохранение и получение (при запуске приложения) параметров, выбранных в настройках.

Следующие классы отвечают за принятие данных из БД для дальнейшей работы с ней:

- `ClothesInitialFactory` – хранит в себе массивы шаблонной одежды каждого типа;
- `ClothCustom` и `ClothInitial` – файлы принимают всю возможную информацию для дальнейшей обработки. Первый для пользовательской одежды, второй для шаблонной;
- `CustomClothModel` и `InitialClothModel` – файлы принимают информацию, необходимую для отображения изображений. Первый для пользовательской одежды, второй для шаблонной;
- `IClothModel` – позволяет в одном списке объединить объекты из файлов `CustomClothModel` и `InitialClothModel`.

Классы, отвечающие за работу с одеждой в гардеробе:

- `ClothesListActivity` – открытие подкатегории, редактирования, создания новой одежды и вызов функции вывода. Принимает информацию, что кнопки были нажаты и посылает эту инфу в `ClothesListAdapter`;
- `ClothesListAdapter` – отрисовывает одежду и названия, заполняя элементы списка вых данными. А также, обрабатывает информацию, какая из кнопок были нажаты и вызывает соответствующую функцию в `ClothesListFragment`;
- `ClothesListFragment` – обрабатывает добавление/удаление шаблонной одежды из БД. И отбирает ту одежду, которая будет отрисовываться в соответствии с полом и наличием у пользователя.

Классы, отвечающие за создание новой одежды или редактирование уже существующей пользовательской:

- `NewClothActivity` – принимаем операцию из `Activity`, проверяет что за действие выбрано, выбирает `id` или тип одежды и вызывает функцию обработчик из файла `NewClothFragment`;

— NewClothFragment – открывает соответствующее окно (создания или редактирования) и отвечает за все действия, что происходят при создании и редактировании.

А также есть классы:

— ClothesMainActivity – прописана работа навигационной (боковой) панели;

— RecommendationFragment – получает геолокацию, выводит температуру, корректирует температуру в зависимости от теплоощущения, подбирает рандомную одежду и выводит её на главный экран;

— WardrobeFragment – общая вкладка гардероба. Создает и обрабатывает кнопки открытия подкатегорий и кнопку “расширить гардероб”;

— SettingsFragment – работа с вкладкой “Настройки”, обработка выбора каждого из параметров (теплоощущения, гендера и температурной шкалы).

2.9 Разработка диаграммы компоновки

Для более детального понимания связи компонентов и работы приложения в целом, была разработана диаграмма компоновки, показанная на рисунке 28 [31].

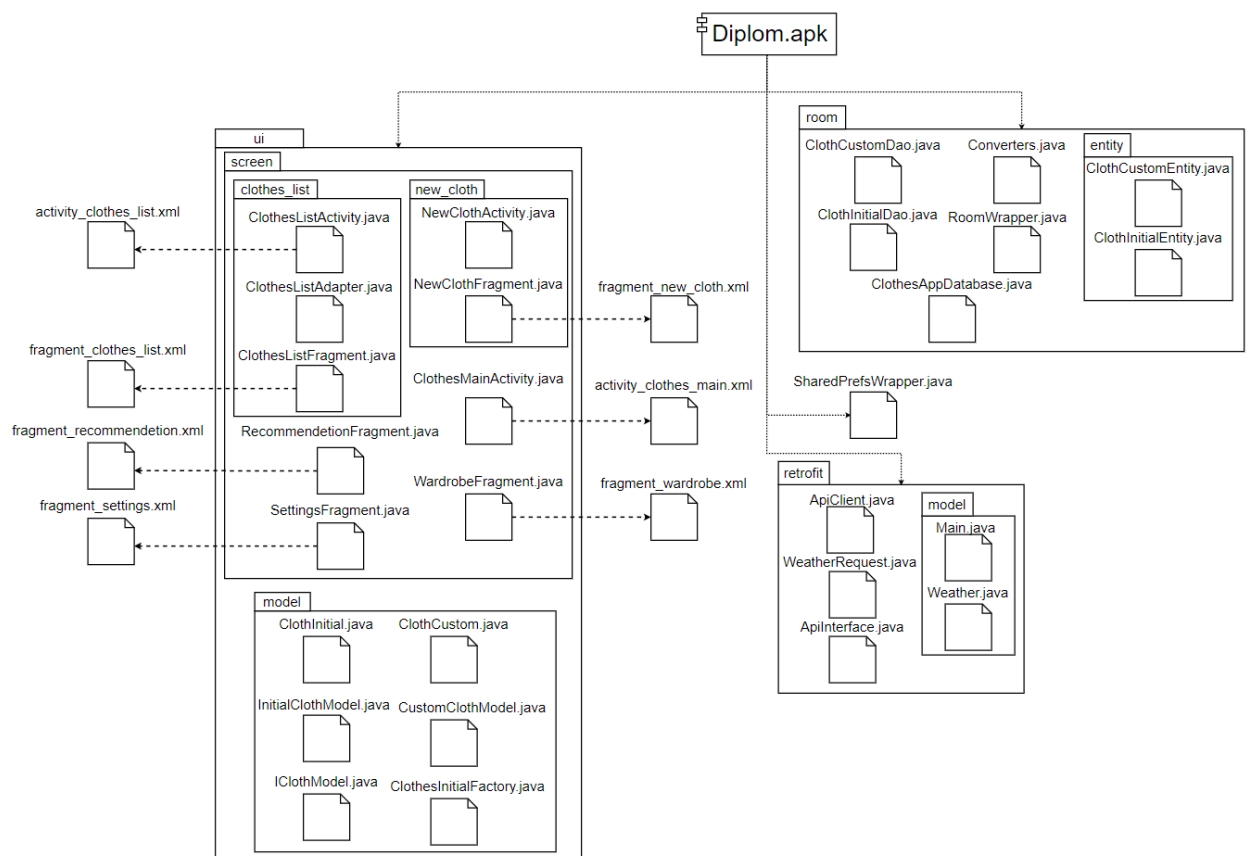


Рисунок 28 – Диаграмма компоновки

Исполняемый арк файл собирается из исполняющих файлов с расширением .java и вспомогательными файлами с расширением .xml, которые содержат обертку файлов .java. Файлы разделены по трем папкам, с собственными подпапками, в соответствии с выполняемыми функциями.

Файлы, входящие в папку ui, отвечают за взаимодействие пользователя с интерфейсом приложения, а также за логику работы программы (ClothesListActivity, ClothesListAdapter, ClothesListFragment, NewClothActivity, NewClothFragment, WardrobeFragment, ClothesMainActivity, RecommendationFragment, SettingsFragment, ClothInitial, ClothCustom, InitialClothModel, CustomClothModel, IclothModel, ClothesInitialFactory).

Файлы, входящие в папку room, отвечают за базу данных приложения (ClothCustomDao, ClothInitialDao, Converters, RoomWrapper, ClothesAppDatabase, ClothCustomEntity, ClothInitialEntity).

Файлы, входящие в папку retrofit, отвечают за взаимодействие приложения с интернет-ресурсом openweather (ApiClient, WeatherRequest, Main, Weather и ApiInterface).

2.10 Комплексное тестирование программного продукта

2.10.1 Функциональное тестирование

Функциональное тестирование подразумевает под собой тестирование по принципу “черного ящика”. Вводя данные, которые может ввести пользователь во время использования программы. В тестировании используются три вида тестовых наборов:

- эквивалентное разбиение,
- анализ граничных значений,
- анализ причинно-следственных связей.

Эквивалентное разбиение работает по принципу деления вводимых данных на эквивалентные группы, проверяя тем самым одним тестом множество частных тестов. Сначала создаются классы эквивалентности, и после на их основе проводят тесты. Несколько примеров тестирования эквивалентным разбиением приведены в таблице 6 [32].

Классы эквивалентности можно разбить на числа и не числа. Числа можно разбить на рациональные и целые. Целые числа можно разбить на отрицательные и положительные.

Таблица 6 – Тестирование эквивалентным разбиением

| Номер теста | Назначение теста | Значение исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|-------------|--|---|------------------------|------------------------|---------------------------|
| 1 | Проверка на ввод целых положительных чисел при добавлении одежды | Название: 10 Максимальная температура: 9 Минимальная температура: 1 | Добавление одежды в БД | Добавление одежды в БД | Программа работает верно. |

Продолжение таблицы 6

| Номер теста | Назначение теста | Значение исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|-------------|--|--|--------------------------------------|--------------------------------------|---------------------------|
| 2 | Проверка на ввод целых положительных чисел при редактировании одежды | Название: 11 Максимальная температура: 9 Минимальная температура: 1 | Информация об одежде обновилась в БД | Информация об одежде обновилась в БД | Программа работает верно. |
| 3 | Проверка на ввод целых отрицательных чисел при добавлении одежды | Название: -10 Максимальная температура: -1 Минимальная температура: -9 | Добавление одежды в БД | Добавление одежды в БД | Программа работает верно. |

Эквивалентное тестирование показало, что приложение учитывает, что вводиться могут самые различные параметры.

Далее идет проверка условий, которые могут быть введены по ошибке. Несколько примеров тестирования анализа граничных значений приведены в таблице 7.

Таблица 7 – Анализ граничных значений

| Номер теста | Назначение теста | Значение исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|-------------|--|---|--------------------------------------|--------------------------------------|---------------------------|
| 1 | Проверка на ввод 0 при добавлении одежды | Максимальная температура: 0 Минимальная температура: 0 | Добавление одежды в БД | Добавление одежды в БД | Программа работает верно. |
| 2 | Проверка на ввод 0 при изменении одежды | Максимальная температура: 0 Минимальная температура: 0 | Информация об одежде обновилась в БД | Информация об одежде обновилась в БД | Программа работает верно. |

Продолжение таблицы 7

| Номер теста | Назначение теста | Значение исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|-------------|---|---|---|---|---------------------------|
| 3 | Проверка на ввод большого числа при добавлении одежды | Максимальная температура: 1000 Минимальная температура: 1000 | Уведомление об ошибке в полях температуры | Уведомление об ошибке в полях температуры | Программа работает верно. |

Тестирование на граничных условиях показало, что при вводе значений, не предусмотренных программой, программа выдает уведомления об ошибках и работает корректно.

Еще одна стратегия черного ящика – это анализ причинно-следственных связей, которые тоже могут проявить некоторые непредвиденные ошибки. Несколько примеров тестирования приведены в таблице 8.

Таблица 8 – Анализ причинно-следственных связей

| Номер теста | Назначение теста | Значение исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|-------------|--|--------------------------|--|--|---------------------------|
| 1 | Проверка на не ввод данных при добавлении | Все поля пустые | Уведомление о некорректном вводе | Уведомление о некорректном вводе | Программа работает верно. |
| 2 | Проверка на не ввод данных при их изменении | Все поля пустые | Уведомление о некорректном вводе | Уведомление о некорректном вводе | Программа работает верно. |
| 3 | Проверка на не выбранную картинку при добавлении | Картинка не выбрана | Уведомление о том, что не выбрана картинка | Уведомление о том, что не выбрана картинка | Программа работает верно. |

Тестирование причинно-следственной связью показало, что программа работает корректно.

Тестирования методом черного ящика, или же, функциональное тестирование, показало, что при различных возможных вводах и действиях, программа работает правильно. Полное тестирование программного продукта методом черного ящика приведено в приложение Д.

2.10.2 Автоматическое тестирование

Автоматические тесты будут проводиться по принципу белого ящика. То есть заранее будет известно, что и как тест будет проверять. Они будут проверять работоспособность приложения с точки зрения кода, обращаясь через него к элементам интерфейса.

Для таких тестов были выбраны Espresso тесты. То, как они реализованы подробнее рассказано в главе 3. В таблице 9 приведены результаты тестирования.

Таблица 9 – Тестирование белым ящиком

| Назначение теста | Результат теста |
|--|-------------------------------------|
| Работа кнопки “Меню” | Взаимодействия происходит корректно |
| Работа кнопки “Что надеть?” | Взаимодействия происходит корректно |
| Работа кнопки “Гардероб” | Взаимодействия происходит корректно |
| Работа кнопки “Отметить” | Взаимодействия происходит корректно |
| Работа кнопки “Снять отметку” | Взаимодействия происходит корректно |
| Работа кнопки “Выбор теплоощущения” | Взаимодействия происходит корректно |
| Работа кнопки “Выбор шкалы температуры” | Взаимодействия происходит корректно |
| Работа кнопки “Выбор пола” | Взаимодействия происходит корректно |
| Работа кнопки “Сохранить” | Взаимодействия происходит корректно |
| Работа кнопки “Настройки” | Взаимодействия происходит корректно |
| Работа кнопки “Обновить” | Взаимодействия происходит корректно |
| Работа кнопки “Головные уборы” | Взаимодействия происходит корректно |
| Работа кнопки “Верхняя одежда” | Взаимодействия происходит корректно |
| Работа кнопки “Кофты, свитера и прочее” | Взаимодействия происходит корректно |
| Работа кнопки “Майки, футболки и прочее” | Взаимодействия происходит корректно |

Продолжение таблицы 9

| | |
|--------------------------------------|-------------------------------------|
| Работа кнопки “Штаны, юбки и прочее” | Взаимодействия происходит корректно |
| Работа кнопки “обувь” | Взаимодействия происходит корректно |
| Работа кнопки “Расширить гардероб” | Взаимодействия происходит корректно |
| Работа кнопки “Добавить” | Взаимодействия происходит корректно |
| Работа кнопки “Редактировать” | Взаимодействия происходит корректно |
| Работа кнопки “Удалить” | Взаимодействия происходит корректно |

Espresso тестирование показало, что код корректно отображает элементы интерфейса при каких-либо взаимодействиях с ним.

2.10.3 Оценочное тестирование

После проведения тестов на работоспособность, приложение можно протестировать с помощью конечных пользователей, которые дадут свою оценку разработанному приложению по 10 бальной шкале. В процессе оценочного тестирования участвовали 10 студентов из разных университетов. Участники тестирования оценивали программу по четырем критериям:

- удобство интерфейса – насколько прост и интуитивно понятен интерфейс;
- работоспособность приложения – стабильность работы;
- правильность работы – правильно ли работают алгоритмы и интерфейс приложения;
- универсальность – под все ли возможные погодные, или какие-либо другие, условия подходит приложение.

Результаты опроса показаны в таблице 10.

Таблица 10 – Оценочное тестирование

| № Пользователя | Удобство интерфейса | Работоспособность приложения | Правильность работы | Универсальность |
|----------------|---------------------|------------------------------|---------------------|-----------------|
| 1 | 9 | 10 | 10 | 8 |
| 2 | 10 | 10 | 9 | 10 |
| 3 | 10 | 10 | 10 | 7 |
| 4 | 10 | 10 | 10 | 8 |

Продолжение таблицы 10

| | | | | |
|-----------------|-----|----|-----|-----|
| 5 | 9 | 10 | 10 | 9 |
| 6 | 10 | 10 | 9 | 7 |
| 7 | 8 | 10 | 10 | 9 |
| 8 | 10 | 10 | 10 | 8 |
| 9 | 9 | 10 | 10 | 10 |
| 10 | 10 | 10 | 9 | 6 |
| Средняя оценка: | 9,5 | 10 | 9,7 | 8,2 |

Итоги тестирования показали, что приложение работает стабильно, интерфейс понятен и удобен, алгоритмы и интерфейс ПО работают корректно, но оно не универсально: нет учета дождя и нет сочетания стилей одежды под планы пользователя на день.

3 Разработка технологии тестирования приложения

Для проверки корректной работоспособности программы необходимо её протестировать. Чтобы проверить все аспекты работы приложения, его необходимо протестировать в как можно больших ситуациях, когда могут встретиться ошибки.

Какие возможные ошибки можно выделить в разработанном ПО:

- ошибке в коде,
- некорректное отображение интерфейса,
- недопустимый ввод данных
- некорректная работа системы в целом.

Каждое из тестирований в какой-то мере затрагивает каждое из мест возможных ошибок.

На первом этапе необходимо проверить работу приложения изнутри, то есть с точки зрения кода. Помимо проверки написанного кода на корректность, это также затронет отображение элементов интерфейса. Тесты будут обращаться к объектам интерфейса и проверять их корректное отображение и взаимодействия. Инструментов для такой проверки может стать автоматическое тестирование посредством использования Espresso тестов [33].

Espresso – тестовый фреймворк от Google для проверки работы отдельных элементов интерфейса для систем под управлением ОС андроид [34][35].

Листинг тестов будет показан на примере тестирования работы бокового меню.

Листинг, объявляющий область, где будет проводиться тестирование:

```
@Rule
public ActivityTestRule<ClothesMainActivity>
activityActivityTestRule = new
ActivityTestRule<>(ClothesMainActivity.class);
```

1) Листинг проверки вызова бокового меню:

```
@Test
public void test1() {
    onView(withId(R.id.drawer_layout)).perform(click()).check(
```

```
matches(isDisplayed())));
}
```

2) Листинг нажатия кнопки “Что надеть?” и переход на главный экран:

```
@Test
public void test2(){
    onView(withId(R.id.
nav_recommendation_layout)).perform(click()).check(matches (
isDisplayed())));
}
```

3) Листинг нажатия кнопки “Гардероб” и переход на экран гардероба:

```
@Test
public void test3(){
    onView(withId(R.id.nav_wardrobe_layout)).perform(click())
.check(matches(isDisplayed())));
}
```

4) Листинг нажатия кнопки “Настройки” и переход на экран настроек:

```
@Test
public void test4(){
    onView(withId(R.id.nav_settings_layout)).perform(click())
.check(matches(isDisplayed())));
}
```

Такими тестами будет проверена работа приложения изнутри, с точки зрения работоспособности кода, а также они проверят корректность отображения элементов интерфейса.

Далее ПО необходимо протестировать на возможные взаимодействия пользователя с ним – то есть черным ящиком. Это необходимо, чтобы проверить приложение на нюансы, которые могли быть не учтены при разработке. А также этот вид тестирования задействует проверку работоспособности интерфейса.

Черный ящик – это тестирование приложения при самых различных ситуациях, которые пользователь может сделать над программой (например, ввести очень большое число). Оно включает три набора тестов:

— эквивалентное разбиение, чтобы не проверять все возможные ситуации, так как нельзя предугадать все возможные варианты ввода, а также это

заняло бы слишком много времени, для этого тесты разбивают по эквивалентным классам, то есть один такой тест заменяет множество аналогичных;

- анализ граничных значений, эти тесты проверяют приложения на ввод значений, которые не предполагаются программой (например, температура 100000 градусов);
- анализ причинно-следственных связей, тесты проверяют правильно ли реагирует программа, когда происходит то, или иное действие.

Сначала будет задействовано эквивалентное разбиение, чтобы выявить ошибки при обычных возможных ситуациях (которые были продуманы программистом). Далее тесты граничных значений, чтобы проверить моменты, которые могли быть не учтены при разработке. И после анализ причинно-следственных связей чтобы проверить приложение на возможные действия пользователя.

В заключении разработанный продукт будет протестирован реальными пользователями. Оценочное тестирование позволит выявить ошибки, которые могли быть не найдены при проведении предыдущих тестов. А также пользователи могут дать отзывы по самой программе в целом и поводу того, что можно улучшить и\или доработать (как минимум с визуальной стороны), в частности.

ПО будет протестировано тремя способами для выявления наибольшего количества ошибок и недоработок.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы бакалавра, в соответствии с техническим заданием, было произведено проектирование и реализация приложения для подбора одежды в соответствии с погодой, а также написана сопутствующая техническая документация [36].

Было проведено исследование предметной области и проанализированы существующие аналоги, выявлены их сильные и слабые стороны, на основе чего были определены основные функции разработанного приложения. Создана концепция приложения и поэтапная его реализация. В процессе разработки была создана диаграмма вариантов использования, концептуальная модель предметной области, структурная схема, разработаны формы интерфейса и граф состояния интерфейса, выбран метод хранения информации и спроектирован алгоритм подбора одежды в соответствии с погодой. В результате выполнения работы было разработано программное приложение, позволяющее подбирать одежду в зависимости от погодных условий.

Помимо этого, была разработана технология тестирования ПП и произведено комплексное тестирование, включающее в себя функциональное, автоматическое и оценочное, которое показывает корректную работу приложения и потенциал для его развития.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Яндекс изнутри: сколько людей пользуются его сервисами [Электронный ресурс]. – Режим доступа: <https://tokar.ua/read/9412> (дата обращения: 25.09.2021)
- 2 Google Play [Электронный ресурс] – режим доступа: <https://play.google.com/store?hl=ru&gl=US> (дата обращения: 21.09.21)
- 3 Getwardrobe - гардероб, стилист, одежда, мода [Электронный ресурс] – режим доступа: <https://play.google.com/store/apps/details?id=com.ThreeBoots> (дата обращения: 22.09.21)
- 4 Combyne - Outfit ideas & outfit creation [Электронный ресурс] – режим доступа: <https://play.google.com/store/apps/details?id=com.combyne.app> (дата обращения: 22.09.21)
- 5 Wear This - Outfits by Weather [Электронный ресурс] – режим доступа: <https://play.google.com/store/apps/details?id=com.thistechnologies.wearthis> (дата обращения: 22.09.21)
- 6 Weather Awear - Dress for the weather [Электронный ресурс] – режим доступа: <https://play.google.com/store/apps/details?id=com.weatherawear> (дата обращения: 23.09.21)
- 7 Weatherproof - Weather & Clothes [Электронный ресурс] – режим доступа: <https://play.google.com/store/apps/details?id=com.changemystyle.weatherproof> (дата обращения: 23.09.21)
- 8 What To Wear Today [Электронный ресурс] – режим доступа: <https://play.google.com/store/apps/details?id=nl.diecke.wtw> (дата обращения: 23.09.21)
- 9 Все виды одежды: список с названиями и фото [Электронный ресурс]. – Режим доступа: <https://obliqo.ru/vse-vidy-odezhdy-spisok/> (дата обращения: 13.10.2021)
- 10 Какие бывают стили одежды [Электронный ресурс]. – Режим доступа: <https://mavango.ru/blogs/blog/kakie-byvayut-stili-odezhdy> (дата обращения: 13.10.2021)

11 100 предметов идеального гардероба [Электронный ресурс]. – Режим доступа: https://ideal-garderob.ru/ideas/stilnie_sovety/100-predmetov-idealnogo-garderoba-2020/ (дата обращения: 13.10.2021)

12 50 вещей, которые должны быть в гардеробе [Электронный ресурс]. – Режим доступа: <https://www.yapokupayu.ru/blogs/post/50-veschey-kotorye-dolzhen-byt-v-garderobe-kazhdoy-zhenschiny> (дата обращения: 13.10.2021)

13 Базовый гардероб [Электронный ресурс]. – Режим доступа: <https://zen.yandex.ru/media/secretofbeauty/10-vescei-bazovogo-garderoba-2020-2021-5f903cf36ca69c780267cc89> (дата обращения: 13.10.2021)

14 H&M – интернет-магазин одежды [Электронный ресурс]. – Режим доступа: <https://hmonline.ru/> (дата обращения: 13.10.2021)

15 Wildberries – интернет-магазин одежды и обуви [Электронный ресурс]. – Режим доступа: <https://www.wildberries.ru/> (дата обращения: 13.10.2021)

16 OSTIN магазин одежды и аксессуаров, URL: <https://ostin.com/> (дата обращения: 13.10.2021)

17 Подбираем одежду исходя из ее температурных режимов [Электронный ресурс]. – Режим доступа: <https://reimashop.com.ua/podbiraem-verhnjuju-odezhdu-ishodja-iz-ee-temperaturnyh-rezhimov> (дата обращения: 15.10.2021)

18 Какую верхнюю одежду при какой температуре носить [Электронный ресурс]. – Режим доступа: <https://coppeer.ru/kakuju-verhnjuju-odezhdu-pri-kakoj-temperature-nosit/> (дата обращения: 15.10.2021)

19 Таблица соответствия одежды и температуры на улице [Электронный ресурс]. – Режим доступа: <https://www.babyblog.ru/user/Skorpion4ik/3016342> (дата обращения: 15.10.2021)

20 Kotlin for Android | Kotlin [Электронный ресурс] – режим доступа: <https://kotlinlang.org/docs/android-overview.html> (дата обращения: 18.09.21)

21 Swift. Язык программирования с открытым кодом [Электронный ресурс] – Режим доступа: <https://www.apple.com/ru/swift/> (дата обращения: 17.09.2021)

22 8 языков программирования для Android-разработчика [Электронный ресурс] – Режим доступа: <https://tproger.ru/articles/8-jazykov-programmirovanija-dlja-android-razrabotchika/> (дата обращения: 17.09.21)

23 Паттерны разработки: MVC vs MVP vs MVVM vs MVI [Электронный ресурс] – режим доступа: <https://habr.com/ru/post/344184/> (дата обращения: 20.09.21)

24 Лекция 5 по архитектуре андроид приложений. Паттерн MVP [Электронный ресурс] – режим доступа: <https://www.fandroid.info/lecture-5-on-the-architecture-of-the-android-application-mvp-pattern/> (дата обращения: 21.09.21)

25 Различия между MVVM и остальными MV*-паттернами [Электронный ресурс] – режим доступа: <https://habr.com/ru/company/mobileup/blog/313538/> (дата обращения: 21.09.21)

26 Ещё раз про семь основных методологий разработки / Хабр [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/edison/blog/269789/> (дата обращения: 07.02.2022)

27 Использование диаграммы вариантов использования UML при проектировании программного обеспечения / Хабр [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/566218/> (дата обращения: 08.02.2022)

28 Наука и образование: научно-техническое издание: Экспертная система «Кардиолог» [Электронный ресурс]. – Режим доступа: <http://engineering-science.ru/doc/95195.html> (дата обращения: 11.02.2022)

29 Room: Хранение данных на Android для всех и каждого / Хабр [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/336196/> (дата обращения: 17.12.2021)

30 SharedPreferences | Android Developers [Электронный ресурс]. – Режим доступа: <https://developer.android.google.cn/reference/android/content/SharedPreferences?hl=ru> (дата обращения: 25.10.2021)

31 Иванова, Г.С. Технология программирования: Учебник для вузов [Текст]. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. 238 с. (дата обращения: 22.02.22)

32 Иванова Г.С., Ничушкина Т.Н. Тестирование программного обеспечения: Методические указания по выполнению лабораторной работы по дисциплине "Технология разработки программных систем" [Текст]. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2019. – 13 с.

33 Автоматизация тестирования мобильных приложений: сравнение инструментов / Хабр <https://habr.com/ru/company/simbirsoft/blog/459292/> (дата обращения: 15.03.2022)

34 Автоматизация Android. Суперпростое руководство по созданию первого Espresso-теста / Хабр [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/otus/blog/472372/> (дата обращения: 16.03.2022)

35 Android: Тестируем с Espresso [Электронный ресурс]. – Режим доступа: <http://developer.alexanderklimov.ru/android/debug/espresso.php> (дата обращения: 16.03.2022)

36 ГОСТ 7.32-2001 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления [Электронный ресурс] – режим доступа: <https://docs.cntd.ru/document/1200026224> (дата обращения: 10.03.2022)

Приложение А
Техническое задание
Листов 8

Приложение Б
Руководство пользователя
Листов 16

Приложение В
Графический материал
Листов 6

Приложение Г
Фрагмент исходного кода
Листов 5

Приложение Д
Таблицы тестов функционального тестирования
Листов 10