САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ ПРОЦЕССОВ УПРАВЛЕНИЯ
КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Ревякина Вероника Яановна

Выпускная квалификационная работа бакалавра

Разработка мобильного приложения для мониторинга физической активности человека

Направление 01.03.02
Прикладная математика и информатика

Научный руководитель:
старший преподаватель
Малинин К. А.

Санкт-Петербург
2019

Содержание

Введение

В современном мире смартфоны стали неотъемлемой частью жизни каждого человека. Мы носим их с собой и используем различные функции каждый день. Согласно утверждению исследования [1] мобильные устройства используются в 3,5 раза чаще, чем персональные компьютеры (ПК). В наши дни смартфоны - это не только средство для совершения голосовых звонков или отправки текстовых сообщений, но и личное устройство, имеющееся почти у каждого взрослого человека и предоставляющее множество развлечений, а также персональной информации. В связи с этим появляются стали возможными разработки использования смартфона в качестве прибора для мониторинга физической активности человека.

Поскольку в последнее время всё более популярны тенденции ведения здорового образа жизни, можно с уверенностью отметить значительный скачок интереса общественности к оцениванию уровня своей физической активности. Кроме того мониторинг физической активности всегда являлся актуальной областью исследований в спортивной сфере для оценки результатов и разработки плана тренировок. Однако в последнее время эти вопросы стали невероятно популярны и среди обычных людей, так как наблюдение за собственным прогрессом является серьёзным мотиватором к поддержанию двигательной активности и улучшению физического состояния.

С давних пор проводилась разработка устройств, позволяющих проводить некоторый анализ физической деятельности человека. В данный момент эти устройства претерпели множество изменений и в последнее время стали востребованы не только среди спортсменов, но и среди обычных людей. Несмотря на то, что первые специализированные мобильные

приложения по оценке физической активности появились не так давно, они имеют довольно большую популярность [2]. По официальным данным, фитнес-приложения занимают первые строчки в магазинах мобильных приложений, ориентированных на определённую операционную систему.

Согласно статистике последних трёх лет, наибольшую долю на рынке занимает ОС Android, её часть составляет более 2/3 объёма рынка продаж. Количество активных пользователей устройств на базе Android в магазине Google Play превысило отметку в 1 млрд, что является абсолютным рекордом в сфере продаж мобильных приложений и даёт широкие возможности разработчикам в сфере предоставления продукта на рынке. Поэтому в настоящее время рынок мобильных приложений в области оценки двигательной активности человека стремительно расширяется и является одним из актуальных направлений. При этом индустрия мобильных приложений является одним из самых быстро развивающихся отраслей бизнеса в мире.

Следующим толчком к развитию данного направления стало появление новых технологий. Благодаря внедрению в устройства новейших датчиков, а также появлению продвинутых систем анализа и обработки данных, полученных с датчиков устройств, стала реальной разработка мобильных приложений для отслеживания физической активности. Эти приложения удовлетворяют потребность пользователя в максимально быстром и удобном получении информации о его двигательной активности, будучи при этом более энергоэффективными, что является одним из важных критериев для множества пользователей. Большинство существующих на рынке приложений, решающих задачу мониторинга физической активности, требуют от устройства пользователя наличия конкретной версии программного обеспечения или специфических технических характеристик устройства. Такой подход сильно снижает количество потенциальных пользователей. Также следует подчеркнуть, что многие существующие

программные продукты являются коммерческими и распространяются на платной основе.

Постановка задачи

Целью данной дипломной работы является разработка мобильного приложения для мониторинга физической активности человека. Для достижения поставленной цели решаются следующие основные задачи:

1) исследование подходов для оценки расхода калорий
2) изучение возможности использования данных высот при оценке расхода калорий
3) обзор библиотек для разработки под ОС Android
4) реализация алгоритма по расчёту калорий с учётом дополнительных параметров
5) разработка мобильного приложения, осуществляющего расчёт расхода калорий и хранение актуальной информации о физической активности пользователя

Обзор литературы

Среди используемой литературы можно выделить 4 главных источника:

1. Mifflin M.D. ,St Jeor S.T., Hill L.A., Scott B.J., Daugherty S.A., Koh Y.O. A new predictive equation for resting energy expenditure in healthy individuals ,1990

В данной статье был проведён множественный регрессионный анализ, в ходе которого было выявлено, что такие параметры как вес, рост, пол и

возраст оказывают наибольшее влияние на точность оценочной модели скорости базового расхода калорий (RMR). В результате было разработано прогностическое уравнение Миффлина-Сент-Джеора для расчёта скорости базового расхода калорий (RMR) на основании данных признаков.

2. Frankenfield D., Roth-Yousey L., Compher C. Comparison of predictive equations for resting metabolic rate in healthy nonobese and obese adults: a systematic review, 2005

В данной работе был осуществлён сравнительный анализ наиболее популярных уравнений для измерения скорости метаболизма в покое (RMR). При исследовании точности на основе различных метрик уравнение Миффлина-Сент-Джеора было выявлено как наиболее надежное, прогнозирующее RMR с точностью в пределах 10% от измеренного у большего числа людей, что является наилучшим результатом в сравнении с другими решениями.

3. Peter Kokkinos, Leonard A. Kaminsky, Ross Arena, Jiajia Zhang, Jonathan Myers New Generalized Equation for Predicting Maximal Oxygen Uptake (from the Fitness Registry and the Importance of Exercise National Database,2017

В данной статье проводится многопараметрический линейный регрессионный анализ для выявления основополагающих признаков, учёт которых в большей степени отражается на точности модели прогнозирования показателя интенсивности физической активности (VO_2 max). Получены следующие результаты: во-первых, точность оценки интенсивности физической активности значительно улучшается при учёте признаков скорости и взаимодействия скорости и угла наклона; во-вторых, разработано (FRIEND) уравнение для прогнозирования значений VO_2 max.

4. Wang Y, Zou Y, Henrickson K, Wang Y, Tang J, Park B-J Google Earth elevation data extraction and accuracy assessment for transportation applications, 2017

В данном исследовании производится сравнительный анализ методов получения данных о высоте над уровнем моря на основе известных географических координат с точки зрения простоты их использования и точности. В результате на основании оценок точности при помощи различных метрик было дано заключение, что Google Earth является проверенным и надёжным источником данных по высоте.

Глава 1. Обзор существующих подходов мониторинга и расчёта показателей физической активности

1.1 Методы расчёта базового расхода калорий

Базальная скорость метаболизма (BMR) - это скорость расхода энергии, необходимая для поддержания базальных функций тела или, другими словами, это минимальные затраты энергии, требуемые для выживания [3].

Скорость метаболизма в покое (RMR) [4] тесно связана с основной скоростью метаболизма (BMR) и представляет собой количество энергии, необходимое для поддержания нормальной метаболической активности организма, такой как дыхание, поддержание температуры тела и пищеварение. В частности, это количество энергии, необходимое в состоянии покоя без выполнения дополнительной активности. Потребляемая энергия достаточна только для функционирования жизненно важных органов, таких как сердце, легкие, нервная система, почки, печень, кишечник, половые органы, мышцы и кожа.

На протяжении многих лет разрабатывались различные уравнения для оценки BMR и RMR, они были основаны на фактических измерениях BMR населения, учитывая различные персональные параметры. Эти уравнения, как правило, считаются настолько точными, насколько это возможно, без предоставления дополнительной информации, такой как частота сердечных сокращений, процентное содержание мышечной массы тела и других более сложных измерений. Однако и эти прогностические уравнения, могут генерировать ошибки, причём достаточно большие, оказывая сильное влияние на конечный результат. Вследствие чего возникла необходимость оценки точности этих разработок и сравнения их друг с другом для применения на практике. Четыре уравнения прогнозирования были определены как наиболее часто используемые в клинической практике, такие как Харрисона-Бенедикта, Миффлина-Сент-Джеора, Оуэна и уравнение Всемирной организации здравоохранения (ВОЗ). Причём прогностическое уравнение Миффлина-Сент-Джеора было разработано в исследовании [5], в ходе применения множественного регрессионного анализа для определения взаимосвязи между скоростью базового расхода калорий (RMR) и весом, ростом и возрастом у мужчин и женщин. В результате чего формулы (1) и (2) были разработаны.

Несколько позднее были произведены практические исследования [6] , в ходе которых группа экспертов выявила наиболее точные уравнения для

измерения скорости метаболизма среди перечисленных ранее наиболее используемых уравнений, на основе данных различных людей как без ожирения, так и с ожирением, а также у лиц различных этнических групп, возрастных групп и т.д. Из представленных вариантов по результатам исследования, указанного выше, уравнение Миффлина-Сент-Джеора было выявлено как наиболее надежное, прогнозируя RMR с точностью в пределах 10% от измеренного у большего числа людей, что является наилучшим результатом в сравнении с любыми другими уравнениями.

Прогностическое уравнение для расхода энергии в состоянии покоя Миффлина-Сент-Джеора, согласно формуле, полученной в исследовании [5], выглядит следующим образом:

· Для женщин:

RMR= 9,99 * вес(кг) + 6.25 * рост (см) – 4,92 * возраст – 161 (1)

· Для мужчин:

RMR= 9,99 * вес (кг) + 6.25 * рост (см) – 4,92 * возраст + 5 (2)

На основании выводов исследования [6], было принято решение использовать в данном дипломном проекте уравнение Миффлина-Сент-Джеора для оценки значений метаболизма в покое (RMR). Главным критерием выбора являлось то, что в сравнении с другими протестированными уравнениями данные формулы считаются самыми точными на сегодняшний день. Кроме того данное уравнение является наиболее удобным в использовании, с точки зрения пользователей, поскольку не требует проведения сложных измерений дополнительных параметров, таких как мышечная масса тела (по сравнению с уравнением Каннингема, например).

Таким образом, требуется внедрить данные формулы (1),(2) при разработке, учитывая четыре индивидуальных параметра пользователя, таких как пол, вес(кг),рост(см) и возраст.

1.2 Расчёт добавочного расхода калорий

Физическая активность оказывает сильное влияние на скорость расхода энергии, и физические упражнения, которые обладают высокой интенсивностью, значительно увеличивают расход энергии по сравнению с базовым RMR. Интенсивность — это темп занятий физической активностью или величина усилий, необходимых для осуществления какого-либо вида активности или упражнения. Её можно охарактеризовать словами: насколько напряженно работает человек для выполнения определенного вида активности [7]. Физическая активность людей варьируется по степени интенсивности. Влияние на расход калорий, которое оказывает та или иная физическая активность, также зависит от имеющегося у человека опыта в выполнении физических упражнений и относительного уровня его физического состояния, а также возраста, пола и других параметров. Для выражения степени интенсивности физической активности в теории исследования расхода калорий широко используется метаболический эквивалент (MET). С одной стороны, MET–это объективная мера отношения скорости, с которой человек расходует энергию, относительно массы этого человека при выполнении определенной физической активности по сравнению с эталоном, установленным условно в 3,5 мл. кислорода на килограмм в минуту [8]. С другой стороны, один MET – это количество энергии, затрачиваемое человеком в состоянии покоя (RMR) и эквивалентное сжиганию 1 ккал/кг*мин. Соответственно, в зависимости от подхода к рассмотрению MET значения существуют различные способы его измерения и стандартизации. Ещё одним показателем, с помощью которого может быть выражена интенсивность физической активности, с точки зрения выносливости человека, является максимальное потребление кислорода (VO_2 max) [9]. Данное значение широко используется в кардиологии и выражается как относительная скорость в миллилитрах кислорода на

килограмм массы тела в минуту (мл /кг*мин). Точные измерения данного показателя крайне важны с научной точки зрения и производятся специалистами и врачами при использовании специализированно технического оборудования, на основе результатов поэтапного тестирования на нагрузку по правилам разработанных протоколов, вследствие чего производится оценка VO_2 max. Однако с точки зрения данной дипломной работы важно, что показатели MET и VO_2 max являются взаимозаменяемыми в силу того, что 1 ккал/кг*мин=1 мл/кг*мин и отображают интенсивность физической активности с различных сторон. Соответствующее преобразование может быть осуществлено на основании следующей формулы:

1 MET (ккал/кг*мин) = 3,5*(1 VO_2 max(мл/кг*мин) (3).

Стоит также отметить, что в силу существующих тенденций в подходах расчёта расхода калорий в качестве основного показателя интенсивности в данной работе будет рассматриваться значение MET. Итак, расход энергии во время выполнения какой-либо активности может быть выражен при помощи умножения стандартного RMR, высчитанного для каждого человека, на это специальное значение метаболического эквивалента MET, отражающее интенсивность той или иной активности.

Существуют опубликованные таблицы, в которых различные виды деятельности подразделены на три основные категории в зависимости от интенсивности. В первую категорию попадает физическая деятельность, обладающая лёгкой интенсивностью со значениями MET < 3, во второй категории упражнения умеренной интенсивности, для которых значения 3<MET<6, и в последней категории отображены активности высокой интенсивности с MET>6. Также соответственно для каждой физической активности в категории определены частные значения MET. Один из вариантов такой таблицы продемонстрирован в таблице 1. Подобное табличное представление характерно также и для показателя VO_2 max. Использование данных из таблицы даёт возможность определить

интенсивность конкретной физической активности. Например, при ходьбе со скоростью 5 км в час, которой соответствует табличное значение 3,6 МЕТ, человек расходует в 3,6 раза больше энергии в сравнении с состоянием покоя, выраженным RMR. И так далее, шкала МЕТ значения активности колеблется от 0,9 (в спящем режиме) до 23 при сильной нагрузке.

Физическая активность	**Значение МЕТ**
Лёгкая интенсивность	**< 3**
Сон	0.9
Состояние покоя	1.0
Медленная ходьба (2,7 км/ч) по ровной местности, прогулочный шаг	2.3
Ходьба со скоростью 4 км/ч	2.9
Умеренная интенсивность	**>3 и <6**
Езда на велосипеде с малыми усилиями (100 Вт) по ровной местности	3.0
Ходьба со скоростью 4,8 км/ч	3.3
Гимнастика, йога, приседания	3.5
Интенсивная ходьба со скоростью 5,5 км/ч	3.6
Езда на велосипеде со скоростью менее 16 км/ч под малым уклоном	4.0
Езда на велосипеде (500 Вт) с умеренной интенсивностью	5.5
Высокая интенсивность	**>6**

Бег трусцой, обычный	7.0
Интенсивные физические упражнения	8.0
Бег трусцой со скоростью 5,6 км / ч	8.8
Прыжки со скакалкой 66 раз в минуту	9.8
Прыжки со скакалкой 80 раз в минуту	10.3
Прыжки со скакалкой 100 раз в минуту	11.0
Бег со скоростью 11 км/ч	11.2

Таблица 1. Значения метаболических эквивалентов (MET) для различных видов физической активности в соответствии с тремя категориями

Итак, расчёт итогового расхода калорий производится по следующей формуле (4),описанной в данной книге [10]:

$$C=T*MET*(RMR/1440) \quad (4)$$

где: C-расход калорий (ккал),T- Продолжительность активности (мин) , MET-Значение MET интенсивности активности (ккал/кг/мин), RMR- Скорость метаболизма в покое для данного человека (RMR/1440) (ккал/мин).

Деление значения RMR на 1440 производится для перевода из единиц измерения (ккал/день) в (ккал/мин), 1 день=24 часа * 60 минут=1440 минут.

Рассмотрим пример расчёта расхода энергии при беге для человека с RMR, равным 1663 ккал/день. Соответственно, RMR в минуту равняется: 1663/1440=1.16 ккал/мин. Предположим, что человек занимался бегом в течение 30 минут, в таблице представлено значение MET для физической активности бег, которое равняется 8.0. Таким образом, его энергетические затраты по формуле (3) могут быть оценены как : 30(минут)* 1.16(RMR в минуту)*8.0(MET для бега)=278,4 ккал.

Однако проблемой применения табличного подхода на практике является существование огромного количества различных таблиц интенсивности активностей с дискретными значениями MET. Это является особенно

актуальным вопросом определения интенсивности таких активностей, как ходьба и бег. Трудность заключается в проведении оптимального объединения значений MET данных таблиц, полученных на основе множества исследований. В то время как в каждой таблице данные могут быть представлены в различных единицах измерения и варьироваться в зависимости от абсолютно разнообразных дополнительных параметров. Примером может служить таблица 2, в которой в качества параметра для определения значения MET активности бега указана скорость, а для прыжков на скакалке количество повторов в минуту. Возможным решением данной проблемы являются ограничения, налагаемые на значения интенсивности при использовании одной таблицы. Данный подход может использоваться для определения значений MET менее исследованных видов активностей. Однако при слишком малом наборе значений MET и без учёта дополнительных параметров для таких видов физической активности, как ходьба или бег, точность расчётов снизится в несколько раз. Таким образом, было заключено, что применение табличного подхода для расчёта интенсивности бега или ходьбы на практике крайне затруднительно. В связи с этим была выявлена необходимость использования некоторых регрессионных уравнений, которые предсказывали бы значение интенсивности MET. Причём в данных уравнениях в силу описанных выше причин обязательно должны быть учтены некоторые основополагающие признаки, помогающих различать подвиды активностей. Такими для ходьбы и бега, например, являются скорость, степень энергозатратности и условия окружающей среды, в которой совершается активность. Однако проблемой применения табличного подхода на практике является существование огромного количества различных таблиц интенсивности активностей с дискретными значениями MET. Это является особенно актуальным вопросом определения интенсивности таких активностей как ходьба и бег. Трудность заключается в проведении оптимального объединения значений MET данных таблиц, полученных на основе множества исследований. В то время как в

каждой таблице данные могут быть представлены в различных единицах измерения и варьироваться в зависимости от абсолютно разнообразных дополнительных параметров. Примером может служить таблица 1, в которой в качества параметра для определения значения MET активности бега указана скорость, а для прыжков на скакалке количество повторов в минуту. Возможным решением данной проблемы являются ограничения, налагаемые на значения интенсивности при использовании одной таблицы. Данный подход может использоваться для определения значений MET менее исследованных видов активностей. Однако при слишком малом наборе значений MET и без учёта дополнительных параметров для таких видов физической активности как ходьба или бег, точность расчётов снизится в несколько раз. Таким образом, было заключено, что применение табличного подхода для расчёта интенсивности бега или ходьбы на практике крайне затруднительно. В связи с этим была выявлена необходимость использования некоторых регрессионных уравнений, которые предсказывали бы значение интенсивности MET. Причём в данных уравнениях в силу описанных выше причин обязательно должны быть учтены некоторые основополагающие признаки, помогающих различать подвиды активностей. Такими для ходьбы и бега, например, являются скорость, степень энергозатратности и условия окружающей среды, в которой совершается активность.

Выбор основополагающих признаков в данной работе происходит на основании результатов исследования [11]. В ходе которого был проведён многопараметрический линейный регрессионный анализ для выявления наиболее важных признаков, учёт которых в большей степени отражается на точности модель прогнозирования VO_2 max. Были рассмотрены такие признаки активности как скорость, угол уклона, взаимодействие скорости и угла уклона, частота сердечных сокращений в покое, индекс массы тела, артериальное давление в покое и т.д. В результате применения к модели нескольких различных критериев для оценки точности, было выявлено, что эффективность работы модели значительно улучшается при введении

признаков скорости и взаимодействие скорости и угла наклона, причём добавление оставшихся признаков не оказывало заметного влияния на точность модели. Таким образом, была доказана важность и эффективность учёта скорости и взаимодействия скорости с углом наклона для наиболее точных оценок интенсивности активности. Вследствие чего в данном дипломном проекте указанные признаки рассматриваются при прогнозировании и необходимой корректировке значений интенсивности MET.

Следовательно, интенсивность упражнений также меняется в зависимости от скорости или уклона поверхности, в условиях выполнения физической активности. Итак, для расчёта скорости и угла наклона необходимым является вычисление дистанции, преодоленной во время выполнения активности. На основании результатов исследования [12], в котором был произведён сравнительный анализ различных алгоритмов вычисления расстояния на поверхности Земли с учётом данных координат двух точек, в данном дипломном проекте было решено осуществлять вычисление пройденной дистанции при помощи метода, основанного на формулах Винсенти [13]. Так как в силу использования в данном методе точной эллипсоидальной модели Земли, как было указано в предоставленном выше исследовании, он является более точным в сравнении с методами, предполагающими сферическую Землю, такие как расстояние по большому кругу, например метод Хаверсайна. После на основе данных о длительности активности, а также с применением вычисленной по методу Винсенти дистанции, производится расчёт скорости как отношение преодолённого расстояния к длительности активности. В данном дипломном проекте было принято решение отказаться от расчёта скорости совершаемой активности на основе интегрирования ускорения, полученного с сенсора акселерометра в силу зашумленности его данных, а также слабой отказоустойчивостью. Соответственно расчёт угла также осуществляется при использовании вычисленной разности высот в некоторых точках активности и на основе

посчитанной выше дистанции между этими точками, угол вычисляется как арктангенс данных значений для выражения в градусной мере.

Также, как упоминалось ранее, в силу трудностей использования табличных данных MET для оценки интенсивности ходьбы и бега возникает необходимость использования прогнозирующих уравнений. Однако уравнений, предсказывающих значения MET в чистом виде не существует, вследствие чего рассматриваются решения для прогнозирования оценок смежного с MET показателя оценки эффективности VO_2 max, которые впоследствии преобразуются к значениям MET при помощи деления на 3,5 в соответствии с формулой (3). Итак, основными уравнениями для прогнозирования значений VO_2 max являются разработки Американского колледжа спортивной медицины (ASCM). Эти уравнения варьируются в зависимости от учитываемых параметров таких как скорость угол наклона, частота и высота шага, мощность (Вт) и т.д. Тем не менее наиболее часто используемые уравнения ASCM для прогнозирования меры интенсивности ходьбы и бега были разработаны почти 4 десятилетия назад, то есть подсчёт производился на менее точном оборудовании. И при этом также при разработке уравнений использовались данные в большинстве только молодых спортсменов (19-26 лет). Вследствие чего, как было выяснено при множественных исследованиях (в том числе [11]) точности оценок данных уравнений, в большинстве случаев они завышают реальное значение VO_2 max. Соответственно в связи с описанными причинами в данной работе было принято решение использовать Fitness Registry and the Importance of Exercise National Database (FRIEND) уравнение для прогнозирования значений VO_2 max, разработанного в выше упомянутом исследовании [11]. Общее значение ошибки данного уравнения в сравнении с ASCM, проведённом на большом наборе данных различных людей, в 4 раза ниже, а также в данном решении производится учёт скорости и угла наклона. Основная формула FRIEND сохраняет общую структуру уравнений ASCM, предназначенных для оценки интенсивности бега и ходьбы VO_2 max= Resting Component + Horizontal

Component + Vertical Component, однако корректирует значения коэффициентов A и B:

$$VO_2 \text{ max (мл/кг*мин)} = 3{,}5 + \text{speed}*(A) + \text{grade}*\text{speed}*(B) \quad (5)$$

где speed (м/мин) значение скорости бега либо ходьбы, grade(относительные градусы) значение угла наклона, A=0,17 коэффициент влияния скорости, B=0,79 коэффициент влияния взаимодействия угла наклона и скорости. Рассмотрим пример, демонстрирующий расчёт значения оценки интенсивности бега со скоростью 134 м/мин под уклоном в 5%:
VO2 max= 3,5 + 134*0,17+0,05*134*0,79=31,573. Далее происходит расчёт MET в соответствии с (3), значение MET примерно равняется 9, что примерно соотносится со значением в таблице 1. После этого происходит расчёт калорий на основании формулы (4) и использовании вычисленного значения MET в качестве параметра интенсивности.

При этом для оценки меры интенсивности состояния покоя или более энергозатратных активностей, таких как езда на велосипеде или быстрый бег, в зависимости от доступности данных используются линейные приближения табличных значений MET, а также корректировка домножаемых коэффициентов в формуле (5) с целью увеличения влияния скорости и угла на конечную оценку эффективности. Примером коэффициентов для быстрого бега, умеренной езды на велосипеде является уравнение ASCM, в котором A=0,2 и B=0,9.

Таким образом, по изложенному выше принципу будет произведён расчёт расхода энергии пользователя в мобильном приложении данного дипломного проекта, при этом значения интенсивности выявленных видов активности пользователя, таких как, езда на велосипеде, состояние покоя, бег, ходьба будут корректироваться в зависимости от скорости выполнения активности и перепада высот.

1.3 Обзор существующих приложений для мониторинга физической активности человека

При создании оригинального приложения одним из ключевых моментов является поиск и рассмотрение сходных по тематике уже существующих программных продуктов, предоставляющих похожую функциональность. Именно поэтому следует провести анализ наиболее популярных из этих приложений с целью выявления их плюсов и минусов, с точки зрения пользователя. Вследствие чего на основе результатов проведённого анализа необходимо попытаться устранить как можно больше недостатков и учесть положительные моменты в создаваемой разработке.

Google Fit — одна из последних разработок компании Google в сотрудничестве с Всемирной организацией здравоохранения и Американской кардиологической ассоциацией. Это не просто приложение, а облачный сервис, ведущий наблюдение за активностью человека и некоторыми характеристиками его организма. Приложение совместимо с рядом планшетов и смартфонов, но, как отмечает производитель, наибольшая точность результатов достигается при интеграции Google Fit с сервисами фитнес-трекеров, умных часов и других специализированных устройств. Это обстоятельство скорее является минусом, поскольку не все пользователи обладают технической возможностью использования данных устройств. Во время тренировки приложение ведет учет пройденного расстояния, количества сделанных шагов, сожженных калорий и другой информации. Основные функции приложения продемонстрированы на рисунке 1. На данный момент приложение имеет рейтинг 3,8 в GooglePlay [14] , некоторые пользователи жалуются на утечку данных и многие разочарованы в связи с последними обновлениями. Также минусом является доступность приложения только для последних версий операционных систем, что сильно снижает охват пользователей.

Рис. 1. Демонстрация приложения Google Fit

Главным конкурентом Google Fit, ранее предназначенного только для платформы Android (с 24 апреля 2019 года стало доступно для iOS [15]) ,является приложение Apple Health под iOS в App Store. Приложение «Здоровье» помогает следить за своим самочувствием и достигать поставленных целей. В приложении «Здоровье» ведётся учёт четырёх категорий: Активность, Сон, Осознанность и Питание, данные в них поступают с сенсоров устройства или сторонних приложений. Отличительной чертой данного приложения является обязательная интеграция со множеством других приложений, например, для отслеживания сна и рекомендательная система, заложенная в функциональность. Основные категории и функциональные возможности приложения, предоставляемые пользователю, отображены на рисунке 2. Соответственно основным недостатком, с точки зрения пользователя, является необходимость самостоятельного внесения в систему большого количества информации, а также использование множества сторонних приложений, отсутствие этих данных приводит к невозможности использования множества функций приложения. Также минусом данного приложения является то, что в нём не производится распознавание пользовательской активности, подсчёт калорий

и других параметров при отсутствии взаимодействия со сторонними решениями или устройствами, а осуществляется и отображается лишь подсчёт шагов, при любой активности, похожей на шагание. Данное приложение является встроенным ,начиная с версии iOS 8, то есть присутствует на каждом устройстве.

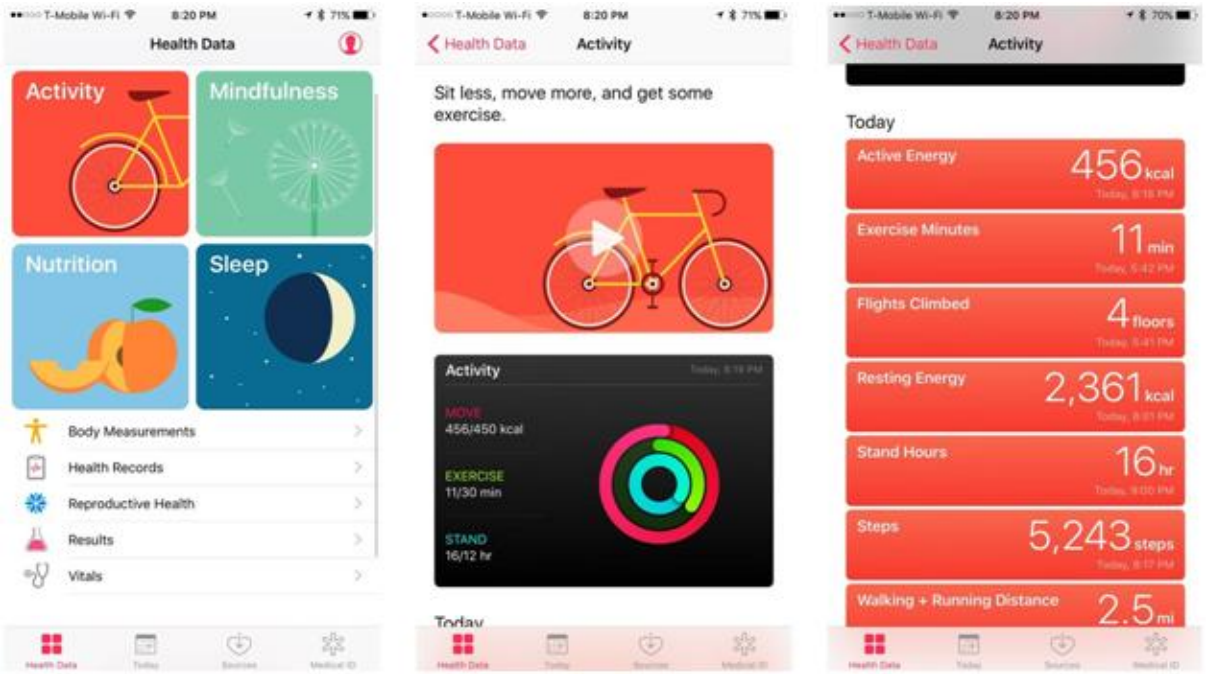

Рис. 2. Демонстрация приложения Apple Health

Ещё одним популярным продуктом является Endomondo–это мобильное приложение, доступное для iPhone, Android и Windows Phone, предназначенное в большей степени для отслеживания тренировок на свежем воздухе. Очевидно, одним из главных плюсов является его доступность на множестве операционных систем и устройств. Приложение отслеживает продолжительность вашей активности, расстояние, темп, скорость, предоставляет возможности для отображения маршрута на карте в реальном времени и многое другое. В приложении также предусмотрены такие функции, как общение с друзьями, что позволяет делиться с ними своими результатами, существует аудиотренер и различные мотиваторы. Описанные выше возможности, а также пользовательский интерфейс приложения можно наблюдать на рисунке 3. Однако очевидным недостатком для большинства

пользователей станет сильно ограниченный бесплатный функционал, чтобы воспользоваться всеми преимуществами приложения необходимо премиум-членство. Также пользователи отмечают в качестве минуса отсутствие русского языка и большой объём, занимаемый данным приложением в памяти устройства.

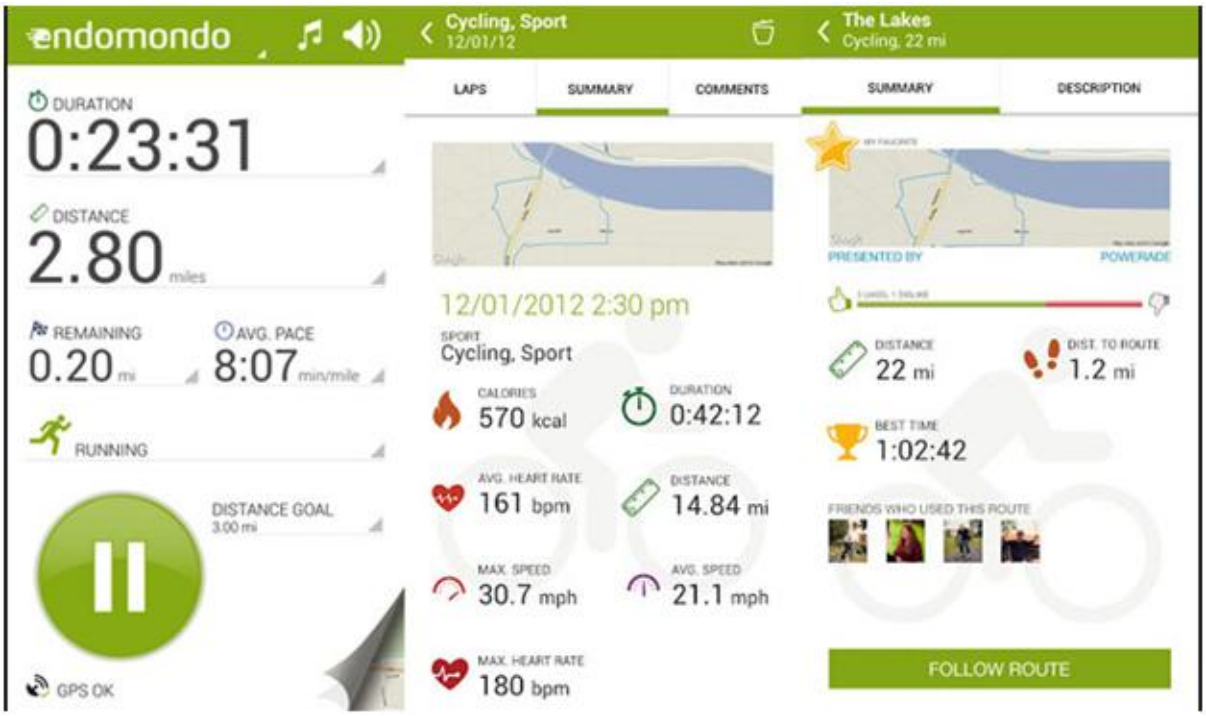

Рис. 3. Демонстрация приложения Endomondo

Таким образом, анализ приложений с похожей функциональностью показал, что многие из них накладывают слишком жесткие ограничения на версию операционной системы, что сильно сужает охват пользователей, которые могут использовать данное приложение. Также минусом некоторых приложений является обязательное внесение большого числа пользовательских данных или необходимая интеграция со сторонними устройствами или приложениями, при отсутствии которой приложение теряет основные функциональные возможности. При этом во многих существующих решениях используется слишком большой объем памяти или оказывается сильное влияние на заряд устройства, что делает их неудобными в использовании и отталкивает большинство пользователей.

1.4 Требования к создаваемому приложению

Требуется разработать оригинальное приложение, которое позволяло бы проводить расход калорий за определённый период времени на основании небольшого набора пользовательских данных и информации о физической активности пользователя в данный период времени. Отличительной чертой разработанного продукта является учёт дополнительных параметров, влияющих на расход калорий при совершаемой активности, таких как скорость и перепад высот. Важным требованием к разрабатываемому приложению также является снижение его влияния на заряд батареи устройства.

Кроме того, следует выбрать оптимальную минимальную версию операционной системы, чтобы обеспечить доступность приложения максимальному числу пользователей.

Глава 2. Обзор используемых программных продуктов

2.1 Получение данных о местоположении пользователя

Важно отметить, что в настоящее время в связи с одновременным развитием стандартов технологического оборудования и интерфейсов прикладного программирования (API) появляется возможность получить такую точность определения местоположения, которая ранее была недостижимой при использовании смартфонов. Как утверждается на конференции Google I/O 2018 года, этот год провозглашается временем развития технологий позиционирования. Однако получение информации о местоположении пользователя с помощью мобильного устройства всё ещё является достаточно сложной задачей. Одной из основных проблем для разработчика является вариативность используемых технологий определения местоположения. Выбор используемого подхода определяется специализацией задачи, внешними факторами среды и тем, каким данным доверять в конкретный момент времени. Сложность данной задачи

обусловлена также и мобильностью пользователя, поскольку местоположение устройства изменяется и возникает необходимо учитывать его перемещение, периодически переоценивая значение координат. Однако при осуществлении этой переоценки запрос к получению данных от системных сервисов должен производиться оптимальное количество раз, так как одним из критериев эффективности работы приложения является потребление энергии аккумулятором устройства. Кроме того, в каждый момент времени требуется принимать во внимание погрешности полученных результатов. Эти моменты могут затруднить получение надежных данных при определении местоположения пользователя.

Итак, существует несколько основных технологий для получения информации о местоположении пользователя, которые в основном подразделяются по признаку используемого источника данных [16]. Первый поход с использованием глобальных навигационных спутниковых систем GNSS является наиболее точным. GNSS включает в себя региональные системы GPS и ГЛОНАСС, так как на данный момент только эти две спутниковые системы обеспечивают полное и бесперебойное покрытие земного шара. Использование данной технологии предполагает наличие на устройстве приёмника сигнала, GPS-модуля, которым оборудованы практически все современные смартфоны. Принцип работы спутниковых систем навигации заключается в том, приемник измеряет задержку распространения сигнала от спутников, положение которых известно с большой точностью, до приемника, и на основании этой задержки сигнала, умноженной на скорость света, получает конечный результат. Таким образом, на основании данных о расстоянии до нескольких спутников системы с помощью обычных геометрических построений вычисляется положение устройства в пространстве. Однако необходимо учитывать, что данная технология работает лучше на открытых участках пространства, так как в помещении или среди высотных зданий сигнал от спутника много раз преломляется, вследствие чего сильно страдает точность полученных

результатов. Причем в сравнении с остальными походами GPS возвращает данные о местоположении с некоторой задержкой, так как данные с нескольких спутников должны достигнуть устройства и быть комбинированы между собой. Также минусом использования GPS является большее влияние на заряд батареи устройства.

Соответственно, вторым основным походом является метод получения информации о местоположении пользователя при помощи сетевых технологий, основываясь на данных вышек сотовой связи и сигналах Wi-Fi. Сетевые технологии были разработаны за много лет до широкого распространения GPS на мобильных телефонах. Принцип работы технологии определения местоположения на основе вышек сотовой связи заключается в том, что мобильное устройство, которое должно быть оборудовано модулем сотовой связи (SIM) имеет информацию о приемопередатчике базовой станции, которым оно обслуживается, и далее на основе базы данных координат передатчиков базовых станций есть возможность получить данные о местоположении устройства. Последующее приближение может быть сделано путем интерполяции сигналов между соседними антенными вышками. Позиционирование в сети Wi-Fi происходит схожим образом на основе передачи сигнала от Wi-Fi адаптера устройства до ближайших точек доступа и дальнейшей обработки результатов, посредством использования информации о затухании сигнала с помощью различных алгоритмов, таких как, например, весовые алгоритмы, триангуляция, механизм наибольшей мощности и тд. Для повышения точности позиционирования в сети Wi-Fi точки доступа должны быть расположены как можно чаще, так как угасание сигнала и расстояние от точки доступа имеют экспоненциальную зависимость. Данный подход определения местоположения пользователя отлично работает как в помещении, так и на улице, при условии, что плотность антенных вышек (базовых станций) достаточно высока или наличия вблизи точек доступа. Также преимуществом использования сотовых вышек и WI-FI является то, что данные технологии быстрее

реагируют на изменения положения пользователя в сравнении с подходом на основе GPS и потребляет меньше энергии аккумулятора устройства. Однако в связи с указанными выше необходимыми условиями работы, вдали от населенных пунктов и в сельской местности точность полученных результатов значительно снижается. Минусом использования сетевых технологий также является тесное взаимодействие устройства с поставщиком услуг, поскольку оно влечет за собой установку определённого аппаратного и программного обеспечения.

Однако в современном мире в связи с различными техническими характеристиками разных устройств, а также в условиях постоянного изменения факторов окружающей среды, влияющих на точность результатов, оптимальным является объединение указанных выше подходов. Гибридные системы определения местоположения используют комбинацию сетевых и спутниковых технологий для определения местоположения с учётом их минусов и плюсов, подчёркнутых выше. При данном подходе оптимальным образом реализуются проверки доступности GPS данных и точек доступа Wi-Fi и сотовых вышек в каждый момент времени, а далее на основе анализа данных производится выбор наиболее подходящего источника. Одним из важных факторов эффективности работы данного подхода является максимальное снижение количества обращений к системным сервисам и применение надёжного алгоритма интеллектуального комбинирования различных сигналов с целью минимизации потерь в точности и ошибок.

Таким образом, ввиду описанных выше плюсов и минусов в данной работе было принято решение использовать гибридный подход комбинации GNNS и сетевых технологий для получения данных широты и долготы местоположения пользователя. Этот подход реализован в сервисах Google Play при помощи API поставщика объединённого местоположения (Fused Location Provider Client) [17]. Он удовлетворяет указанным выше требованиям и интеллектуально комбинирует различные сигналы для предоставления информации о местоположении, что было

продемонстрировано на конференции Google I/O. Также это решение позволяет максимально экономно расходовать заряд аккумулятора устройства и удовлетворяет ограничениям на количество запросов, наложенных новейшими версиями операционных систем. При этом согласно исследованиям, представленным на конференции Google I/O средняя точность определения местоположения при использовании Fused Location Provider Client на данный момент колеблется в пределах от 3 до 5 метров, но в ближайшем будущем, благодаря внедрению технологий Dual Frequency и Wi-Fi RTT, станет возможным снижение ошибки до 1 метра.

2.2 Получение данных о текущей высоте над уровнем моря

Существует несколько методик для получения текущей высоты над уровнем моря, основываясь на информации о местоположении на поверхности Земли. Наиболее простым решением является считывание данных о высоте (altitude) с GPS-приёмника. Однако этот подход практически не применяется в силу того, что в большинстве случаев традиционные GPS-приёмники устройств содержат данные только о двумерных гео-координатах, и не предоставляют информацию о высоте. Либо же приёмник регистрирует высоту, но данные сильно теряют в точности в связи с плохим спутниковым сигналом или слабыми техническими характеристиками устройства. За последние несколько десятилетий новые методы обработки данных и технологии сбора, хранения, запросов и визуализации данных значительно увеличили доступность данных о высотах. Соответственно, вторым подходом является использование информации на основе цифровой модели рельефа (DEM). Географический охват различных баз данных, составляющих DEM, продемонстрирован на рисунке.

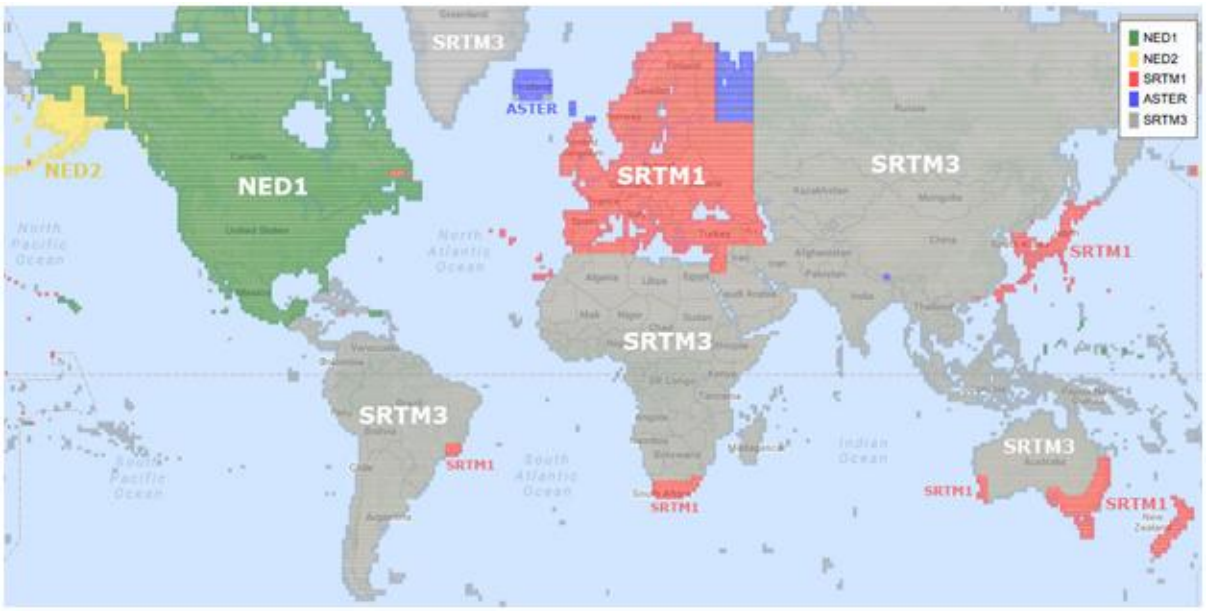

Рис. 4. Демонстрация географического охвата различных баз данных (DEM) по состоянию на март 2017 года

В настоящее время основными источниками наборов данных DEM о высотах над уровнем моря являются [18] национальный набор данных рельефа США, созданный Геологической службой США (NED USGS) ,глобальный набор данных GTOPO30,обладающий слабой точностью на некоторых местностях, набор данных высот из программы топографической радиолокационной съемки "Шаттл" (SRTM) от НАСА, полученный с космического аппарата "Спейс шаттл", ACTEP GDEM ,который является совместным продуктом НАСА и Министерства Японии, наборы данных определения высоты и дальности (LIDAR), а также и другие. Проблемами использования наборов данных DEM для определения высот являются низкое разрешение некоторых наборов данных, вертикальная неопределенность источников для нескольких высотах, а также различный охват территорий, в силу чего сложно производить объединение различных источников. Итак, третьим походом к определению высоты на поверхности Земли является использование проекта компании Google, Google Earth. Google Планета Земля(GE) представляет собой виртуальную 3D модель

Земли и позволяет использовать всевозможную функциональность, в том числе получение данных высот для любой точки мира. Исследовательская группа провела сравнительный анализ [19] подходов получения данных высот при помощи Google Eearth и одного из набора данных DEM USGS NED. Результат сравнения показывает, что данные высот, извлеченные посредством GE, по крайней мере так же точны, как данные NED USGS, а для некоторых территорий средняя ошибка данных высот GE значительно меньше данных NED USGS. Исходя из этого, был сделан вывод, что использование Google Earth предпочтительнее в сравнении с наборами данных DEM, так как было выявлено, что GE также является более простой в использовании и включает в себя районы, в которых данные DEM имеют малую точность. Однако для рассмотрения Google Earth в качестве ценного источника общенациональных данных о высоте, необходимо также было провести оценку точности данных полученных при помощи данного подхода в сравнении с реальными известными значениями высот. Эти дополнительные испытания и анализ точности были проведены в указанном исследовании [19], где в качестве эталонных значений использовались реальные показатели GPS, собранные в 2013 году в США. В целом был произведён подсчёт оценок точности на основании различных метрик и значение минимальной средней ошибки (MAE) составило 10,72 метра, а измеренное среднеквадратичное отклонение 22,31 метра. На основании полученных результатов, было дано заключение, что Google Earth является проверенным и надёжным источником данных по высоте. С учётом описанных выводов в данной дипломной работе было принято решение опираться в большей степени на получение данных о высоте на основе Google Earth.

Данный подход реализуется при помощи веб-сервиса Google Elevation API [20], который предоставляет простой интерфейс для получения данных о высоте при помощи запросов. Кроме того, данный API даёт возможность запрашивать выборочные данные высот вдоль маршрутов, что позволяет

рассчитывать изменения высот вдоль маршрутов. Служба Elevation предоставляет данные о высотах для всех местоположений на поверхности земли, включая глубины океана (которые возвращают отрицательные значения). В тех случаях, когда Google не располагает точными измерениями высоты в указанном месте по соответствующей долготе и широте, служба будет интерполировать и возвращать усредненное значение, используя четыре ближайших местоположения, значения высот выражаются относительно высоты над уровнем моря. Итак, данные о высотах получают посредством запроса к Elevation API в виде URL-адреса, в котором данные о местоположении можно указать одним из двух способов: как набор из одного или нескольких местоположений или как ряд связанных точек вдоль пути. Соответственно, в каждом из этих способов используется параметр location с указанием координат широты / долготы для идентификации местоположений или вершин пути. При этом значения широты и долготы должны соответствовать действительному местоположению на поверхности земли. Широты могут принимать любое значение от -90 до 90, а значения долготы могут принимать любое значение от -180 до 180. Если же указывается недопустимое значение широты или долготы, ваш запрос будет отклонен как неправильный. Также одним из доступных параметров запроса является формат вывода-outputFormat данных, который может принимать значение либо json (рекомендуемый производителем формат), указывает ответ на запрос в JavaScript Object Notation (JSON); либо -xml формат ,который производит вывод в XML, заключенный в узел <ElevationResponse>. Ещё одним обязательным параметром запроса является ключ API приложения, получаемый разработчиком при создании проекта на Google Console и вводящий некоторые ограничения на количество запросов в день, поступающих от приложения. Этот ключ позволяет пользоваться разработками Google в области карт и других технологий. Как и во всех URL-адресах, параметры разделяются с помощью символа амперсанда (&).Также важно помнить, что URL-адреса должны быть правильно закодированы,

чтобы быть действительными, и ограничены 8192 символами для всех веб-служб. Для каждого действительного запроса служба Elevation возвращает ответ в формате, указанном в URL-адресе запроса. В Приложении 1 приводится пример получения данных о высоте для моего домашнего адреса, а также ответ Elevation API в формате JSON (см. Приложение 1).

2.3 Получение данных о физической активности пользователя

Как упоминалось ранее, благодаря техническому прогрессу в развитии мобильных устройств у разработчиков появилась возможность реализации алгоритмов, позволяющих получать данные о физической активности пользователя. Общим подходом к получению информации о физической активности пользователя при помощи мобильного устройства является сбор и обработка данных со встроенных датчиков, таких как акселерометр, барометр, компас (геомагнитное поле), гироскоп, вспышка, датчик близости, присутствующих на большинстве стандартных устройств. Однако необходимо выделить основные моменты, которым должен удовлетворять данный алгоритм для его надёжной работы с мобильными устройствами. Во-первых, для эффективной реализации данного подхода необходимо, чтобы обращение к сенсорам и расчёты производились без вмешательства главного процессора, что позволяет сохранять низкий уровень энергопотребления. Снижение влияния на заряд батареи также может осуществляться при дозированной передаче данных с сенсоров центральному процессору, а не в виде устойчивого потока, что позволяет снизить расход ресурсов. Кроме того следует учитывать различные технические характеристики устройств, вследствие чего алгоритм должен быть устойчивым к отсутствию тех или иных датчиков.

Долгое время разработка подобных алгоритмов обнаружения изменений в состоянии активности пользователей была основной проблемой разработчиков контекстно-зависимых приложений [21], которые анализируют состояние пользователя и окружающую среду и адаптируют

свою работу под изменяющиеся условия. При реализации собственной логики большая часть времени разработчиков тратится на реализацию алгоритма классификации, удовлетворяющего указанным выше требованиям. Однако в большинстве случаев такие решения обладают большой задержкой, теряют в точности или являются узконаправленными, так как трудно приспособить свою разработку к постоянно меняющейся среде и активности пользователя и данные с датчиков обычно сильно зашумлены . Помимо того что при этом подходе затрачивалось время на разработку, эти решения также приводили к снижению ресурса батареи для пользователей, так как при реализации собственной логики неизбежны частые проверки изменения в пользовательской активности. Для решения этой проблемы в 2018 году компания Google предоставила для разработчиков новый Activity Recognition Transition API(или Transition API для краткости). Данный API позволяет автоматически обнаруживать изменения в действиях пользователя, периодически считывая короткие пакеты данных с датчиков и обрабатывая их с использованием моделей машинного обучения, что является долгожданной функциональностью для многих разработчиков. То есть с помощью Transition API появляется возможность использовать те же наборы обучающих данных и алгоритмическую фильтрацию, которые используются Google, для уверенного и достаточно точного обнаружения начала и конца определенного пользовательского действия, такого как вождение, ходьба или бег. Также плюсом использования данного API по сравнению с написанием собственных алгоритмов является значительное снижение задержки обнаружения активности, так как процесс обработки сигналов происходит быстрее. Согласно бета-тестированию данной функциональности, проведённому компаниями Life360 Engineering и HyperTrack [22] , Activity Transition API предоставляет достаточную точность, не содержит ложных срабатываний и энергопотребление происходит эффективнее в сравнении с комбинированием и использованием других технологий, не жертвуя при этом задержкой и точностью результатов.

Таким образом, в данной работе на основе проведённого анализа подходов и доступных технологий, используются следующие разработки Google: для получения информации о местоположении пользователя Fused Location Provider API, значения высот высчитываются при помощи Elevation API и происходит распознавание активностей на основе Transition API.

Глава 3. Разработка мобильного приложения для мониторинга физической активности человека на платформе Android

В данной дипломной работе был сделан выбор в пользу разработки мобильного приложения на базе ОС Android. Основными факторами, повлиявшими на данное решение, явились лидирующие позиции данной системы на рынке мобильных устройств и возможность охвата наибольшего числа пользователей. Также важным критерием явилась высокая доступность средств разработки и открытость исходного кода данной системы, написанного с использованием основного языка разработки Java. В качестве среды разработки была использована Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, предлагающая встроенный SDK (Software Development Kit – комплект разработки программного обеспечения) и предоставляющая возможность использования Android AVD Manager (AVD = Android Virtual Device) для создания эмулятора устройства, на котором возможно осуществлять проверки работоспособности приложения. На основании данных процентного распределения версий операционной системы Android среди устройств [23], для охвата большего числа пользователей в качестве минимальной версии операционной системы для разработанного приложения был выбран API Level: 21,Lolipop.

3.1 Алгоритм работы основного функционала приложения

Для реализации функции расчёта расхода калорий на основании совершаемой пользователем физической активности в данной дипломной работе был разработан следующий алгоритм, в котором производится учёт влияния скорости и угла наклона на интенсивность совершаемой активности.
Шаг 1: В начальный момент времени создаётся получатель, который следит за возникновением изменения в пользовательской активности. Как только устройство передаёт сигнал о начале какой-либо активности, такой как, например, бег, ходьба или езда на велосипеде, основные данные об этой активности в начальной точке времени фиксируются. То есть производится запоминание контрольных значений, таких как тип данной активности, текущие координаты пользователя в данной точке и высота над уровнем моря.
Шаг 2: После чего вновь устанавливается наблюдение получателя за окончанием данного вида активности и началом нового. Однако в силу того что подсчёт дистанции, соответствующей каждой активности, производится на основе значения координат, недостаточно запоминания контрольных значений только в начальной и конечных точках активности. Примером является бег по кругу с окончанием в совпадающей по координате точке. В этом случае на основании расчёта с использованием координат начальной и конечных точек данной активности дистанция будет обнулена, хотя фактически пользователь совершил множественные перемещения. Также при использовании контрольных значений только в начальной и конечной точке могут возникать ошибки неправильного учёта перепада высот. Например, пользователь поднялся и спустился с горы, затратив большое количество энергии, но в конечной формуле при оценке интенсивности пользовательской активности данный факт не будет отображён в силу того, что разница высот в начальной и конечной точке составляет малое значение. Этот случай продемонстрирован на графике 1.

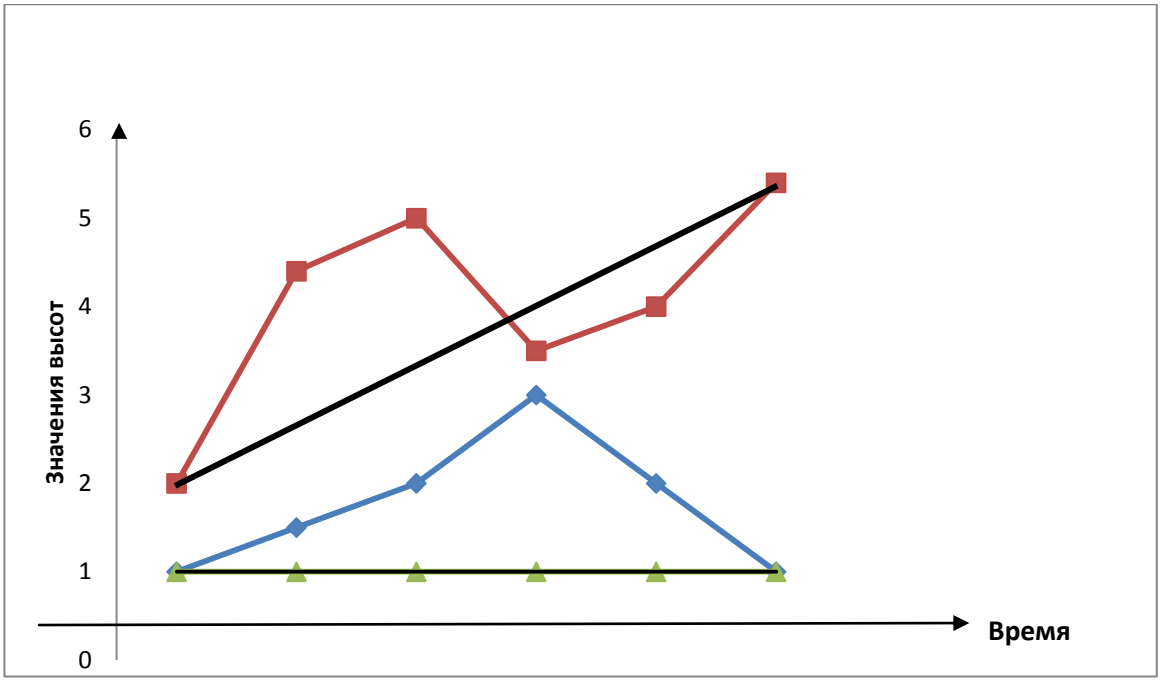

График 1. Демонстрация ошибочной интерпретации пользовательской активности

Для решения данной проблемы в разработанном алгоритме на протяжении пользовательской активности с определённой периодичностью производятся запросы получения обновлённых контрольных значений координат пользователя, времени, а также значений высот над уровнем моря. Таким образом, происходит фиксация значений в контрольных точках, которые можно наблюдать на графике 1. Этот процесс продолжается до тех пор, пока получатель не получает сигнал о начале новой активности.

Шаг 3: После чего между каждой парой точек, которые отсортированы во времени, то есть для каждого отрезка кусочно-линейной функции, отображённой на графике 1, производится вычисление результирующих значений. Данное вычисление происходит в следующей последовательности:

1.Итак, в первую очередь на основе координат широты и долготы каждой из точек производится расчёт плоской дистанции (без учёта перепада высот) между данными точками, на основании метода, приведённого в главе 1.

2.Затем осуществляется вычисление угла наклона на основе перепада высот, вычисленного при помощи вычитания значений высот в каждой из

контрольных точек, и полученной раннее (в пункте 1) значения плоской дистанции.

3.После чего производится вычисление полной дистанции на основе полученных значений угла и плоской дистанции.

4.Также производится расчёт длительности активности между контрольными точками с помощью вычитания значений времени в этих двух точках.

5.Наконец, при помощи деления значения полной дистанции на длительность находится значение скорости.

6.Таким образом, на основе полученных значений скорости и угла наклона производится расчёт интенсивности данной активности в промежутке между контрольными точками. И в итоге при помощи формулы (4) из главы 1 производится конечный расчёт калорий для активности, ограниченной контрольными точками момента времени.

Шаг 4: Далее происходит итеративное повторение шага 3 для каждой пары последовательных контрольных точек до момента достижения конечной точки. При возникновении ситуации временной близости предпоследнего значения по отношению к конечной точке, это значение отбрасывается, так как не оказывает влияния на конечный результат.

Шаг 5: После чего в конечном шаге алгоритма значения, полученные для каждой пары точек, суммируются, и таким образом происходит полный расчёт калорий при выполнении данного вида активности.

Шаг 6: Для расчёта расхода калорий за определённый период времени при совершении различных видов активности значения расхода калорий для каждой из этих активностей суммируются, и вычисляется итоговый расход калорий пользователя за данный период времени.

3.2 Использование программных библиотек

3.2.1 Fused Location Provider

Первым шагом для получения доступа к FusedLocationProviderClient в файле сборки проекта были подключены службы Google Play Services. Далее каждое приложение, использующее службы определения местоположения, должно запрашивать разрешение для использования системных сервисов у пользователя. Android предлагает два разрешения на местоположение: ACCESS_COARSE_LOCATION и ACCESS_FINE_LOCATION, которые прописываются в качестве uses-permission в файле манифеста проекта. Проверка наличия данных разрешений проводится один раз при установке приложения на устройство, основной код реализован в методе requestPermission(). Итак, третьим шагом для использования Fused Location Provider API необходимо создать экземпляр FusedLocationProviderClient для обращения к системным сервисам, в качестве параметра при создании для него устанавливается высокая точность HIGH_ACCURACY.

Для получения последнего известного местоположения пользователя используется переопределение системного метода getLastLocation(), который возвращает конечный результат в виде объекта Location. Для получения необходимых данных долготы и широты, в свою очередь, используются соответствующие статистические методы класса Location, getLatitude() и getLongitude(). Также данный API предоставляет возможность запрашивать периодические обновления на основе экземпляра FusedLocationProviderClient. Данная функциональность реализуется с помощью предоставляемого метода requestLocationUpdates(). Причём параметры точности и частоты обновлений устанавливаются при помощи методов setInterval() и setPriority() при создании экземпляра LocationRequest.

Таким образом, происходит получение необходимых данных долготы и широты местоположения пользователя, которые используются в дальнейшем согласно алгоритму, описанному ранее.

3.2.2 Elevation API

Итак, для использования Elevation API для получения данных о высоте над уровнем моря в данной работе была создана учетная запись на Google Cloud

Platform (которая предоставляет доступ к Google Maps Platform). Далее в данной учётной записи был создан проект [24], для которого было активировано использование Elevation API и получен специальный ключ разработчика API_KEY, упомянутый ранее в главе 2. Хранение данного ключа осуществляется в качестве мета данных в файле манифесте проекта, таким образом, что ключ защищён от злонамеренного использования со стороны. Благодаря данному ключу у разработчика имеется возможность просматривать информацию по удалённому проекту на Google Cloud Platform, которая содержит статистику по количеству запросов, их задержке, количеству ошибок и так далее. Итак, в силу создания специального ключа разработчика и получения доступа к веб-сервису Elevation API становится возможным осуществление запроса, формирование которого происходит по правилам, описанным в главе 2. Основным функционалом для получения информации о высоте пользователя над уровнем моря в данном приложении является метод getElevation(). Параметрами данного метода являются значения широты и долготы, а результатом работы значение высоты над уровнем моря (elevation), соответствующее данным координатам. В начале происходит формулировка запроса, по правилам описанным в главе 2, в запрос передаются параметры метода, значение ключа и запрашивается формат вывода в виде JSON. Далее создаётся асинхронная задача ProgressTask при помощи наследования класса AsyncTask, для перемещения трудоёмких операций в фоновый поток. В данной задаче строка запроса в методе getContent() преобразуется в объект URL, и осуществляется создание HTTP соединения. Далее уточняется тип запроса GET, накладывается максимальное время ожидания, после чего при помощи буферизированного потока ввода (Buffered InputStreamReader) происходит чтение ответа строку. При помощи метода onPostExecute()прочитанная строка возвращается в метод getElevation() , где на её основе создаётся JSON объект, и после осуществления парсинга данного объекта, происходит выделение конечного

значения высоты над уровнем моря (elevation) из всех данных ответа на запрос.

3.2.3 Activity Recognition Transition API

Для использования Transition API в приложении необходимо объявить зависимость от API Location and Activity Recognition Google версии 12.0.0 или выше(gms:play-services-location:12.0.0) в файле build.gradle сборки, а также указать разрешение user-permission ACTIVITY_RECOGNITION в файле манифесте приложения. После чего, чтобы начать получать уведомления об изменениях активностей от устройства, необходимо реализовать объект ActivityTransitionRequest, который указывает тип детектируемой активности и перехода(ENTER или EXIT), а также реализовать обратный вызов PendingIntent на основе Intent(TRANSITION_ACTION_RECEIVER), куда приложение получает уведомления от системных сервисов о полученном сигнале. Далее для создания объекта ActivityTransitionRequest вызывается метод setUpTransitions(), в котором инициализируется список объектов ActivityTransition, которые содержат ограничения на тип перехода и активности, о которой требуется получать уведомления. В данном проекте такими активностями являются бег, ходьба, состояние покоя и езда на велосипеде (RUNNING,WALKING,STILL,ON_BICYCLE). Для более удобного представления типа перехода и активности в виде строк реализованы методы toTransitionType() и toActivityString. Итак, необходимо зарегистрироваться для получения обновлений об изменении в активности, передав созданный экземпляр ActivityTransitionRequest и свой объект PendingIntent методу requestActivityTransitionUpdates (). Далее после успешной регистрации обновлений об изменении активности приложение получает уведомления в зарегистрированное намерение PendingIntent. Далее важнейшим компонентом является обработка данного события при помощи реализации собственного обратного вызова, создав подкласс BroadcastReceiver myTransitionReceiver и переопределив в нём метод

onReceive ().На входе в данном методе производится проверка соответствия полученного Intent объекта нужному типу ACTION_RECEIVER, и далее проводится проверка при помощи метода hasResult() на наличие результата. После прохождения проверки на основе объекта intent получается результат ActivityTransitionResult, затем из него извлекается список событий events при использовании метода getTransitionEvents(), срабатывающий при изменении активности. Далее производится обработка данных событий с помощью методов getActivityType() и getTransitionType(), и таким образом получается информация о типе активности пользователя и типе перехода. Соответственно после чего эти значения заносятся в базу данных, при помощи классов и методов, описанных ниже, в таблицу activities, также описанную ниже. Затем для реализации шага 2 происходит выделение нового фонового потока, в котором реализуются периодические запросы об обновлении данных местоположения и значения высот в ключевых временных точках, результат работы которых также заносится в базу данных в таблицу valuesTemp. После получения Intent от системных сервисов о конце данного вида активности и начале нового, производится закрытие выше указанного фонового потока. И на основании данных, занесенных в базу, производится последовательная обработка значений таблицы valuesTemp и расчёт калорий согласно последовательности шага 3. После чего результат расчётов и длительность в целом для данной активности также заносится в базу данных, в таблицу activities. По завершении данной операции начинается обработка новой полученной активности по такому же принципу. Также для сохранения заряда устройства необходимо завершать работу с Transition API, то есть отменить регистрацию для обновлений при изменении активности, вызвав метод removeActivityTransitionUpdates () и передав объект PendingIntent в качестве параметра.

3.3 Хранение данных

В связи с необходимостью хранения достаточно большого объёма повторяющихся и структурированных данных, а также их доступности и

уникальности для каждого пользователя устройства и отсутствия необходимости обмена, в данном дипломном проекте было решено осуществлять хранение при помощи локальной базы данных на устройстве. Итак, хранение данных осуществляется в двух основных таблицах, таких как valuesTemp и activities. Структура базы данных может быть найдена в приложении 2.

Для реализации данной задачи в Android по умолчанию используется встроенная в систему легковесная реляционная СУБД SQLite. Однако в силу низкоуровневости данной разработки и на основании рекомендаций официального сайта Android для разработчиков в данной работе было решено использовать библиотеку Room, представляющую собой уровень абстракции над SQLite. Для осуществления CRUD операций обращения к базе данных в данной библиотеке используются аннотации, то есть все команды DML (Data Manipulation Language) теперь аннотированы, кроме команды SELECT, которая работает с @Query.

Room состоит из трёх основных компонентов:[25] Entity, который представляет таблицу в базе данных, Database в котором происходит создание singletone-объекта базы данных, а также собственно DAO(Data Access Objects), который действует как посредник между пользователем и базой данных. Схема взаимодействия компонентов продемонстрирована на рисунке 5.

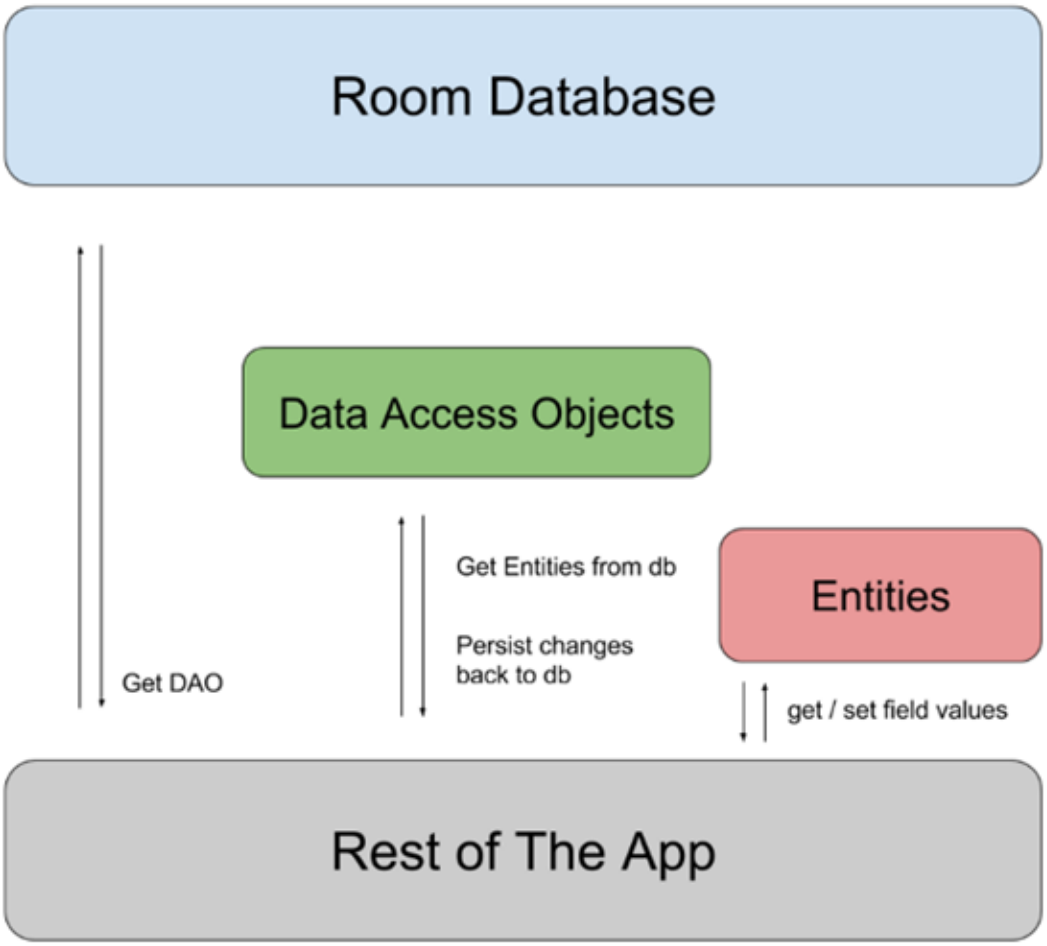

Рисунок 5. Взаимосвязь между компонентами Room

Соответственно компонент Entities реализуется при помощи двух классов ValuesTemp и DetectedActivities, в каждом из которых происходит взаимодействие с соответствующими таблицами valuesTemp и activities. Указанные классы содержат различные методы get, set для наполнения таблиц, а также с помощью полей классов определяются имена столбцов в таблицах и их типы данных. Важно отметить, что SQLite не имеет класса хранения, выделенного для хранения дат и / или времени. Вместо этого встроенные функции даты и времени SQLite могут хранить даты и время в виде значений типов данных TEXT(аналог String в Java), INTEGER(аналог long в Java) и REAL(аналог double в Java).[26] Остальные типы следует конвертировать, прежде чем сохранять в базе данных. Соответственно для корректной конвертации поля time, являющегося объектом типа Date и необходимого для отображения текущего времени, был реализован класс

DateConverter, в котором осуществляется двухстороннее преобразование типов Date и Long. Данный класс прописывается при помощи аннотации @TypeConverters в классе ValuesTemp. Следующим важнейшим компонентом являются объекты доступа DAO, которые также реализуются при помощи двух интерфейсов ValuesTempDao и DetectedActivitiesDao. Здесь содержатся методы, используемые для доступа к базе данных, то есть реализуются запросы и CRUD операции. Основной функциональностью этих интерфейсов являются методы вставки и удаления значений из таблиц, реализуемые при помощи аннотаций @Insert и @Delete. Также именно в данных интерфейсах при помощи аннотации @Query создаются запросы для получения значений из базы данных. И последний важнейший компонент реализуется при помощи абстрактного класса AppDatabase, наследника класса RoomDatabase. Главным в данном классе является создание singletone-объекта базы данных при помощи метода getDatabaseInstrance. При этом также осуществляется связь с компонентом сущностей благодаря указанию в аннотации @Database в параметре entities ссылки на классы DetectedActivities и ValuesTemp. Также прописываются абстрактные методы без параметров activityDao() и valueDao(), возвращающие значения соответствующих типов, с помощью данных методов осуществляются запросы и CRUD операции. Таким образом, реализуется взаимосвязь всех трёх основных компонентов Room и решается задача хранения и взаимодействия с данными в разработанном приложении.

3.4 Архитектура приложения

Реализованное приложение имеет стандартную архитектуру и состоит из двух основных уровней, таких как View Layer и Data Layer. Уровень представления (View layer) отвечает за обработку данных и их отображение. Данный уровень состоит из набора активностей(Activities)- компонентов приложения, которые выдает экран, и с которым пользователи могут взаимодействовать для выполнения каких-либо действий. Каждая активность

представляется в виде класса, в котором находятся все основные методы, при помощи которых осуществляются расчёты, указанные в алгоритме 3.1, а также здесь осуществляется реализации графического пользовательского интерфейса. Каждой из активностей соответствует свой xml-файл описания расположения визуализируемых объектов. Основной активностью данного приложения является MainActivity, в которой происходит инициализация базы данных и находятся все основные методы, реализующие алгоритм 3.1 при использовании выше указанных разработок Fused Location Provider, Elevation API и Transition API для получения данных от сервисных служб. Data Layer же состоит из локальной реляционной базы данных SQLite, структуру которой составляют две основные таблицы, описанные в предыдущем пункте. Взаимодействие между данными уровнями осуществляется посредством использования компонентов библиотеки Room, реализация которых представлена в пункте 3.1. Таким образом, общую схему взаимодействия активностей между собой и с базой данных можно наблюдать на рисунке 6.

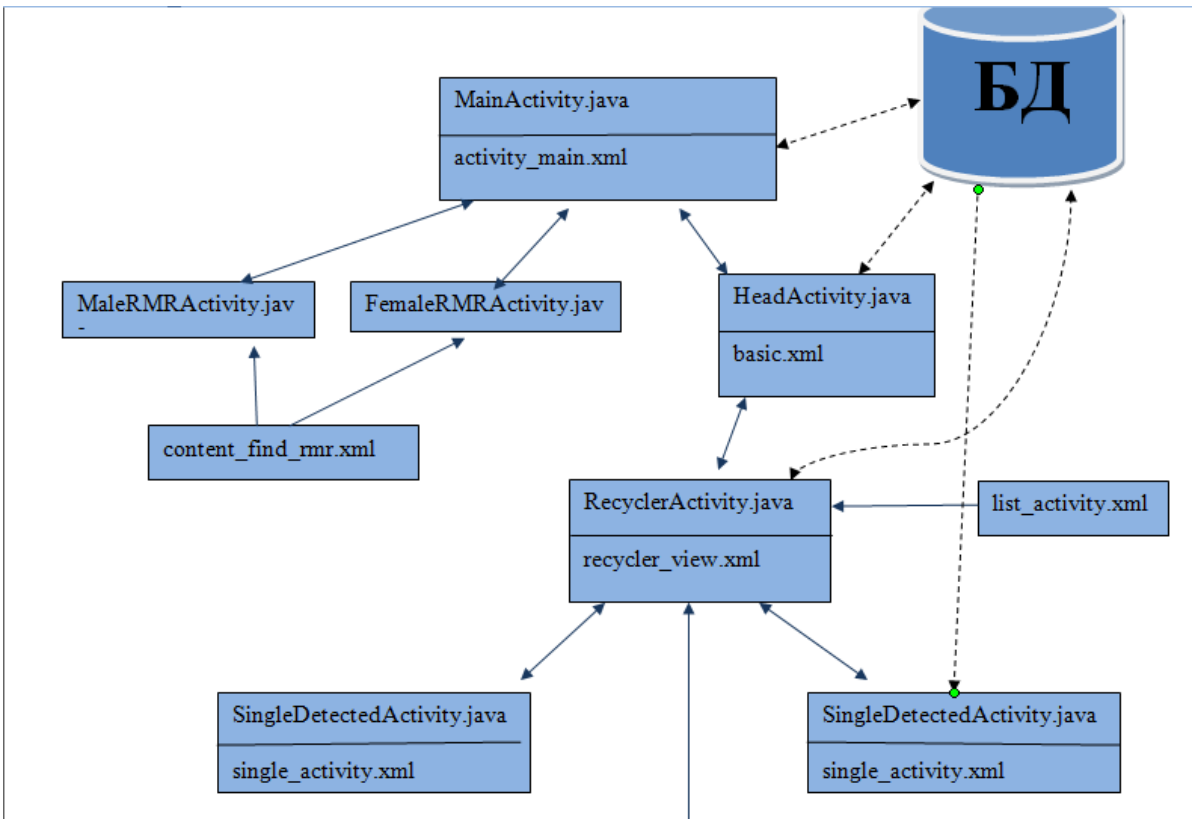

Рисунок 6. Демонстрация взаимодействия основных компонентов приложения

Для организации эффективного взаимодействия данных уровней и снижения нагрузки на главный поток приложения, используется выведение операций взаимодействия с базами данных, сложных расчётов и запросов на получение данных от сервисов в фоновые потоки приложения, которые впоследствии синхронизируются между собой.

3.5 Описание функциональности

При запуске приложения на экране отображается графический интерфейс MainActivity. Он продемонстрирован на рисунке 7, и предоставляет возможность перемещения во связанные активности при помощи использования кнопок, отображённых на экране. При использовании нажатия на кнопку Male или Female получаются данные о поле пользователя. После чего при помощи классов MaleRMRActivity и FemaleRMRActivity, в которых реализован расчёт базового расхода калорий (RMR), на основе обработки введённых пользователем личных данных, что реализуется при помощи компонента EditView и отображено на рисунке 7, производится расчёт базового расхода калорий.

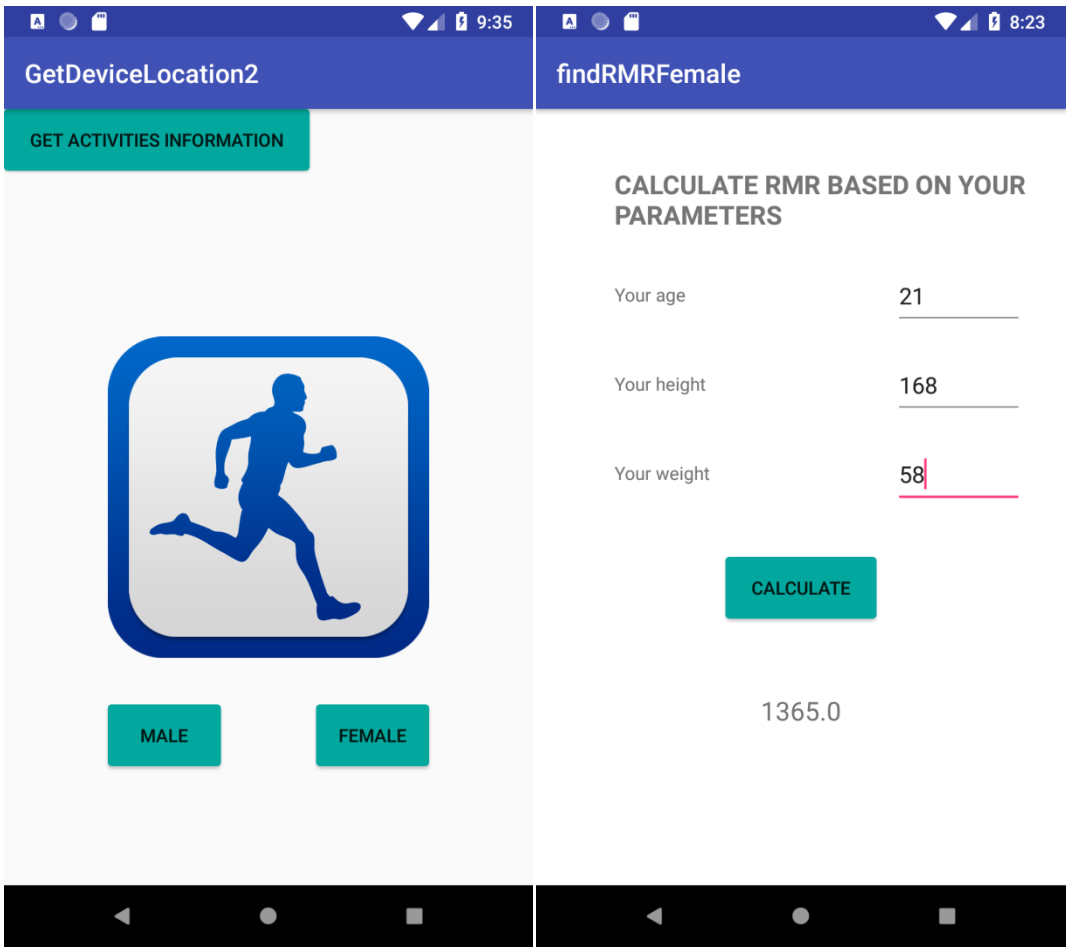

Рисунок 7. Демонстрация некоторых компонентов графического интерфейса приложения

Также MainActivity с помощью кнопки связана с компонентом HeadActivity. Здесь производится вывод информации о расходе калорий за последние 24 часа, а также нескольких других параметров, которые можно наблюдать на рисунке 8. Также реализован компонент RecyclerActivity, представляющей функциональность получения списка активностей за определённый период времени, установленный пользователем, с последующим выводом данного списка на экран. В списке содержится информация о названии активности, а также времени детекции. Графический интерфейс представлен на рисунке 8. Благодаря реализации классов RecyclerTouchListener и MyAdapter была предоставлена возможность выбора определённой активности из списка при нажатии на неё. После чего происходит отображение длительности данной активности и расхода калорий в активности при помощи компонента SingleDetectedActivity.

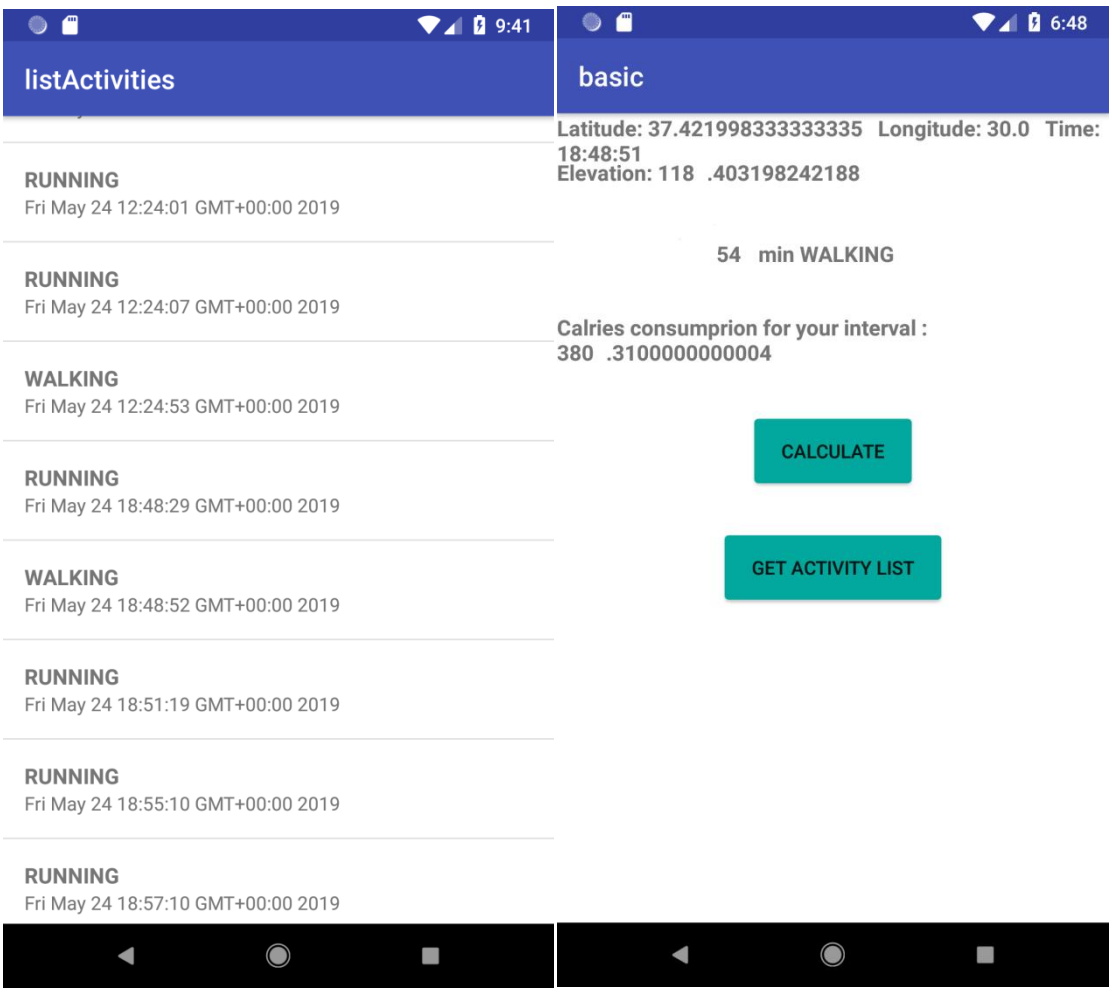

Рисунок 8. Демонстрация некоторых компонентов графического интерфейса приложения

3.6 Тестирование

Работоспособность разработанного приложения была проверена при помощи различных эмуляторов смартфонов, а также на устройствах разных производителей. В ходе данных проверок приложение показало свою состоятельность и стабильность работы вне зависимости от версий операционных систем, а также технических характеристик устройств.

Были проведены практические исследования, в ходе которых испытуемые имели в своём распоряжении несколько устройств с установленными на них различными приложениями, реализующими функцию расхода калорий. В их числе было и приложение, предложенное в данном дипломном проекте. В результате этого исследования было выявлено, что полученные значения расхода калорий коррелируют с

существующими решениями, реализующими похожую функциональность. Однако в силу учёта влияния на расход калорий дополнительных параметров, таких как угол наклона и скорость, проведение полного анализа точности результатов не представляется возможным в силу отсутствия технического оборудования и необходимости поведения практического анализа среди группы людей с известным эталонным значением расхода и потребления калорий .

Также для проверки отказоустойчивости взаимодействия компонентов приложения между собой и с базой данных были проведены юнит-тесты.

Выводы

В результате проделанной работы было создано мобильное приложение на базе ОС Android, осуществляющее расчёт расхода калорий и хранение актуальной информации о физической активности пользователя.

В ходе работы были исследованы различные подходы для оценки расхода калорий с учётом дополнительных параметров скорости и перепада высот. Также были рассмотрены существующие приложения для мониторинга физической активности человека. Исходя из достоинств и недостатков рассмотренных программных решений, были составлены требования для разрабатываемого программного обеспечения. В процессе разработки был проведён обзор различных специализированных технологий, среди которых были выбраны подходящие под составленные требования.

Был разработан и реализован алгоритм, позволяющий получать данные о расходе калорий пользователем за определённый период времени, основываясь на данных о его физической активности с учётом влияния скорости активности и угла наклона местности. Также была реализована возможность получения базового расхода калорий и представлен

графический интерфейс для отображения результирующих данных пользователю.

Результаты, полученные при использовании приложения, коррелируют с данными существующих решений, реализующих похожую функциональность. Также приложение показало свою работоспособность, как на стандартных эмуляторах, взятых из SDK Android, так и на реальных устройствах на платформе Android.

В качестве дальнейшего развития приложения может быть произведено решение следующих задач:

1. Проведение практического исследования на контрольной группе людей для более полной оценки точности получаемых результатов

2. Интеграция с внешними носителями

3. Расширение предоставляемой функциональности

4.Улучшение графического пользовательского интерфейса

Заключение

В результате работы было создано приложение, отличающееся от аналогов внедрением дополнительных параметров учёта скорости и перепада высот при расчёте расхода калорий за определённый период времени, основанный на физической активности данного пользователя в указанный период времени. Эта возможность была реализована за счет применения иных формул расчёта интенсивности физической активности, в которых производится комбинирование значений скорости и угла наклона. Созданное мобильное приложение также предоставляет функциональность, позволяющую пользователю получить информацию о базовом расходе калорий, его текущем местоположении и высоте над уровнем моря.

Список литературы

1. Nisarg Gandhewar, Rahila Sheikh Google Android: An Emerging Software Platform For Mobile Devices // International Journal on Computer Science and Engineering (IJCSE). 2010. NCICT 2010, Special Issue, P. 12-17
2. Statistics Number of mHealth app downloads worldwide from 2013 to 2017 (in billions). https://www.statista.com/statistics/625034/mobile-health-app-downloads/
3. Basal metabolic rate (BMR) https://en.wikipedia.org/wiki/Basal_metabolic_rate
4. Resting metabolic rate (RMR) https://en.wikipedia.org/wiki/Resting_metabolic_rate
5. Mifflin M.D. ,St Jeor S.T., Hill L.A., Scott B.J., Daugherty S.A., Koh Y.O. A new predictive equation for resting energy expenditure in healthy individuals // The American Journal of Clinical Nutrition. 1990. Vol. 51, Issue 2, P.241–247
6. Frankenfield D., Roth-Yousey L., Compher C. Comparison of predictive equations for resting metabolic rate in healthy nonobese and obese adults: a systematic review // Journal of the American Dietetic Association . 2005. Vol. 105, Issue 5,P.775–789.
7. ВОЗ: Интенсивность физической активности https://www.who.int/dietphysicalactivity/physical_activity_intensity/ru/
8. Metabolic equivalent(MET) https://en.wikipedia.org/wiki/Metabolic_equivalent
9. Maximum oxygen consumption (VO_2 max) https://en.wikipedia.org/wiki/VO2_max
10. Key Concepts in Sport and Exercise Sciences / Edited by: David Kirk, Calton Cooke, Anne Flintoff, Jim McKenna. 1 edition SAGE Publications Ltd(UK), 2008. 160 p.

11. Peter Kokkinos, Leonard A. Kaminsky, Ross Arena, Jiajia Zhang, Jonathan Myers New Generalized Equation for Predicting Maximal Oxygen Uptake (from the Fitness Registry and the Importance of Exercise National Database) // The American journal of cardiology. 2017. Vol. 120, Issue 4, P. 688–692
12. Özge Gizem Esenbuğa,Alper Akoğuz1,Emre Çolak, Beril Varol1 ,Bihter Erol1 Comparison of principal geodetic distance calculation methods for automated province assignment in Turkey // 16th International Multidisciplinary Scientific GeoConference SGEM2016
13. Vincenty's formulae https://en.wikipedia.org/wiki/Vincenty%27s_formulae
14. Google Fit Android reviews https://play.google.com/store/apps/details?id=com.google.android.apps.fitness&hl=en&showAllReviews=true
15. Google Fit iOS https://itunes.apple.com/app/id1433864494
16. Mobile phone tracking location https://en.wikipedia.org/wiki/Mobile_phone_tracking
17. Fused Location Provider API https://developers.google.com/location-context/fused-location-provider/
18. DEM data sources http://www.gpsvisualizer.com/elevation
19. Wang Y, Zou Y, Henrickson K, Wang Y, Tang J, Park B-J Google Earth elevation data extraction and accuracy assessment for transportation applications //PLOS ONE. 2017. Vol. 12, Issue 4, P. e0175756
20. Google Elevation API https://developers.google.com/maps/documentation/elevation/intro
21. Контектно-зависимые приложения https://en.wikipedia.org/wiki/Context-aware_pervasive_systems
22. Результаты бета-тестирования Google Transition Activity Recognition API https://medium.com/life360-engineering/beta-testing-googles-new-activity-transition-api-c9c418d4b553/

https://medium.com/hypertrack/hypertrack-supports-the-new-transition-api-for-android-activity-recognition-5a5f22930285

23. Breakdown of Android versions
https://en.wikipedia.org/wiki/Android_(operating_system)

24.Проект на Google Cloud Console
https://console.cloud.google.com/apis/dashboard?project=getdevicelocation-236408

25. Room https://developer.android.com/training/data-storage/room/index.html

26. SQLite data types https://www.sqlite.org/datatype3.html

27. Исходный код приложения
https://github.com/VeronikaRevjakina/GetDeviceLocation

Приложение 1

В следующем примере запрашивается высота для моего домашнего адреса в формате JSON:
https://maps.googleapis.com/maps/api/elevation/json?locations=60.040603,30.3377286&key= YOUR_API_KEY

Валидным ответом на данный запрос является:


```
{
  "results" : [
    {
      "elevation" : 32.72240447998047,
      "location" : {
        "lat" : 60.040603,
        "lng" : 30.3377286
      },
      "resolution" : 15.7032318115234
    }
  ],
  "status" : "OK"
}
```

Приложение 2

Структура базы данных

Таблица 1. valuesTemp промежуточных значений детектированных активностей пользователя

id	Автоматически генерируемый идентификатор
activity_id	Числовой эквивалент DetectedActivity:1-езда на велосипеде, 3-состояние покоя, 7-ходьба,8-бег
transition_type	0-ACTIVITY_TRANSITION_ENTER 1- ACTIVITY_TRANSITION_EXIT
latitude	Значение широты в пределах от -90 до 90
longitude	Значение долготы в пределах от -180 до 180
elevation	Значение высоты над уровнем моря
time	Время полученной активности в формате Date

Таблица 2. activities детектированных активностей пользователя

id	Автоматически генерируемый идентификатор
detected_activity_name	Название детектируемой активности: STILL,WALKING,RUNNING

detected_activity_id	Числовой эквивалент DetectedActivity:1-езда на велосипеде, 3-состояние покоя, 7-ходьба,8-бег
transition_type	0-ACTIVITY_TRANSITION_ENTER 1- ACTIVITY_TRANSITION_EXIT
duration	Значение длительности в минутах
calories_consump	Расход калорий для данной активности