

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерные Технологии

«Допущен к защите»  
Заведующий кафедрой Куралбаев З.К.  
профессор, к.ф.-и.н.  
(Ф.И.О., ученая степень, звание)  
«    »    20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка базы данных для  
мобильного приложения "30ж"

Специальность 5В070400 - ВТ и ПО

Выполнил (а) Абзал Д.А. ВТ-12-2  
(Фамилия и инициалы) группа

Научный руководитель К.Ф.-и.н. доцент Эубакир С.Б.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бексүлей А.С., к.э.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
Б-С « 31 » 05 20 16 г.  
(подпись)

по безопасности жизнедеятельности:

Ахмедов Н.Г. В.х.н., профессор  
(Фамилия и инициалы, ученая степень, звание)  
А-С « 20 » 05 20 16 г.  
(подпись)

по применению вычислительной техники:

Эубакир С.Б., к.ф.-и.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
К-С « 03 » 06 20 \_\_ г.  
(подпись)

Нормоконтролер: К-С Эубакир С.Б. к.ф.-и.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
К-С « 29 » 06 20 \_\_ г.  
(подпись)

Рецензент: \_\_\_\_\_  
(Фамилия и инициалы, ученая степень, звание)  
«    »    20\_\_ г.  
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и Информационных технологий  
Специальность 5В070400 - ВТнПО  
Кафедра Компьютерная технология

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Абзал Диммуханов Абзалов  
(фамилия, имя, отчество)

Тема проекта Разработка базы знаний для мобильного приложения "Золот"

утверждена приказом ректора № 148 от «19» октября 2015 г.  
Срок сдачи законченной работы «01» 06 2016 г.  
Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта мобильное приложение на платформе Android, в ноутбука, количества рабочих мест 2.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

- 1) обзор принципов проектирования
- 2) исследование предметной области
- 3) выбор модели представления знаний
- 4) проектирование базы знаний
- 5) разработка мобильного приложения



Перечень графического материала (с точным указанием обязательных чертежей)

8	рисунков	в	главе 1
7	рисунков	в	главе 2
21	рисунков	в	главе 3
1	рисунков	в	главе 4
1	рисунков	в	главе 5

Рекомендуемая основная литература

- 1) Гаврилова - Базис знаков интеллектуальной системы: Пособие, 2003 - с. 234
- 2) Ахметова М. - системы интеллектуального интеллекта: Конспект лекций 2013-2016
- 3) Android Application development, Rich Rogers, John Lombardo, Zigid Madricks - 3990

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БЖД	Трифедина Н.Г.	18.03-20.05.16	Трифедина Н.Г.
Эконом. часть	Светлачева А.Ч.	20.03-31.05.16	Светлачева А.Ч.
Основная часть	Зубович С.Б.	10.03-10.05.16	Зубович С.Б.
Расчетная часть	Зубович С.Б.	13.04-29.05.16	Зубович С.Б.
Корректировки	Зубович С.Б.	27.05-30.05.16	Зубович С.Б.



## **Аннотация**

В данном дипломном проекте рассмотрена разработка базы знания для мобильного приложения на платформе Android.

Целью дипломного проекта является создание мобильного приложения более актуального чем его аналоги.

Сделан анализ условий труда для сотрудников непосредственно принявших участие в разработке данного приложения.

Также составлено экономическое обоснование проекта, что подтверждает его экономическую целесообразность.

## **Аңдатпа**

Бұл дипломдық жобада Android платформасында жазылған мобильдік қосымша үшін білім базасын дамыту қарастырылды.

Дипломдық жобаның мақсаты, оны әріптестерімен салыстырғанда неғұрлым маңызды ұялы қолданбасын жасау болып табылады.

Қарастылатын қосымшаны жасауға тікелей қатысқан жұмыскерлердің еңбек жағдайларын әзірлеу үшін талдау жасалды.

Сондай-ақ, оның экономикалық орындылығын растайтын, жобаның экономикалық негіздемесі құрастырылды.

## **Annotation**

Development of the knowledge base for mobile application on Android platform has been considered in this thesis project.

The aim of the diploma project is to create a mobile app more relevant than its counterparts.

The analysis of the working conditions made for employees directly participated in the development of the application that is under consideration.

Also the economical justification of the diploma project was composed, that confirms its economic expediency.

## Содержание

Введение.....	11
1 Принципы проектирования.....	13
1.1 База знаний и интеллектуальные системы.....	13
1.2 Технологии и языки разработки базы знаний.....	14
1.3 Исследование предметной области.....	20
1.3.1 Общая классификация методов.....	20
1.3.2 Электронное здравоохранение.....	21
1.3.2 Упражнения и фитнес.....	22
1.3.4 Смартфоны.....	25
2 Классификация моделей представления знаний.....	30
2.1 Формализация знаний в интеллектуальных системах.....	30
2.2 Продукционная модель представления знаний.....	31
2.3 Фреймовая модель представления знаний.....	32
2.4 Семантические сети.....	35
3 Разработка базы знаний.....	38
3.1 Подготовка к работе.....	38
3.2 Проектирование базы знаний.....	40
3.3 Результаты работы приложения.....	48
4 Техничко-экономическая часть.....	57
4.1 Цель проекта и его разработка.....	57
4.2 Трудовые ресурсы, используемые в разработке мобильного приложения.....	58
4.3 Техническое оборудование, использованное при разработке мобильного приложения.....	58
4.4 Программное обеспечение, используемое в ходе разработки.....	58
4.5 Расчет и определение затрат на создание и реализацию проекта.....	59
4.5.1 Расчет затрат по социальному налогу.....	62
4.5.2 Расчет амортизационных отчислений.....	62
4.5.3 Расчет затрат на электроэнергию.....	63
4.5.4 Расчет прочих расходов.....	64
4.5.5 Расчет накладных расходов.....	65
4.5.6 Цена реализации проекта.....	66
5 Безопасность жизнедеятельности.....	68
5.1 Анализ условий труда.....	68
5.3 Расчет освещения.....	74
Заключение.....	77
Список используемой литературы.....	78
Приложение А.....	79

## **Введение**

Вопрос о здоровом образе жизни является широко распространенной темой по всей стране, так как расходы на здравоохранение продолжают расти. Сфера здравоохранения охватывает многие усилия ученых и людей, работающих в области образования и укрепления здоровья, чтобы помочь людям жить более здоровой жизнью. Так как технологии продолжают развиваться, программы электронного здравоохранения и мобильного здравоохранения в настоящее время изучаются как творческие звено для этой деятельности.

Люди пользуются услугами фитнес центров и рассчитывают получить здоровое тело, основываясь на консультациях фитнес инструкторов и здоровом питании. Сфера здравоохранения, количество фитнес центров и количество людей, пытающихся придерживаться здорового образа жизни, растет с каждым годом.

Мобильные устройства, которые открывают новые двери для улучшения здоровья и предоставляет широкие возможности для укрепления здоровья и профилактику заболеваний. Не секрет что мы живем в обществе с сидячим образом жизни, когда люди все чаще сталкиваются с серьезными рисками для здоровья и опасными заболеваниями. Поэтому весьма необходимым является проведение научных исследований, чтобы получить дальнейшее понимание и знания о укреплений здоровья с целью получения более успешных результатов в отношении здоровья. Ожирение и избыточный вес являются причиной смерти 2,8 миллиона человек в году во всем мире. К примеру общий бюджет всемирной организации здравоохранения за 2006-2007 составило 3,3 миллиардов долларов США. Это само по себе ставит человеческим жизням и нашей экономике опасный риск. Менее чем за 40 лет, распространенность ожирения в мире увеличилась более чем на 50%, так что два из каждых трех взрослых людей в мире в настоящее время имеет избыточный вес или ожирение. Таковы лишь некоторые из поразительных статистик, которые провоцируют непосредственную потребность развития сферы здравоохранения и образования. Специальные коммуникационные программы здравоохранения имеют потенциал, чтобы быть отличным инструментом в преодолении эпидемии ожирения. Было решено разработать мобильное приложение для здорового образа жизни, учитывая вышесказанные основания.

Цель в этом проекте, это расширить электронное здравоохранение на один шаг вперед и исследовать выгоды от конкретного применения мобильного приложения. Приложения для здравоохранения имеют огромный потенциал стать хорошим инструментом для помощи в соблюдении здорового

образа жизни каждого индивидуума, так как почти 40% Казахстанцев являются владельцами смартфонов.

Это очень важно, только по этой причине продолжать расширять исследование электронного здравоохранения для получения хорошего самочувствия. Считаю нужным построить на этой идее оздоровительный центр с рабочими местами, используя обратную связь и результаты моего исследования, как представление о созданий успешных программ в области здравоохранения в этом типе сообщества. Кроме того, я намеревался увидеть, будут ли эти типы программ, использующие мобильные технологии на самом деле ценными.

Для достижения данной цели необходимо проделать анализ знаний экспертов, которые составляют программу питания и тренировок пользователя. Поэтому была сделана постановка задач исследование методов составления программ тренировок и представления знаний, выбор модели знаний. Имея под рукой мобильное приложение с такой базой знаний пользователь имеет возможность вводить данные о пользователе, данные об активности пользователя и отслеживать свой прогресс.

Разработка базы знания является важной частью для любых мобильных приложениях, где используются логические правила и знания экспертов. База знаний содержит в себе информацию о человеческом опыте и знаниях в определенной предметной области, в данном случае в области здорового образа жизни.

Тщательный обзор литературы проводится для того, чтобы добиться лучшего понимания текущей темы в области исследований мобильных приложениях здоровья и оздоровительных программ. Несколько тем были определены и обсуждаются в этом проекте, которые обеспечивают понимание путей моего конкретного исследования которые могут расширить потенциальные определения наиболее ценных факторов применения мобильного здоровья и как это влияет на поведение в отношении здоровья индивидуума.



# 1 Принципы проектирования

## 1.1 База знаний и интеллектуальные системы

Эволюция информационных систем, связанная с ее интеллектуализацией. Интеллектуальные системы, как один из видов информационных систем, являются наиболее перспективным направлением для решения научных и прикладных задач. Эти системами представляют собой любые формальные и искусственные системы, которые демонстрируют способность к целенаправленному поведению. Таким образом, информационная система может считаться умной, если она имеет следующие перечисленные функции:

- функция представления и обработки знаний, другими словами, интеллектуальная система должна приобрести знания, проверять их на предмет соответствия, связывать полученные знания с существующими знаниями в базе знаний;

- функция рассуждения, система искусственного интеллекта должна основываться на моделях частичных знаний и логического вывода чтобы получать общие знания;

- функция общения, интеллектуальная система должна общаться с человеком на языке, как можно ближе к естественному языку;

В настоящее время различные сферы человеческой деятельности широко используют интеллектуальные системы на основе знания (экспертные системы), используемые, как правило, для поддержки принятия решений в слабоструктурированных предметных областях. Экспертные системы являются инструментом для повышения интеллектуальных способностей человека-эксперта, и может выполнять следующие функции:

- консультанта для непрофессиональных или неопытных пользователей;
- помощника поддержки принятия решений в связи с необходимостью проанализировать различные варианты решений;

- партнера эксперта в смежных областях деятельности;

Любая интеллектуальная система состоит из взаимосвязанных компонентов, каждый из которых имеет свое функциональное назначение. Основными компонентами интеллектуальной системы являются:

- база знаний – представление знаний в структуре модели представления знаний;

- модуль логического вывода – это механизм, который при приеме данных на входе с использованием данного алгоритма логического вывода на основе существующих правил принятия решений дает пользователю результат;



– подготовку подробного плана разработки.

Верный отбор трудности дает, наверняка, самую критическую дробь исследования в целом. Если избрать неуместную дилемму, разрешено совсем скоро вязнуть в "топком месте" проектирования задач, которые никто никак не понимает, как улаживать. Неуместная неувязка имеет возможность еще привести к творению экспертной системы, стоящая гораздо более, нежели хранящая. Ремесло станет обстоять еще ужаснее, если создать систему, которая действует, однако никак не применима для юзеров. В том числе и если исследование производится лично организацией для личных целей, данная фаза считается пригодным фактором для получения советов снаружи, чтоб обеспечивать успешно избранный и легко осуществимый с тех. точки зрения начальный план.

В выборе сферы внедрения надлежит учесть, что если нужное познание для решения задач, неизменное, отчетливо формулируемое, и соединено с вычислительной отделкой, то обыденные алгоритмические компилятор, следуя всей вероятности, станут наиболее подходящим методом решения нестыковок в данной области. Экспертная система ни в коем случае никак не уничтожит надобность в реляционных базах данных, статистическом программном трансляторе, электрических таблицах и системах текстовой отделки. Однако если действенность задания находится в зависимости от познания, которое считается необъективным, изменяющимся, символьным либо вытекающим отчасти из суждений здравого значения, тогда область имеет возможность продуманно ходить кандидатом на экспертную систему.

Найдем факторы, заключающие о надобности создания и использования экспертных систем:

- недостаток знатоков, которые тратят значимую часть времени для предоставления помощи остальным;
- потребность в многочисленной группе экспертов, поскольку ни один из них не располагает идеальным полным знанием;
- уменьшенная производительность труда, поскольку задание просит полного анализа - сложное набора условий, а рядовой эксперт не в состоянии просмотреть (за предоставленное время) все эти условия;
- значимая разница между решениями сильнейших и худших исполнителей;

Подходящие задачи имеют следующие характеристики:

- являются узкоспециализированными;
- не зависят в какой-либо степени от общих человеческих познаний или здравых рассуждений;
- не являются для знатока ни слишком простыми, ни слишком трудными (время, которое потратит эксперт для того, чтобы справиться с проблемой, может составлять от двух часов до трех-четырех недель);
- условия выполнения задания выбираются самим юзером системы;
- имеет результаты, которым можно дать оценку.

Традиционно экспертные системы разрабатываются маршрутом получения специфичных познаний от профессионала и ввода их в систему. Некие системы имеют все шансы владеть стратегиями индивидуума. Следственно, отыскать пригодного профессионала - наиболее главный шаг в разработке экспертных систем. В процессе исследования и следующего расширения системы эксперт по знаниям и разработчик традиционно действуют совместно. Эксперт по знаниям подсобляет профессионалу структурировать познания, предопределять и формализовать мнения и принципы, нужные для решения трудностей. Во время начальных разговоров они решают, станет ли их совместная работа эффективной. Очень важно, так как две стороны станут действовать совместно, согласно наименьшей мерке, в продолжении одного года. Не считая их, в коллектив создателей будет целенаправленно подключить возможных юзеров и профи разработчиков программного обеспечения.

Подготовительный подготовка к программной разработке задания ориентируется на основе параметров задания и ресурсов, доступных на ее заключение. Знатоки по знаниям выдвигают традиционно некоторое количество разновидностей, связанных с внедрением наличествующих в сфере программных средств. Конечный отбор вероятен только на шаге исследования макета. В завершении того, как задание определено, нужно подсчитать затраты и выгоды от исследования экспертной системы. В затраты относятся издержки на плату труда коллектива проектировщиков. В побочные затраты покупаемого программного инвентаря, с поддержкой которого проектируется экспертная система.

Выручку можно получить с помощью понижения цены конечного продукта, увеличения производительности труда, расширения бумажных документов продукта или услуг или даже создание новых типов продукта или услуг в данной области. Надлежащие затраты и выгоды от системы ориентируются опираясь на время, на протяжении которого окупятся финансы, затраченные на исследование. На нынешней ступени крупная часть компаний, развивающих огромные экспертные системы, выбрали создавать дорогие планы, дающие значимые выгоды.

Наметились зависимости исследования наименее дорогих систем, несмотря на более долгий сроком возвращаемости затраченных на них финансов, так как платформенные методы исследования экспертных систем постоянно улучшаются. Впоследствии, эксперт по знаниям удостоверился, что предоставленное задание имеет возможность быть решенной с поддержкой экспертной системы:

- экспертную систему можно спроектировать предоставленными на рынке инструментами;
- доступен необходимый эксперт;
- предложенные параметры разработки не являются невозможными;

– расходы и время их возвращаемости согласованы с заказчиком.

Этапы разработки экспертных систем. На рисунке 1.1 показана схема последовательности этапов разработки и проектирования экспертных систем.

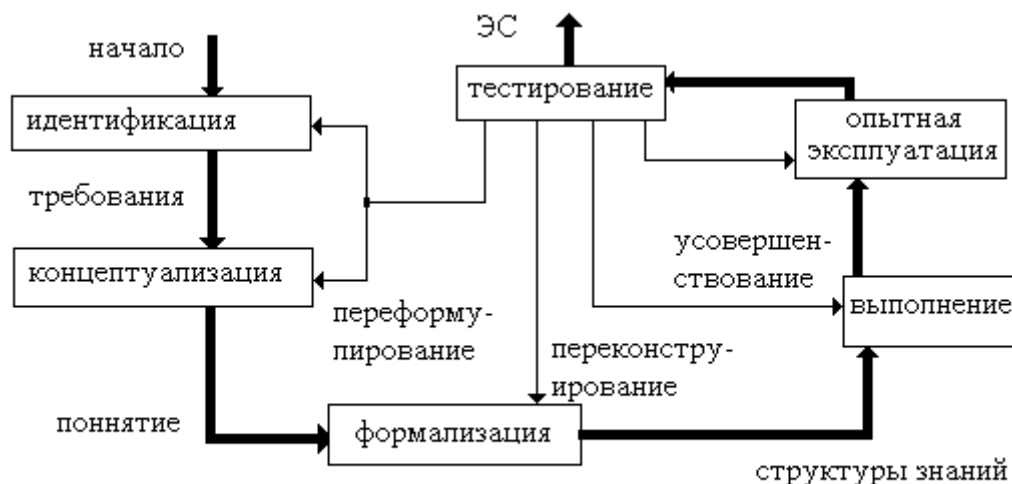


Рисунок 1.2 – Этапы разработки

При проектировании экспертных систем нередко употребляется теория скорого макета. Сущность ее в последующем: сначала формируется не сама экспертная система, а ее макет, обязанный исправлять узкий набор проблем и запрашивать на собственное исследование небольшие сроки. Макет обязан показывать целесообразность грядущей экспертной системы для предоставленной предметной области, испытать верность шифровки прецедентов, взаимосвязей и стратегий размышления профессионала. Помимо этого, он предоставляет шанс знатоку по знаниям пригласить профессионала к функциональной роли в исследовании экспертной системы. Величина макета – несколько десятков принципов.

На сегодняшний день появилась закономерная тенденция создания экспертных систем, содержащая 6 этапов.

Этап 1. Идентификация.

Рассматриваются задачи, которые доступны для решения. Составляется план процесса создания модели экспертной системы, ведется расчет нужных ресурсов (сроки, разработчики, вычислительные машины и т.д.), источники знаний (книжные данные, дополнительные эксперты, методики), существующие аналогичные экспертные системы, цели (передача опыта, механизация повседневных движений и др.), классы поставленных задач и т.д. Этап идентификации – это приветствие и обучение группы проектировщиков. Средняя продолжительность такого этапа обычно составляет 1-2 недели. На этом же этапе проектировки экспертных систем проводится выявление знаний. Знаток по знаниям помогает профессионалу получить и упорядочить



знания, необходимые для существования экспертной системы, с употреблением всевозможных способов: анализ статей, разговоров, логические игры, лекционный материал, дискусирующий монолог, наблюдение и другие. Выявление знаний – это получение знатоком по знаниям более углубленного мнения о предметной области и способах выбора решения в ней.

#### Этап 2. Концептуализация.

Определяется модель выявленных знаний о предметной области. Выясняется: терминология, список главных обозначений и их параметров, вид входных и выходных данных, тактика выбора решений и т.д. Концептуализация – это исследование неофициального изображения познаний о настоящем участке в облике графа, таблицы, диаграммы или слова, что отображает основные пункты и связи между мнениями предметной области. Продолжительность этапа 2-4 недели.

#### Этап 3. Формализация.

В этапе формализации все главные мнения и связи, обнаруженные в прошлом этапе, отображаются в некоем специфичной речи, порекомендованной (избранной) знатоком по знаниям. Далее он принимает решение, то что нужна единица наличествующего приборного имущества соответствует имеющейся трудности либо нужен отбор иного инвентаря, либо потребуются уникальные исследования. Продолжительность этапа 1-2 месяца.

#### Этап 4. Реализация.

Проектируется модель экспертной системы, содержащей базу знаний и другие подпункты. На данном этапе используются следующие инструментальные технологии: программирование на объектно-ориентированных языках (Паскаль, Си и др.), программирование на специфических языках, используемых в задачах искусственного интеллекта (LISP, FRL, SmallTalk и др.) и др. Четвертый этап создания в некотором смысле может называться основным, потому что здесь протекает возникновение программной концепции, показывающей жизнеспособность способа в целом. Продолжительность этапа 1-2 месяца.

#### Этап 5. Тестирование.

Модель проходит проверку на практичность и адекватность аналогов ввода-вывода, эффективность тактики управления, количество проверочных заданий, грамотность базы знаний. Тестирование – это нахождение неточностей в данной методике, поиск багов в реализации модели, а также создание советов по приведению продукта до промышленного уровня.

#### Этап 6. Опытная эксплуатация.

Проверяется работа экспертной системы для конечных юзеров. По итогам шестого этапа возможно будет необходимо значимое обновление экспертной системы.

Ход создания экспертной системы не заключается в строгий порядок перечисленных этапов. В процессе труда необходимо регулярно обращаться к ранним этапам и анализировать принятые до этого решения.

Языки разработки экспертных систем. Для разработки экспертных систем подходят те же языки и методы программирования, как и для рядовых программ, но существование таких необычных для искусственного интеллекта составных частей, как логический вывод, естественно языковой интерфейс, располагает к использованию для разработки экспертных систем таких языков, как Пролог, Лисп, Клипс и особенных ресурсов поддержки создания.

Самой удобной для экспертных систем стала возникновение языка Пролог. Ключевой смысл логического программирования заключается в разделении логики платформы от контроля процессом слежения, что делает ход создания утилиты более легким и понятным. Пролог – язык высокого уровня, направленный на задействование моделей и методов математических законов. Разработан во Франции в Марсельском университете в 1972 году. Ключевым преимуществом Пролога, возвышающего его над остальными языками, является показная форма содержащихся в нем предикатов. Он специализирован для создания систем и подпрограмм искусственного интеллекта; классифицируется к классу языков пятого этапа. При работе с ним писателям кода не нужно расписывать пошагово процедуры — достаточно лишь выявить факты и установить их взаимоотношения. С помощью этих взаимоотношений процедуры, доступные в языке, располагают логическими выводами. Эта фишка делает Пролог наиболее адаптированным к созданию экспертных систем.

Язык Лисп изобретен в Массачусетском научно-техническом вузе в истоке 60-х лет. Языки разработки Лисп и Пролог обладают интегрированными интерпретаторами для манипулирования знаниями. Лисп считается все пригодным языком программирования высочайшего значения и владеет возможностью производить списковые текстуры. Он классифицируется к декларативным языкам многофункционального вида и специализирован на отделеках символьных данных, презентованных в облике списков.

Клипс был создан в лаборатории космических изучений NASA в середине 80х годов. Клипс (Clips) является сокращением от C Language Integrated Production System. Он интегрирует в язык точку зрения образующихся прецедентов и язык охарактеризования процедур. Клипс пользуется продукционной моделью представления знаний и поэтому включает в себя три главных пункта:

- список фактов;
- базу знаний;
- блок вывода.

Базисным различием предоставленного порядка от аналогов считается то, что он целиком создан на языке С. При этом начальный код обнародован в Веб-паутине. В Клипс употребляется уникальный LIPS-схожий язык программирования, направленный в исследование экспертной системы. Не считая этого, Клипс сопоставляется с еще двумя направлениями программирования: объектно-направленное и процедурное.

Помимо Лиспа, Пролога и Клипса существует множество различных языков [4], нацеленных на переработку символьной информации и создание экспертных систем: Smalltalk, FRL, Interlisp. Кроме этих специфичных языков для проектирования экспертных систем могут эксплуатироваться и стандартные языки программирования общего плана: Си, Ассемблер, Паскаль, Фортран, Бейсик и др.

Главным недостатком языков программирования для создания экспертных систем это: значительные затраты времени создания полноценной системы, необходимость задействования высококвалифицированных инженеров-кодеров, неувязки с обновлением итогового продукта. Все это усложняет применение языков программирования для создания экспертных систем и делает весь процесс весьма финансово затратным и трудновыполнимым.

### **1.3 Исследование предметной области**

#### **1.3.1 Общая классификация методов**

У людей существует определенный спрос и потребность в фитнес мотивации. С большим ростом ожирения и недостаточной осведомленности в физических понятиях упражнений существует определенная потребность в последовательной тренировке. Так как много людей используют смартфоны и приложения можно быть уверенным, что в будущем будет только увеличение количества членства спортзалов, если оба были связаны между собой. К примеру, гаджет который позволяет такого рода комбинации, это Fitbit. Fitbit представляет собой устройство, которое отслеживает шаги, расстояние, количество израсходованных калорий и количество ступенек по лестнице по которым поднимались в течение дня. В ночное время устройство измеряет качество сна пользователя и работает как будильник утром. Данные в устройстве отслеживаются и обновляются по беспроводной сети, и пользователь может поделиться своими результатами с друзьями в их социальной сети, пользователь мог получать больше удовольствия от тренировок. Устройство и приложение действуют как преобразователи образа жизни. Если бы такого рода устройства были бы связаны с тренажерным залом, люди, которые уже не занимаются были бы мотивировано еще больше. Полезной функцией для физических упражнений и применения гимнастики является GPS. Она позволяет тренерам в тренажерном зале убедиться в том,

что члены фитнес центров активны за пределами тренажерного зала. Существует, безусловно, потенциал и высокий спрос на приложения для смартфонов, который адаптированной к определенному тренажерного зала и его объектов.

Согласно исследованию, проведенному по использованию магистрантов «мобильных приложений здоровья», около 17% участников этого исследования сообщили об использовании приложения в области здравоохранения. После расчета критерий согласия теста было обнаружено, что гонка была самым важным показателем между полом и классом положения. Основными причинами они заявили, что не имея приложения здоровья было отсутствие интереса, понимание того, что приложения не были эффективными или вообще не подозревая об их существовании. Это явный показатель того, что новая среда и достижения должны быть созданы, чтобы выйти и обеспечить этими приложениями наше поколение.

### **1.3.2 Электронное здравоохранение**

Область электронного здравоохранения, считается имеет большие перспективы для улучшения связи через ориентированного на пользователя дизайн и интерактивность, широкие социальные связи и углубить использование мультимедийных средств массовой информации, которые расширяют доступ населения к информации о состоянии здоровья. Коммуникация здоровья действительно долгий путь, и через развитие технологии и интернет, теперь мы можем уточнить и сосредоточиться подходы к созданию более здорового образа жизни в нашем обществе. Электронное здравоохранение является термином, который в последнее время часто используется в данной сфере. В различных дисциплинах, его определяют его как "широкий термин для гетерогенного и развивающих цифровые ресурсы, которые поддерживают здоровье и здравоохранение. Электронное здравоохранение предлагает так много возможностей, люди в буквальном смысле могут иметь информацию о состоянии здоровья у них под рукой, легкий доступ и удобство в пользовании. Есть много преимуществ для использования и построения программы, основанной на мобильных приложениях. Одним из основных отличий электронного здравоохранения и традиционного медицинского образования является стоимость. Электронное здравоохранение обеспечивает более доступный путь для достижения большего числа людей путем использования технологии. Электронное здравоохранение позволяет человеку следить за их здоровьем больше, чем когда-либо прежде. Оценка состояния здоровья является основной целью приложения. Программы продвижения могут адаптировать сообщения, основанные на конкретных рисках для здоровья человека и предлагают рекомендации, такие как присоединение фитнес-программы, используя управление стрессом, хорошие методы диеты и т.д. В целом, эта тема

приобрела большую популярность в последние годы и во многих отношениях, задача для исследователей и практиков является развитие понимания того, как мы можем наилучшим образом улучшить электронное здравоохранение, сильные стороны приложений для эффективного укрепления здоровья и профилактики заболеваний. Это важно разработать четкие представления о каких областях применения может быть полезно иметь наибольшее влияние на улучшение здоровья.

Мобильные технологии и здоровый образ жизни Я был воодушевлен перспективой использования относительно смартфон приложений, в данном случае в связи с повышением физической культуры среди казахстанцев. Учитывая, что одна треть взрослых казахстанцев физически неактивны. Эффективное приложение «здоровый образ жизни» имеет потенциал оказывать существенное влияние на общественное здравоохранения за счет снижения уровня заболеваемости, связанных с бездействием, например, сердечно-сосудистые заболевания.

Несколько исследований были посвящены изучению индивидуальных потребностей и программы мобильного телефона на основе мотивации и поощрения здорового изменения сидячего образа жизни и с избыточной массой тела или ожирением взрослых. Было шесть фокус-групп в исследовании, для того чтобы узнать взгляды участников на примеры программы изменения образа жизни через мобильные телефоны.

Основные темы в анализе, это социальная и профессиональная поддержка, подгонка времени, частоты и содержание сообщений в мобильном телефоне, сочетание мотивации, самоконтроля, постановки целей, потенциальные барьеры и устойчивости программ для мобильных телефонов. Как ни странно, ни один из участников не обратился в поддержки через группы в социальных сетях. В заключение, акцент делается здесь снова на "покрой" мобильной программы для своей рутины и образ жизни, и что он не должен диктовать свою повседневную деятельность.

Было проведено и другое значительное исследование, в котором приняли участников женского пола на три недели, предоставив шагометр. Затем получали обратные отзывы у каждого участника. Участники были мотивированы и восприняли приложение в качестве виртуального тренера. Как Результат, аспект самоконтроля работает очень хорошо, потому что участники могли следить, как они идут к поставленной цели. Оглядываясь назад, можно сделать вывод что, положительная обратная связь была обнаружена у тех, кто использовал мобильное приложение для здоровья.

### **1.3.2 Упражнения и фитнес**

Физическая подготовка является не только одним из самых важных ключей к здоровому телу, оно является основой динамической и творческой интеллектуальной деятельностью.



Первоначально анатомия людей оставляет им право на свободное передвижение и наше тело требует движения для мышц и костей, чтобы процветать. О нашем историческом времени были подтверждения о том, что там было упоминание о спорте. Изначально активные движения имели место в целях выживания, как охота на продукты питания и обеспечение обитаемого пространства. В ходе цивилизации человечество начало развиваться придерживаясь в спорте с конкурирующими целями, как в Олимпийских играх у греков. В этот момент люди приступили к тренировкам и занимаясь с собственным весом, чтобы достичь полного потенциала, чтобы быть победителями. Данные из далекого прошлого не доступны, но разумно предположить, что ранние люди имели значительно более высокую калорийность расходов на единицу массы тела, чем это делают современные люди. Даже в промышленном периоде, после того, как первый паровой двигатель был разработан и, как сельскохозяйственные способы были уже созданы обществом людей, выросших свои собственные продукты питания, средний человек до сих пор сжигал больше калорий, чем в настоящее время. По мере того как технологическая эра возвысилась, физическая активность снизилась для среднего человека, но, как большинство наблюдателей заявили, было увеличение рекреационных видов спорта. Спорт достиг нового уровня участия. Обучение стало реальным вариантом для всех классов и курсов физической активности, медленно вновь начиная расти в конце 20-го века. В это время, все больше людей имели возможность тренироваться, чтобы стать спортсменом. Это началось в том числе с пакетов услуг личного тренера, соблюдения хорошей диеты и создания в целом более социально приемлемого образа жизни к концу 20-го. Люди из всех слоев общества хотели быть частью этого мира. Персональный тренинг теперь стал обыденным для всех людей. В фитнес-мире люди становятся все тяжелее и массивнее. Представляя новую моду и новый взгляд на бодибилдинг. В отрасли моды модели стали стройнее и здоровее, заставляя широкую общественность жаждать об аналогичном телосложении. Таким образом, персональный тренинг стал неотъемлемой частью жизни каждого. В начале 21-го века здоровый образ жизни стал тенденцией, в котором каждый человек хочет быть частью. Поскольку уровень ожирения вырос в мире от развития фаст-фуда, соблюдались высокие нормы обучения и здоровья. К примеру, если посмотреть на отчеты ассоциации производителей спортивных товаров около 61 млн американцев являются неактивными. На протяжении четырех лет число людей, которые являются неактивными увеличилось на 10,2 млн. Это число достаточно велико, когда смотришь на население Америки 320,560,631. Диаграмма указывает на уровень экономической неактивности растет здесь отображается ниже.

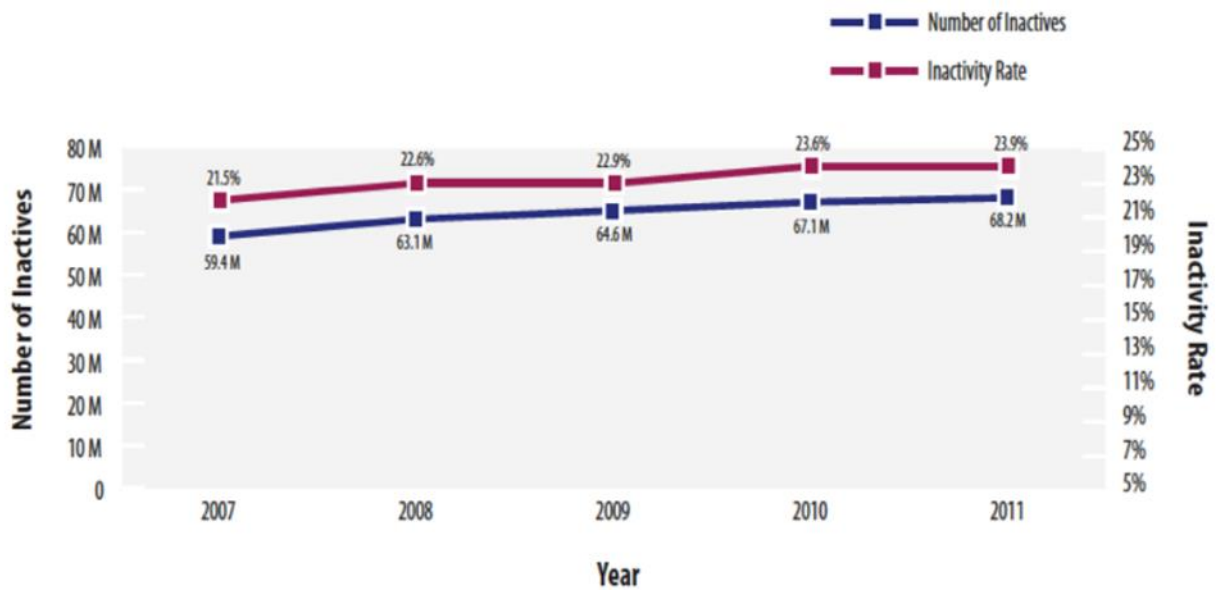


Рисунок 1.3 – Количество проживающих активной жизнью и неактивной

Хотя сравнительно существует четкое стремление американцев участвовать в рядах как фитнес, спорта и активном отдыхе, они не знают, с чего начинать. График показывает, различные виды спорта выбраны по возрасту и степени участия. Можно видеть, что как люди становятся старше, индивидуальная активность снижается, кроме только двух возрастающих факторов фитнес и не спорт. Это явный показатель того, что фитнес-приложение обращается к человеку в любом возрасте.

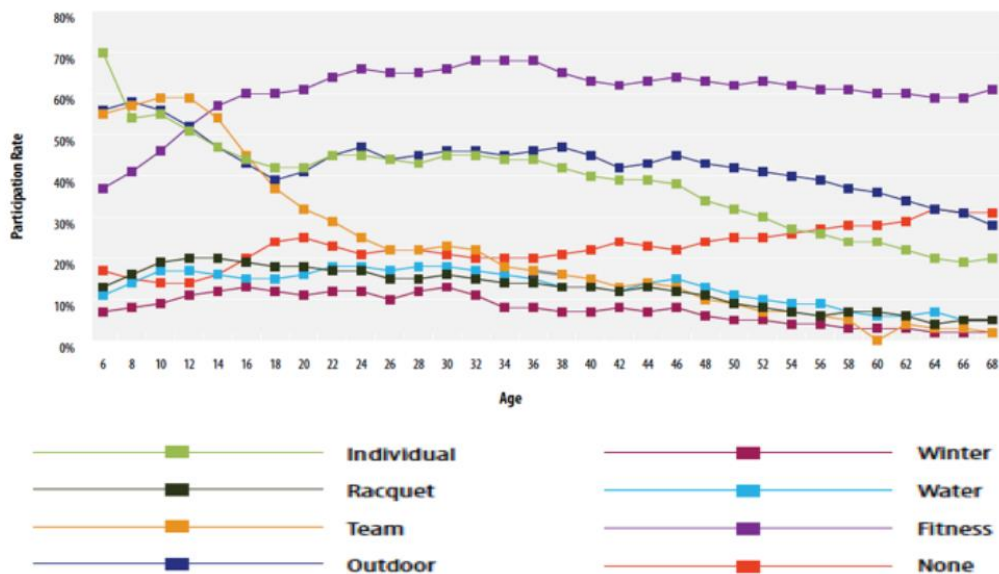


Рисунок 1.4 – Выбор вида спорта в зависимости от возраста

Варианты для фитнеса стали чрезвычайно разнообразны для публики, поэтому каждый человек может адаптировать свои возможности под различные программы. В соответствии с программой под названием «он нашел фитнес» три из пяти казахстанцев хотят потерять вес. Это показывает, что есть хороший рынок здоровья и фитнеса. Главное препятствие - это обращаться к общественности через среду которую они понимают, и которая их привлекает.

Технологии играют решающую роль в воспитании этого поколения. Это в основном то, что они лучше всего знают - можно обеспечить этих людей какой-либо технологии, и они были бы наиболее способные из всей рабочей силы, чтобы работать с этой технологией и адаптироваться к нему в быстрой манере. Вероятно, основной причиной для таких технически продвинутых возможностей является то что это первое поколение, которые имеют доступ к компьютерам из дома, поэтому сбор информации разных ресурсов стал одним из самых посещаемых мест, так как они могут быстро пользоваться огромным объемом информации. 97% этого поколения использует мобильные телефоны на ежедневной основе.

Рост Интернета является одним из основных деталей в воспитании поколения нашего времени и на вершине встал беспроводной Wi-Fi, который перевернул весь мир. Каждое поколение нашего времени обладают смартфонами, планшетами или небольшими другими гаджетами, которые позволяют им получить доступ к интернету через мобильное устройство. Что настолько удивительно, это сколько времени тратится на этих устройствах.

Средний пользователь тратит около 80% в день на мобильном устройстве. Это с учетом что мы не принимаем то, как часто люди используют другие интеллектуальные устройства для работы в различных отраслях. Один гаджет используется для чтения статьи, второй для электронной почты ответа, третий для социальных сетей и на стороне с устройствами связи. Кроме того, причиной успеха Интернета в деловом мире частично было связано с тем, что пользование электронной почтой было выбрано в качестве предпочтительного способа связи между бизнесом, клиентами и просто частных лиц. Существует огромное количество таких гаджетов, функции которых не могут быть проигнорированы и методы, используемые в предыдущих поколениях не работают достаточно эффективно, и не могут удовлетворять потребности современного человека.

### **1.3.4 Смартфоны**

Смартфон — это мобильный телефон со встроенными программами управления персональной информацией и операционной системой, что позволяет устройству получить доступ к компьютерному программному обеспечению для доступа в интернет. Благодаря этому доступу в интернет

можно использовать его для электронной почты, музыки и других приложений. По сути дела, смартфоны можно рассматривать как карманный компьютер. Было отмечено, что в третьем квартале 2012 года насчитывалось около одного миллиарда смартфонов, используемых потребителями во всем мире.

IBM была первой компанией, которая разработала смартфон в 1993 году телефон был с сенсорным интерфейсом, но имел ограниченный доступ к автоматизированным функциям. В связи с тем, что, рынок компьютерной памяти и интегральных схем становятся старыми и стоимость старых технологии уменьшается, прокладывается путь для более похожих на компьютер смартфонов. В 2001 году был введен первый мобильный телефон третьего поколения. Это означало, что сообщения и информации могли передаваться с достаточно высокой скоростью, которые будут получены, как музыка, фотографии, клипы и многое другое. Операционные системы, используемые этими смартфонами, как правило это, Windows mobile os от корпорации Microsoft, Android, Google, в то время как iPhone и исследований компании Apple в Motions Blackberry разработали свои собственные системы.

Одним из наиболее популярных функций в смартфонах является встроенная камера. Это позволяет записывать и передавать короткие клипы и фотографий. Почти каждый смартфон может получить доступ к беспроводным сетям, что позволяет пользователям использовать Интернет, упрощая сбор информации. Смартфоны также имеют встроенный чип системы глобального позиционирования (GPS) местоположения. Пользователь может использовать GPS, чтобы получить информацию из Интернета относительно интересующего места. По данным Business Insider последние данные от IDC показывает, что Android теперь имеет 82,8% мирового рынка смартфонов, когда, как, Apple имеет 13,9% рынка. В приведенной ниже таблице показано, разбивка между операционными системами крупнейших и доли рынка, связанной с ними.

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

Рисунок 1.5 – Доли в мировом рынке крупнейших мобильных операционных систем

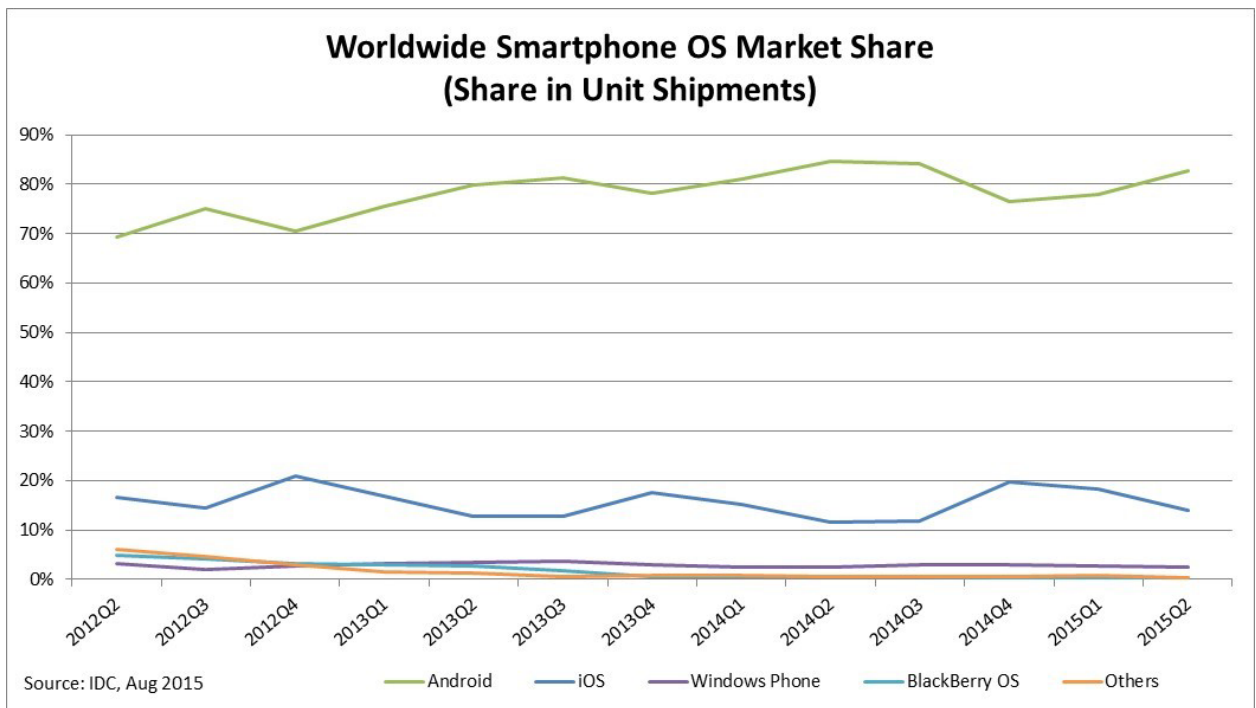


Рисунок 1.6 – Визуальное представление доли в мировом рынке крупнейших мобильных операционных систем

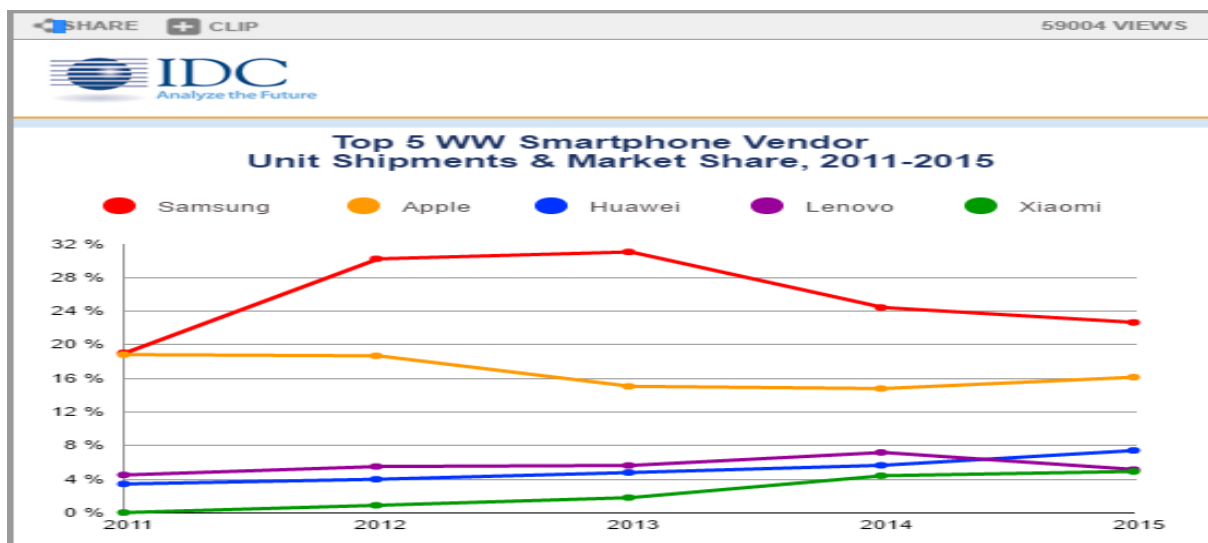


Рисунок 1.7 – Визуальное представление доли в мировом рынке крупнейших производителей мобильных телефонов

Энди Рубин основал Android в октябре 2003 года. Google поддержал его наряду со многими другими большими именами в мире разработки программного обеспечения, как Samsung и Motorola. Первый телефон, использовавший Android был выпущен в октябре 2008 года. HTC Dream был заклеен T-Mobile-ом, как поколение 1. Он включал в себя большинство встроенных приложений от Google, как Google Maps, календарь и Gmail.



Google play, который был запущен также в октябре 2008 года, позволяет пользователю приобрести платные и бесплатные приложения. Корпорация HTC выпустила смартфон, который может производить стереоскопические 3D эффекты в июне 2011 года и продажи Samsung Galaxy взлетели до 18 миллионов в третьем квартале 2012 года.

Apple представила оригинальный iPhone в 2007 году. Он был одним из первых мобильных телефонов использовавший мультитач интерфейс. Он очень быстро набрал популярность благодаря своему большому сенсорному экрану, которому не нужна клавиатура или стилус, как в этом прежде нуждались большинство телефонов и большинство телефонов. В 2008 году был выпущен iPhone второго поколения с более низкой ценой. В то же время компания Apple представила App Store, что позволило всем пользователям iPhone загружать сторонние и собственные приложения без необходимости компьютера. Клиентское программное обеспечение. iTunes, к которому с самого начала все яблочные продукты были связаны, также был использован в качестве средства для просмотра приложения. В 2010 году iPhone 4 был представлен с несколькими усовершенствованиями, включая переднюю камеру. После чего iPhone 5 в сентябре 2012 года был выпущен с новым дизайном, легким и тонким корпусом, чем когда-либо, было улучшено качество камеры и ввели много новых функций. С 2008 года app store вырос с 800 приложений до приблизительно 900000 в соответствии с объявлениями apple. Они породили много успешных предприятий и добавили существенную ценность к тому, за счет чего смартфоны имеют большую популярность. Согласно статистическим данным, опубликованным на сайте [www.statista.com](http://www.statista.com) и представленных на этом графике, это наиболее популярные категории App Store, записанные до марта 2016 года.

Можно увидеть, что игры занимают первую строчку среди загруженных приложений, здравоохранения и фитнес занимает около 2,96% доступных активных приложений. Но здесь есть большой потенциал для роста, особенно среди молодого населения, как было упомянуто прежде, они поколения близкое к технологиям, и имеют большой интерес к тенденциям в области здравоохранения.

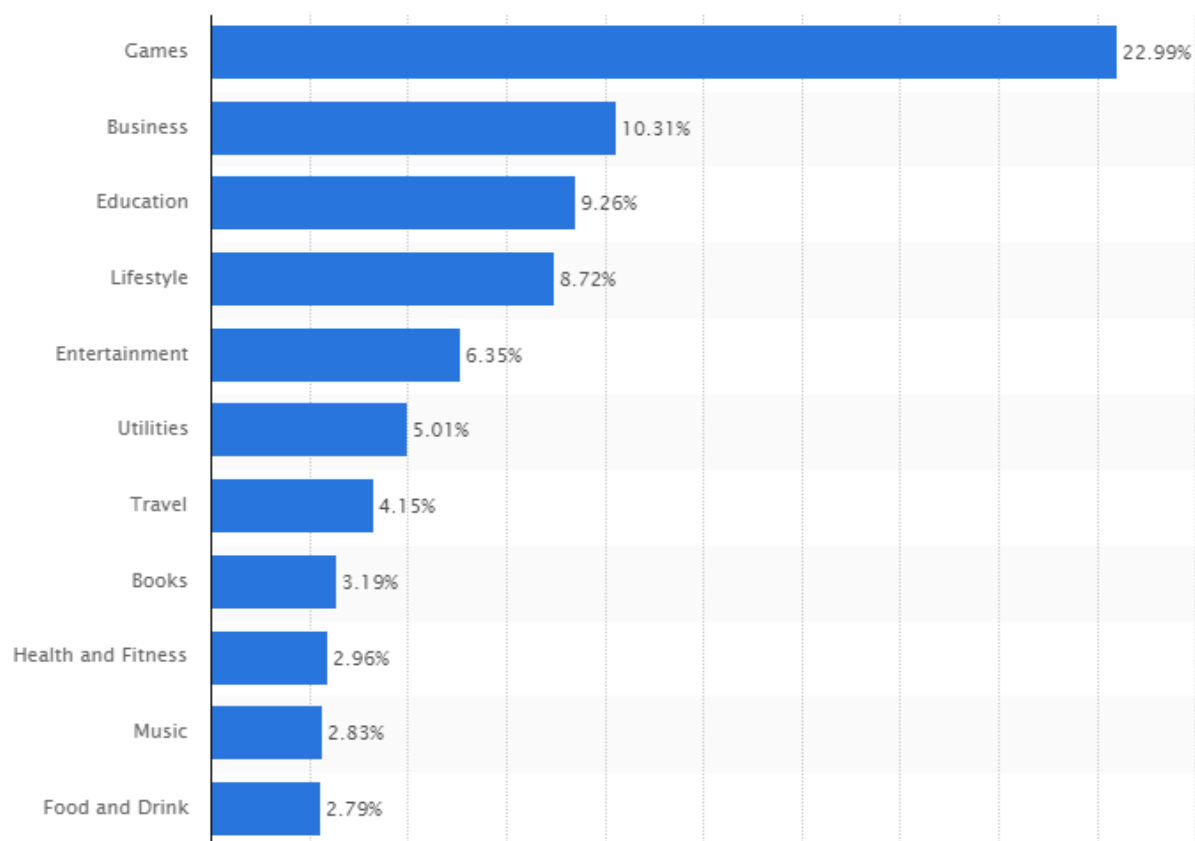


Рисунок 1.8 – Соотношение категорий приложений (март 2016 г.)

## 2 Классификация моделей представления знаний

### 2.1 Формализация знаний в интеллектуальных системах

Как было упомянуто выше, основным компонентом интеллектуальных систем является база знаний. Ее правила описывают закономерности в данной предметной области. Несмотря на достаточно длительный период существования и использования интеллектуальных систем, разработчики по-прежнему сталкиваются с проблемой формирования своих баз знаний.

В данное время существуют два ключевых подхода к формализации знаний в интеллектуальных системах:

- с использованием инженерного метода;
- с использованием алгоритмов и методов интеллектуального анализа данных.

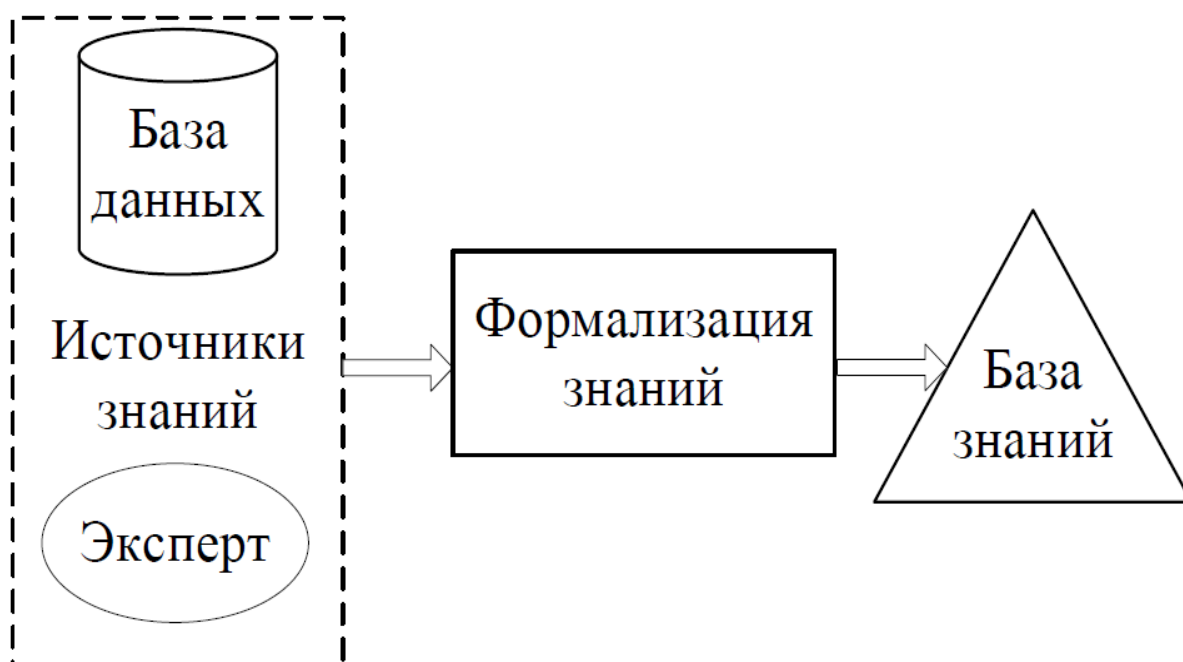


Рисунок 2.1 – Схема формализации знаний в интеллектуальных системах

До недавнего времени для формализации знаний в интеллектуальных системах применялся метод инженерии знаний. Этот метод основан на получении знаний непосредственно у экспертов в данной предметной области и запрашивает большой аналитической работы: эксперту может быть трудно,

а иногда невозможно выразить свои знания, опыт и интуицию в строгих рамках формальных моделей представления знаний.

В последние годы актуальность приобретения методов автоматического создания баз знаний, основанной на применении методов и алгоритмов интеллектуального анализа данных. Такой подход значительно упрощает и ускоряет развитие интеллектуальной системы.

Очевидно, что если имеется достаточное количество экспериментальных данных с использованием второго подхода является более предпочтительным, так как в дополнение к сокращению времени и упрощения общего процесса получения знаний эксперт предметной области может участвовать в процессе оценки сгенерированных правил и законов, которые согласуются с его личными знаниями, опытом и интуицией.

Тем не менее, несмотря на все преимущества этого подхода, в процессе познания часто генерируются сотни или тысячи правил, которые составляют базу знаний интеллектуальной системы, с одной стороны, усложняет работу эксперта в своем анализе и интерпретации, и делает базу знаний излишним. Все это следствие может существенно повлиять на точность и скорость интеллектуальной системой решений, и сводит на нет преимущества процесса создания знаний.

Для повышения эффективности использования интеллектуальных систем необходимо сделать сокращение баз знаний путем структурного упорядочения и сведение к минимуму правил принятия решений, не теряя своей полноты, последовательности и точности, взятая на их основе решений.

## **2.2 Продукционная модель представления знаний**

При формировании базы знаний интеллектуальной системы важно выбрать модель представления знаний. Существует множество моделей представления знаний, среди которых могут быть выделены следующие основные классы:

- модели семантические сети;
- фреймовые модели;
- формальные логические модели;
- продукционные модели.

Наиболее широкое распространение получили продукционные модели представления знаний. Основателем представления знаний в виде продукционных правил является Э. Посту.

В целом, знания представляются в виде нормативных правил, которые имеют следующий вид:

«ЕСЛИ <перечень условий> ТО <множество действий>», где ЕСЛИ и ТО – ключевое слово продукционного правила.

Левая сторона правил называется антецедентом (предпосылка), и правая консеквентом (вывод). Если антецедент является верным, то следует

выполнить консеквент. Условия и многие действия можно описать с помощью логических выражений с использованием операторов «И», «ИЛИ» и «НЕ».

Есть разные структуры продукционных правил:

- простые – продукционные правила, в которых в антецеденте может быть одно условие, а в консеквенте может быть одно заключение;
- фокусирующие – продукционные правила, где в антецеденте несколько условий, и как следствие только одно заключение;
- разветвляющиеся – продукционные правила, в которых в антецеденте может быть одно условие, а в консеквенте может быть несколько заключений;
- составные – продукционные правила, где в антецеденте может быть несколько условий и в консеквенте несколько заключений;

Продукционные модели имеют следующие преимущества по сравнению с моделями семантических сетей и фреймовых моделей:

- значительная часть человеческого знания может быть представлено в виде продукций;
- простота разработки и применения;
- высокая интерпретируемость;
- наличие передовых механизмов логического вывода.

Таким образом, продукционная модель представления знаний является наиболее эффективным и востребованным для использования в интеллектуальных системах коммерческой эксплуатации по сравнению с другими существующими моделями представления знаний.

Можно рассмотреть продукционное правило: ЕСЛИ  $x_1 = A_1$  И  $x_2 = A_2$  И ...  $x_n = A_n$  ТО  $y = B$ . В этом примере входные параметры могут принимать только точные значения, так что, если есть пробелы, то применение этой системы становится невозможным. Обобщение продукционных правил являются нечеткие продукции, которые получили широкое применение при формировании нечетких баз знаний.

Примером нечетко-продукционного правила может служить своего рода нечеткая продукция «ЕСЛИ  $x_1 = \tilde{A}_1$  И  $x_2 = \tilde{A}_2$  И ...  $x_n = \tilde{A}_n$  ТО  $y = B$ », входные значения которой могут быть как четкими и нечеткими. Наиболее хорошо известные модели логического вывода являются модели Мамдани и Такаги-Сугено.

В случае необходимости, использование качественных данных прибегают к использованию синглтона в виде нечетких аналоговых значений с четкой функцией принадлежности, которая принимает одно из двух возможных значений: истина или ложь.

### **2.3 Фреймовая модель представления знаний**

Фреймовая модель – модель представления знаний, которая была введена М.Минским в 1975 году в качестве одного из подходов к описанию



знаний, полезных для понимания сцены и языка. Основной единицей информации является фрейм. Он описывается набором понятий и данных.

Каждый фрейм имеет определенную структуру и название в наборе слотов, которые в свою очередь имеют свою внутреннюю структуру. Фрейм может быть представлен в виде списка некоторых из свойств, определенных в слотах, а также отдельных интервалов, ассоциированных с процедурами присоединения.

Имя фрейма			
Имя слота	Значение слота	Способ получения значения	Присоединенная процедура

Рисунок 2.2 – Структура фрейма

В первых двух столбцах определяется имя слота и его значение. Значением может быть имя иного фрейма, в таком случае образуются сети фреймов. Последние два столбца описывают способ получения знаний и имя прилагаемых к ней процедуры. Организация вывода во фреймовой модели основана на обмене сообщениями между фреймами и на запуске присоединенных процедур.

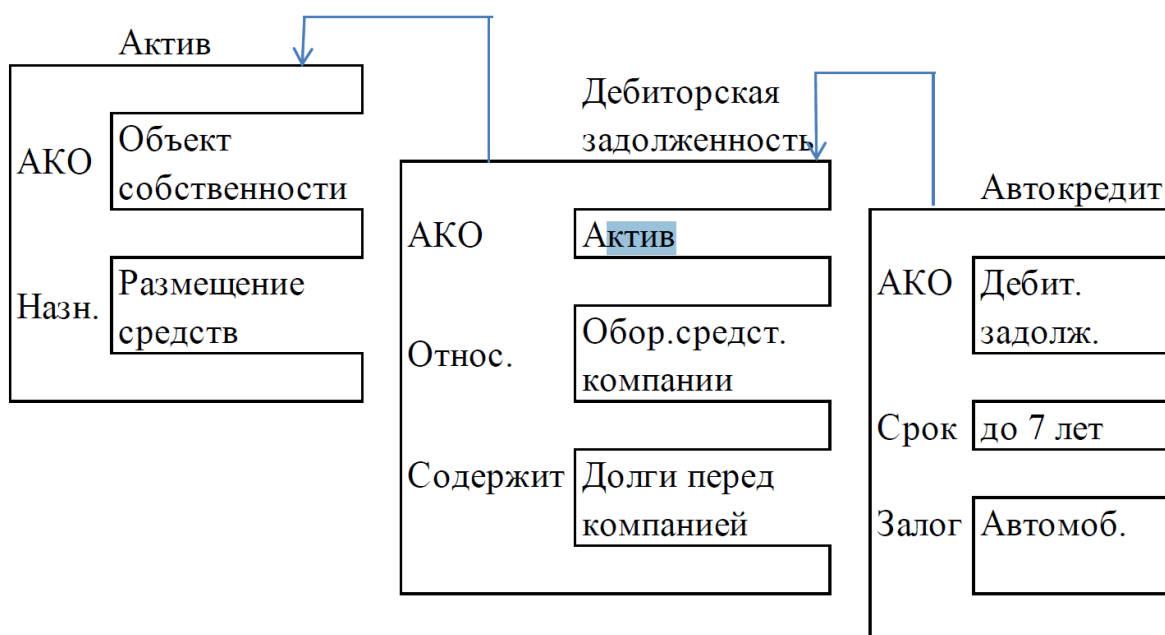


Рисунок 2.3 – Пример сети фреймов

В этом примере фрейм «Автокредит» наследует свойства фреймов «Дебиторская задолженность» и «Актив», находящихся на более высоком уровне в иерархии. Таким образом, на вопрос о том, к чему относятся «Автокредиты» следует ответить к «Оборотным средствам компании», так как

вся дебеторская задолженность имеет данное свойство. Наследование свойств может быть частичным. Важной особенностью фреймов является наследование свойств по АКО-связям. В слоте АКО указывается имя фрейма более высокого уровня иерархии, где наследуются атрибуты.

Основным преимуществом фреймовой модели является то, что она отражает концептуальную основу организации памяти человека и имеет такими характеристиками, как высокая структурированность, гибкость и видимость. Фреймовые модели имеют следующие недостатки:

- в связи с тем, что знание описывается в виде процедур, то становится трудно получить и изменять знание;

- использование фреймов возможно только при решении небольших задач, условие определяется тем, что с увеличением сложности задач, возникают трудности с описанием и управлением знаниями.

Еще одним достоинством фреймовых моделей представления знаний является то что, все термины, располагающиеся в каждом из узлов структуры, могут быть представлены комплексом атрибутов и их параметрами, которые находятся в слотах фрейма.

Слот — это атрибут, который сопрягается с узлом в структуре, построенной на фреймах. Он является главным аспектом фрейма. Имя слота дается исходя из четкого типа атрибута, а значением слота можно представить:

- экземпляр этого атрибута;
- ссылку на другой фрейм;
- ссылку на другой фасет.

1) Каждый уникальный слот может быть сопряжен с одним или несколькими процедурами, которые запускаются, когда трансформируются показатели слотов. Обычно со слотами сопрягаются такие процедуры:

2) ЕСЛИ-ДОБАВЛЕНО (IF-ADDED) — запускается тогда, когда новая инфа появляется в слоте;

3) ЕСЛИ-УДАЛЕНО (IF-REMOVED) — запускается при удалении инфы из слота;

4) ЕСЛИ-НУЖНО (IF-NEEDED) — запускается при запросе инфы из слота в том случае, когда значение слота равно нулю.

Эти процедуры отслеживают преобразование инфы, соответствующей данному узлу, и отслеживают, чтобы при преобразовании тех или иных параметров выполняются необходимые действия.

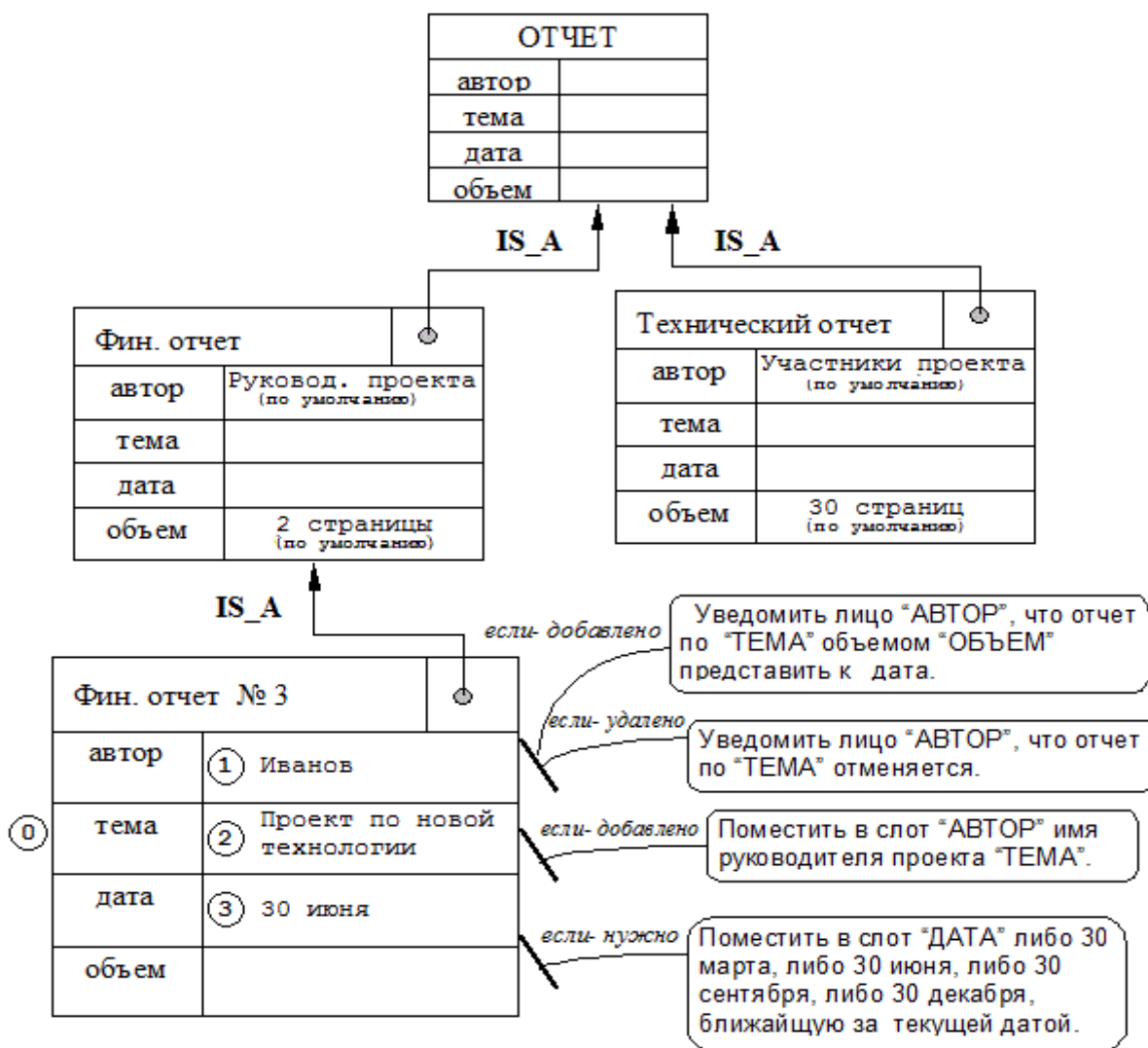


Рисунок 2.4 - Функционирование фреймовой системы

## 2.4 Семантические сети

Семантические сети были введены Квилианом в 1968 году для изучения искусственного интеллекта как способ описания человеческой памяти. Понимание смысла с использованием семантических сетей позволяет выйти за рамки обычных возможностей программного обеспечения интеллектуальных систем. В моделях семантической сети знание предметной области описывается как ориентированный граф. С вершинами связаны соответственно конкретные объекты предметной области, а дугам – семантические отношения между ними. Характерной особенностью моделей семантической сети является наличие трех классов: класс, пример элемента класса и свойство.

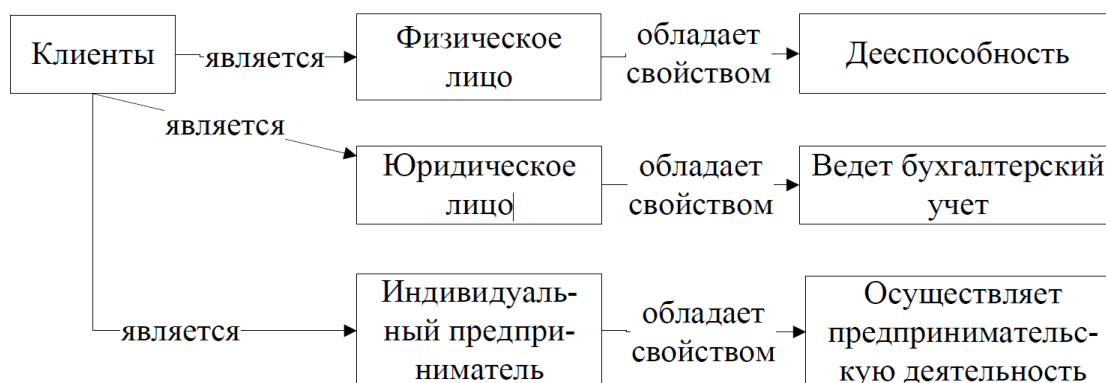


Рисунок 2.5 – Разделение моделей временных рядов

Задача нахождения решения на основе данной модели представления знаний сводится к поиску подсети, отражающей поставленный запрос к базе знаний. Таким образом, недостатком этой модели является сложность вывода на семантической сети.

Для таких отношений основным является то, что экземпляры понятий нижнего уровня имеют все атрибуты мнений верхнего уровня. Это свойство называется наследованием атрибутов между уровнями иерархии (IS - A).

Отношение «целое – часть» можно описать словами, которые демонстрируют то, что примеры понятия «процессор» считаются частью любого примера мнения «компьютер». «Процессор» (PART - OF) «компьютер».

Отношения типа (PART - OF) дают возможность рассмотреть некоторые общие умения (набор атрибутов) для особенного класса мнений. В частности, показанный ранее метод отношения рассказывает, что особенным свойством всех компьютеров является расположение в них процессоров. И данное свойство будет передаваться всем примерам мнения ноутбук.

Особенно часто используется графическое отображение семантических сетей как диаграммы.

Элементы семантической сети. Семантическая сеть изображает ориентированный граф с вершинами и названными дугами. Основными элементами сети считаются дуги и вершины. При этом вершинам семантической сети соответствуют понятия, свойства и события, как показано на рисунке 2.6.

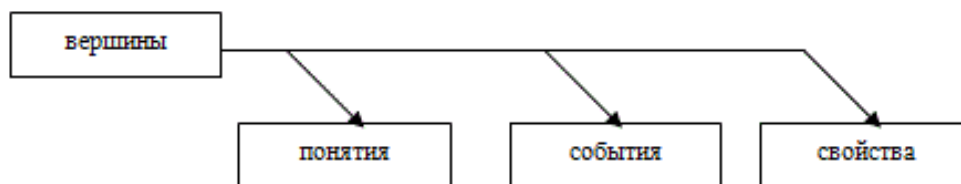


Рисунок 2.6 – Вершины семантической сети

Понятие — это информация об абстрактных или физических объектах предметной области или реального мира.

Событие — это действие, происходящее в реальном мире и определяется:

- указанием вида действия;
- указанием ролей, которые в этом действии играют объекты.

Свойство — используется для уточнения событий и понятия. Что касается концепции, которые описывают их характеристики и особенности (цвет, размер, качество), так и по отношению к событиям — место, время, продолжительность.

Дуги графа семантической сети — представляют собой различные семантических отношений, которые относительно можно разделить на четыре класса (рисунок 2.7).



Рисунок 2.7 - Классификация семантических отношений

### 3 Разработка базы знаний

#### 3.1 Подготовка к работе

Разработка базы знаний для мобильного приложения является не легкой работой, выполняется разработчиком самостоятельно и состоит из нескольких этапов. База знаний, содержащая знания экспертов и логические правила в области здравоохранения, является важнейшей частью в работе мобильного приложения, которое включает в себя помимо базы знаний так же базу данных и пользовательский интерфейс.

Разработки приложения проводится на платформе Android Studio v2.1.1-windows на современном объектно-ориентированном языке программирования Java. Для начала работы необходимо установить вышесказанную платформу (рисунок 3.1).



Рисунок 3.1 – Установка Android Studio 2.1.1




Установка длится в несколько шагов, в которых указывается путь установки, указывается название конечной папки, выбирается полная либо частичная установка, указываются дополнительные компоненты необходимые

для полноценной работы для создания мобильных приложений. После окончания, нужно скачать утилиты работы Android Studio, такие как jdk и sdk.

Следующий шаг – это регистрация на сайте GitHub, основная часть изображена на рисунке 3.2. Регистрация необходима для хранения резервных копии проекта на сервере портала. Также в архивах данного сайта можно найти множество советов и полезных статей по проектированию различных работ для мобильного приложения.

## Join GitHub

The best way to design, build, and ship software.

 <b>Step 1:</b> Set up a personal account	 <b>Step 2:</b> Choose your plan	 <b>Step 3:</b> Go to your dashboard
---	--	--

### Create your personal account

**Username**

This will be your username — you can enter your organization's username next.

**Email Address**

You will occasionally receive account related emails. We promise not to share your email with anyone.

**Password**

Use at least one lowercase letter, one numeral, and seven characters.

---

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

---

[Create an account](#)

### You'll love GitHub

- Unlimited** collaborators
- Unlimited** public repositories
- ✓ Great communication
- ✓ Friction-less development
- ✓ Open source community

Рисунок 3.2 – GitHub

После регистрации разработчик получает возможность связать свой проект в Android Studio, и проект будет храниться в хранилище, как показано на рисунке 3.3. В любое время пользователь может получить доступ к своему приложению на сервере, даже если у него отсутствует личный компьютер, разработчик может зайти под своим аккаунтом портала и получить собственный проект на иной переносной либо персональный компьютер,



необходимо лишь наличие Android Studio соответствующей или новейшей версии и доступа к сети интернет.

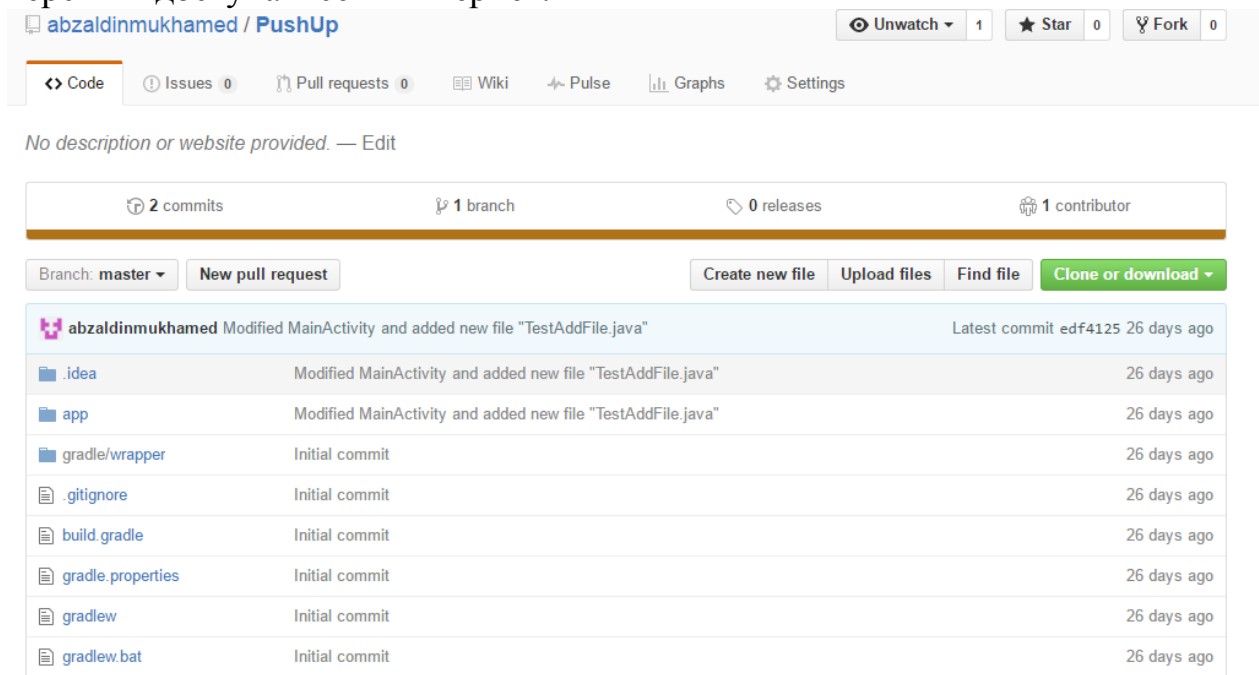


Рисунок 3.3 – Обзор проектов на GitHub

## 3.2 Проектирование базы знаний

Начало работы в Android Studio 2.1.1, создаем новый проект, устанавливаем название проекта и на экране выводится главное окно Android Studio. Главное окно включает в себя панель управления, панель инструментов, панель отладки, панель структуры приложения и мобильного устройства в правой стороне экрана. Все упомянутое отображено на рисунке 3.4.

Разработка приложения состоит в том, что программируется код, который:

- создается класс `CreateAccount`;
- который наследует функции класса `AppCompatActivity`;
- назначаются действия к кнопкам;
- учитываются исключения при вводе данных.

Указанные действия изображены на рисунке 3.5, на котором показана часть кода.

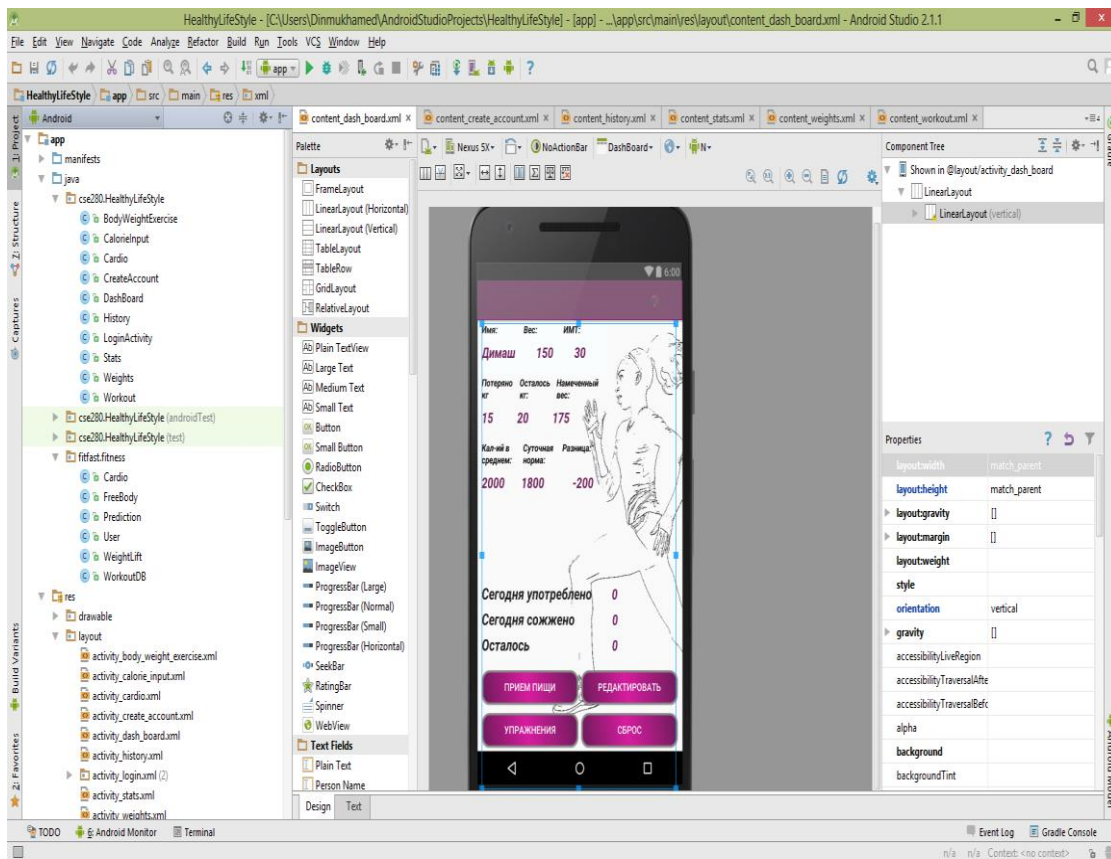


Рисунок 3.4 – Общий вид Android Studio

```

public class CreateAccount extends AppCompatActivity {

    SharedPreferences sPref;
    private final String PREFS = "SharedPreferences";
    private final String genderKey = "gender";
    private final String activityKey = "activity_level";
    private final String timeKey = "time_frame";
    private final String nameKey = "name";
    private final String ageKey = "age";
    private final String heightKey = "height";
    private final String weightKey = "weight";
    private final String goalKey = "goal";
    private final String passwordKey = "password";
    private final String desiredWorkoutsKey = "desired_workouts";
    private Context mContext;

    RadioButton male;
    RadioButton female;
    RadioButton one;
    RadioButton two;
    RadioButton three;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_account);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        sPref = getSharedPreferences(PREFS, Context.MODE_PRIVATE);

        male = (RadioButton) findViewById(R.id.male);
        female = (RadioButton) findViewById(R.id.female);
        one = (RadioButton) findViewById(R.id.one);
        two = (RadioButton) findViewById(R.id.two);
        three = (RadioButton) findViewById(R.id.three);
    }
}

```

Рисунок 3.5 – Создание класса CreateAccount

На рисунке 3.6 видно, что в классе User, хранятся базы знания экспертов и необходимые коэффициенты. Далее на главном меню мы видим коэффициент массы к телу и количество каллории для сохранения массы тела и количество обязательных для сжигания каллории.

```
public static double CalcBMI( double bodyWeight, double height )
{
    return ( bodyWeight / ( height/100 * height/100 ) ); // BMI calculation and return
}
/*
@pre-condition: takes four parameters: a boolean, two doubles and an int
@post-condition: returns the Base metabolic rate as a double based on parameters
*/
public static double CalcBMR( boolean gender, double bodyWeight, double height, int age )
{
    if ( gender )
    {
        return 655 + ( 4.35 * bodyWeight ) + ( 4.7 * height ) - ( 4.7 * age ); // female
    }
    else
    {
        return 66 + ( 6.23 * bodyWeight ) + ( 12.7 * height ) - ( 6.8 * age ); // male
    }
}
/*
@pre-condition: takes five parameters: a boolean, two doubles and two ints
@post-condition: returns maintenance calories as a double approximated by activity level
*/
public static double CalcDailyCals( boolean gender, double bodyWeight, double height, int age, int activityLevel )
{
    double baseCal = CalcBMR( gender, bodyWeight, height, age );
    double coefficient = 1;

    switch( activityLevel )
    {
        case 1:
            coefficient = 1.2;
            break;
        case 2:
            coefficient = 1.375;
            break;
        case 3:
            coefficient = 1.55; // increasing activity levels = more calories
    }
}
```

Рисунок 3.6 – Класс User

Рисунки 3.7, 3.8 и 3.9 демонстрируют работу классов где хранятся коэффициенты необходимые для расчета сожженных каллории.

```

public class BodyWeightExercise extends AppCompatActivity implements AdapterView.OnItemClickListener {
    public static String bodyWorkout;
    private final String PREFERENCES = "SharedPreferences";
    private SharedPreferences sPref;
    Integer caloriesBurned;
    TextView caloriesBurnedDisplay;
    final String caloriesKey = "calories_burned";

    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    @Override
    public void onResume() {...}

    public void onItemClick(AdapterView<?> parent, View view, int pos, long id) {...}

    public void onNothingSelected(AdapterView<?> parent) {...}

    public void calcBurned(View view){
        EditText repsText = (EditText) findViewById(R.id.edit_time);

        EditText setsText = (EditText) findViewById(R.id.edit_distance);

        String type = "Body Weight";
        String workout = bodyWorkout;
        int calBurnedFromBody = 0;

        int sub = 0;
        double coeff = 0;

        if(workout.equals("Отжимания")){
            sub = 0;
            coeff = 0.0009;
        }
        if(workout.equals("Приседания")){
            sub = 0;
            coeff = 0.000655;
        }
    }
}

```

Рисунок 3.7 – Демонстрация класса BodyWeightExercise

```

package cse280.HealthyLifeStyle;

import ...

public class Cardio extends AppCompatActivity implements AdapterView.OnItemClickListener {
    public static String cardioWorkout;
    private final String PREFS = "SharedPreferences";
    private SharedPreferences sPref;
    Integer caloriesBurned;
    TextView caloriesBurnedDisplay;
    final String caloriesKey = "calories_burned";

    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    @Override
    public void onResume() {...}

    public void onItemClick(AdapterView<?> parent, View view, int pos, long id) {...}

    public void onNothingSelected(AdapterView<?> parent) {...}

    public void calcBurned(View view) {
        EditText timeText = (EditText) findViewById(R.id.edit_time);

        EditText distanceText = (EditText) findViewById(R.id.edit_distance);

        String type = "cardio";
        String workout = cardioWorkout;
        double coef = .1;
        if(workout.equals("Ходьба")){
            coef = .0085;
        }
        if(workout.equals("Бег")){
            coef = .0124;
        }
    }
}

```

Рисунок 3.8 – Демонстрация класса Cardio

```

public void onNothingSelected(AdapterView<?> parent) {...}

public void calcBurned(View view) {

    EditText weightText = (EditText) findViewById(R.id.edit_weight);

    EditText repsText = (EditText) findViewById(R.id.edit_time);

    EditText setsText = (EditText) findViewById(R.id.edit_distance);

    String type = "Weights";
    String workout = weightsWorkout;
    int calBurnedFromBody = 0;
    int sub = 0;
    double coeff = 0;

    if(workout.equals("Подъем Гантели На Бицепс")){
        sub = 0;
        coeff = 0.005832861;
    }
    if(workout.equals("Подъем Туловища")){
        sub = 0;
        coeff = 0.00661057;
    }
    if(workout.equals("Приседания")){
        sub = 1;
        coeff = 0.00972;
    }
    if(workout.equals("Становая тяга")){
        sub = 1;
        coeff = 0.00972;
    }
}

```

Рисунок 3.9 – Демонстрация класса Weights

Далее мы создали классы в базе знания которых будут храниться соответствующие формулы для вычитывания количества сожженных пользователем калорий.

```

public class Cardio
{
    private static double calories;

    /*
    @pre-condition: takes four doubles as parameters
    @post-condition: returns the number of calories burned as an int based on parameters
    */
    public static int calcCalories(double bodyWeight, double distance, double time, double coeff)
    {
        double pace = ( distance / time ) * 60;

        calories = bodyWeight * pace * coeff * time; // cardio formula
        int cals = (int) calories;

        return cals;
    }
}

```

Рисунок 3.10 – Демонстрация класса Cardio

```

public class FreeBody
{
    private static double calories;

    /*
    @pre-condition: takes five parameters: two doubles and an int
    @post-condition: returns an int of calories burned by workout based on parameters
    */
    public static int calcCalories( double bodyWeight, int reps, int sets, double coeff, int subtype )
    {
        calories = bodyWeight * coeff * reps * sets; // formula for calories burned

        int cals = (int) calories;

        return cals;
    }
}

```

Рисунок 3.11 – Демонстрация класса FreeBody



```

public class WeightLift
{
    private static double calories;

    /*
    @pre-condition: takes six parameters, three doubles and three ints
    @post-condition: returns calories burned by weightlifting activity as an int based on parameters
    */
    public static int calcCalories( double bodyWeight, double weight, int reps, int sets, double coeff, int subType )
    {
        if (subType == 0 )           //workout that doesn't incorporate body weight
            calories = weight * coeff * reps * sets ;           //calories burned by weightlifting activity
        else // workout includes bodyweight
            calories = ( weight + bodyWeight ) * coeff * reps * sets; //calories burned by weightlifting activity

        int cals = (int) calories;

        return cals;
    }
}

```

Рисунок 3.12 – Демонстрация класса WeightLift

На рисунке 3.13 показана создание нашей базы данных где будут хранятся введенные данные о пользователе и данные об упражнениях. Генерируется Map<String, Integer>. В следующем этапе выводится ссылка для указанного класса в виде ключа, а в виде значения ссылки на введенные данные.

```

public class WorkoutDB
{
    private static Map<String,Integer> idList = new HashMap<>();
    private static Map<Integer,Double> coList = new HashMap<>();

    public WorkoutDB()
    {
        super();
        loadDB();
    }

    public static void loadDB()
    {
        idList.clear();
        //cardio
        idList.put("run",0);
        idList.put("walk",1);

        //bodyweight
        idList.put("pushups",10);
        idList.put("situps",11);

        //weightlifting
        idList.put("curls",100);
        idList.put("benchpress",101);
        idList.put("squats",102);
        idList.put("deadlifts",103);

        coList.clear();
        coList.put(0,0.0124);
        coList.put(1,0.0085);

        coList.put(10,0.0009);           // 0.036 per min *assume 40/min, 0.0009 cal/pushup
        coList.put(11,0.000655);       // 0.0262 per min *assume 40/min, 0.000655 cal/pushup
    }
}

```

Рисунок 3.13 – Храним информацию в базе данных

### 3.3 Результаты работы приложения

По итогам разработки получена мобильное приложение, включающее в себя такие функции как, предоставление полной информации о здоровье пользователя и хранение данных пользователя. Эти части доступны в готовом проекте как показано в последующих рисунках.



Рисунок 3.14 – Главное меню

Главное меню состоит из четырех разделов, доступных пользователю. За прием пищи отвечает вкладка «Прием пищи», перейдя на которую, нам станет будет доступно поле, где мы можем ввести количество съеденных калории. На рисунке 3.15 изображен Layout приема пищи эмулятора, на котором видна вкладка «Ввод» и вкладка возврата «На главную».

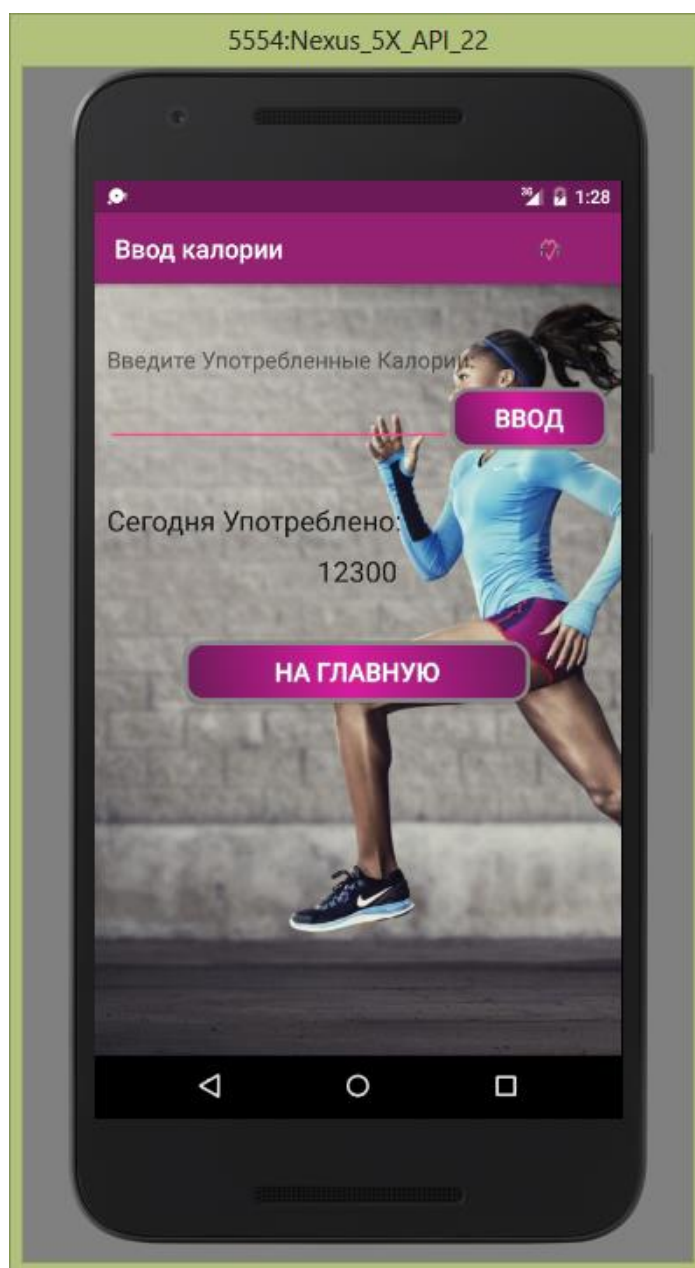


Рисунок 3.15 – Вкладка Прием пищи

Далее, после введения данных пользователь может перейти обратно на главное меню. За физические упражнения отвечает вкладка «Упражнения», перейдя на которую пользователю предстоит выбрать тип физического упражнения 3.16.

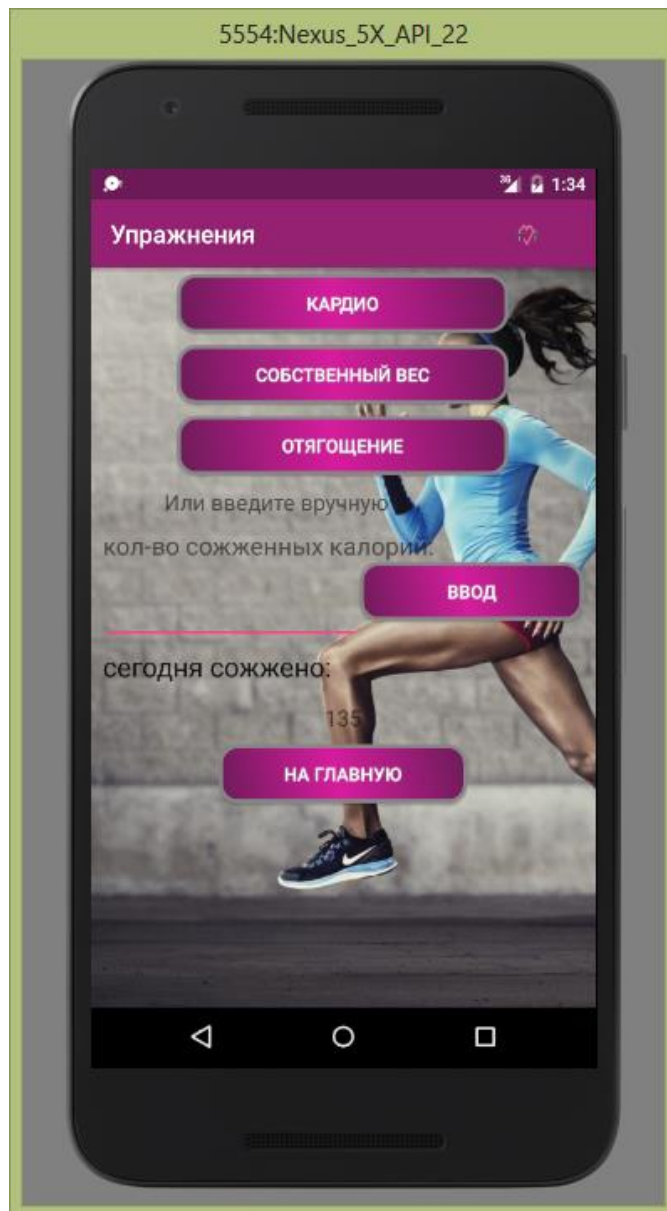


Рисунок 3.16 – Вкладка Упражнения

Как видно на рисунке, данный Layout имеет поле вывода введенных данных которое будет отображать высчитанные приложением калории по введенным данным в соответствующих вкладках. Каждая вкладка соответственно имеет подтип упражнения, которые пользователь может выбрать.

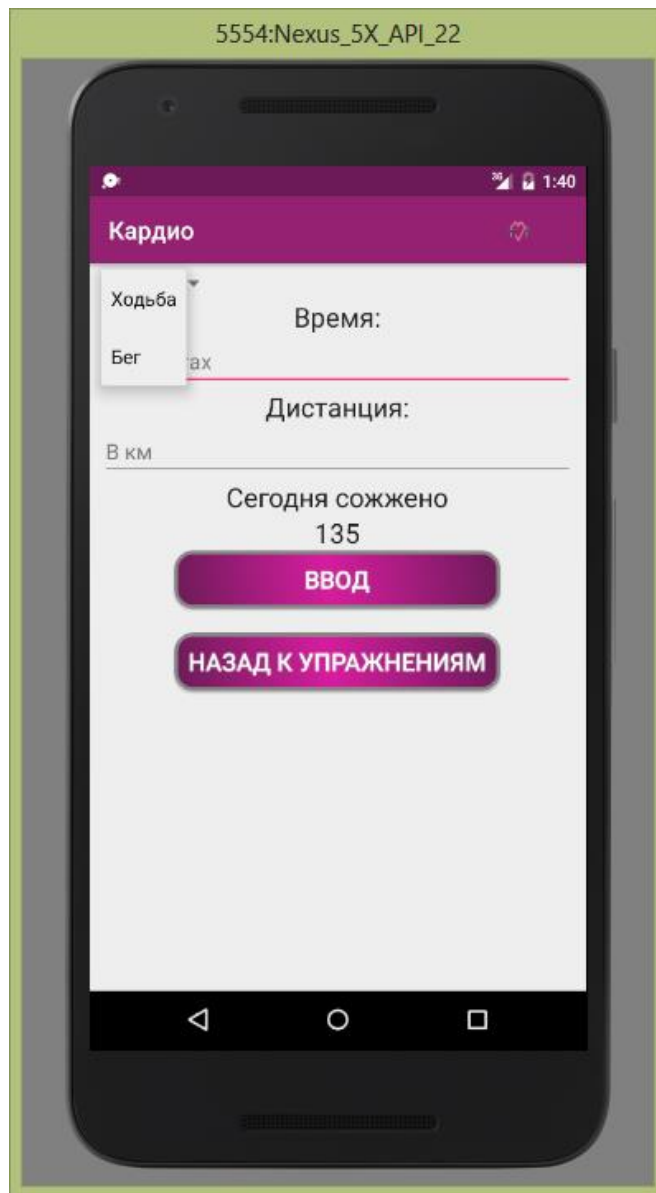


Рисунок 3.17 – Вкладка Кардио

Как видно на рисунке, данная вкладка имеет поля ввода и вывода введенных данных. В выпадающем окне пользователю предоставляется выбрать тип упражнения полезных для сердечно–сосудистой системы.

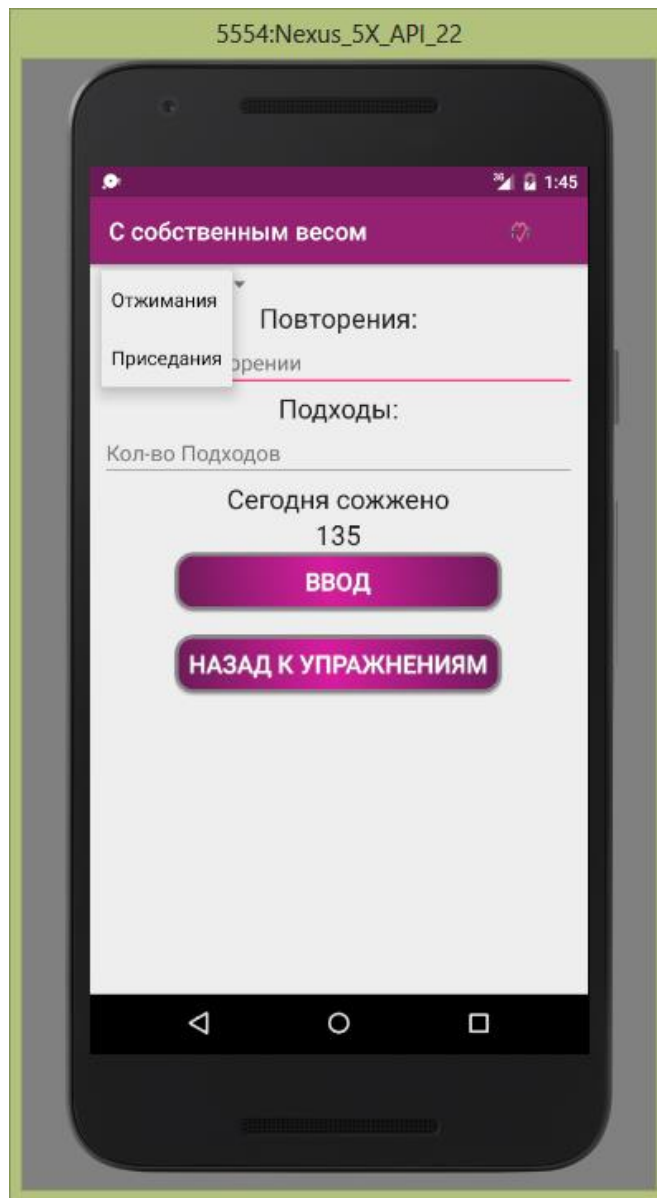


Рисунок 3.18 – Вкладка Собственный вес

Как видно на рисунке, данная вкладка имеет поля ввода и вывода данных. В выпадающем окне пользователю предоставляется выбрать тип упражнения без отягощения. Вводятся соответственно повторения и подходы проделанного упражнения.

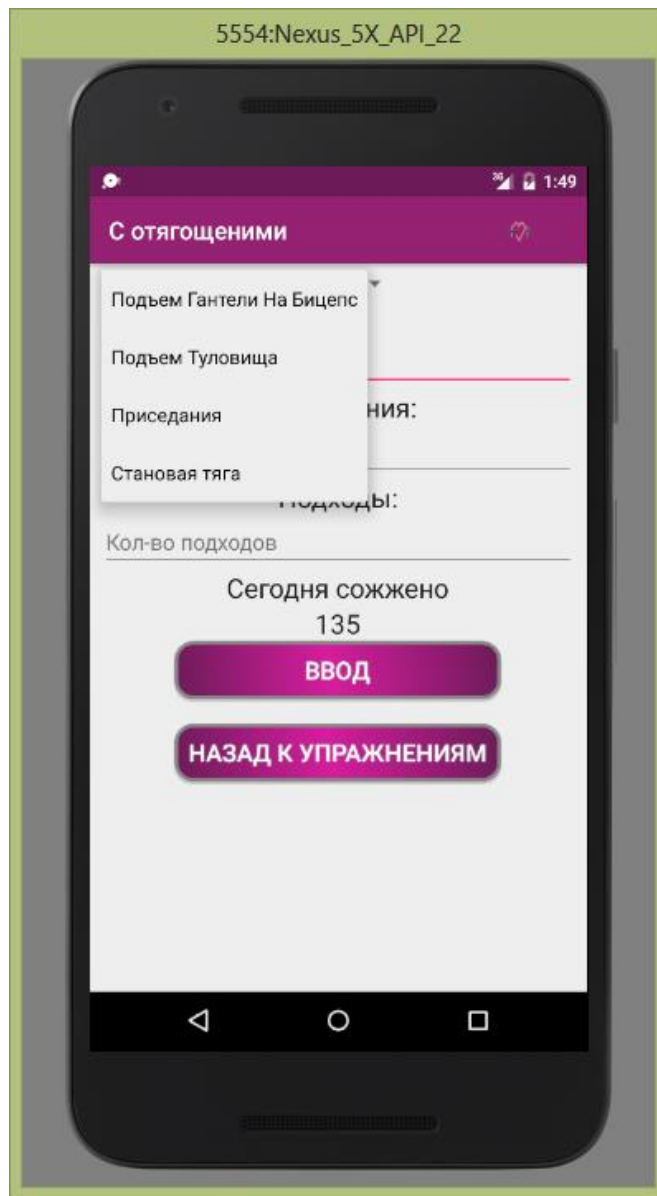


Рисунок 3.19 – Вкладка Собственный вес

Как видно на рисунке, данная вкладка также имеет поля ввода и вывода данных. В выпадающем окне пользователю предоставляется выбрать тип упражнения с отягощением. Вводятся соответственно добавочный вес, повторения и подходы проделанного упражнения.





Рисунок 3.20 – Вкладка Прием пищи

Пользователь в любой момент может отредактировать данные введенные им при регистрации, перейдя по вкладке «Редактировать», как показано на рисунке 3.20.

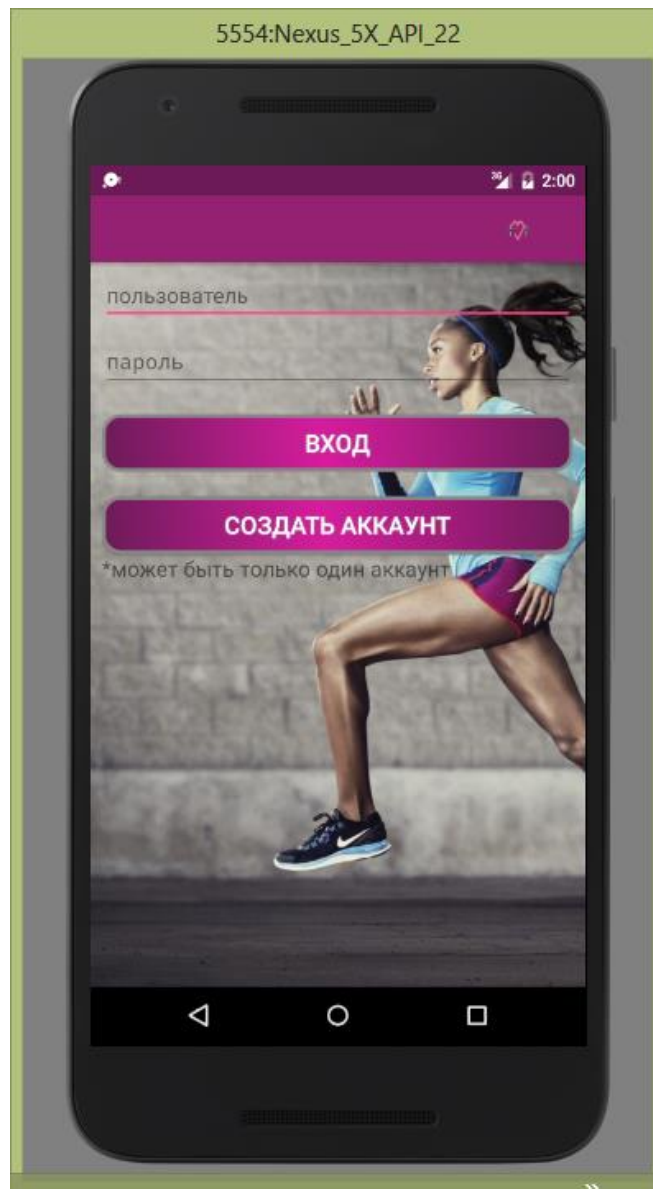


Рисунок 3.21 – Форма аутентификации

В мобильном приложении спроектирована форма запроса, логина и пароля, для хранения данных и действий прошлого сеанса, показанная на рисунке 3.21. В каждом устройстве может быть один пользователь. В случае отсутствия логина у пользователя, имеется вкладка «Регистрация».

В этой главе была подробно описана подготовка к проектированию, сама разработка и принцип работы мобильного приложения, включающего в себя базу знаний. Описание дополнено письменной инструкцией и скринами с демонстрацией работы мобильного приложения.

## 4 Технико-экономическая часть

### 4.1 Цель проекта и его разработка

Целью моей дипломной работы является разработка базы знаний для мобильного приложения «Здоровый образ жизни»

Люди пользуются услугами фитнес центров и рассчитывают получить здоровое тело, основываясь на консультациях фитнес инструкторов и здоровом питании. Мобильное приложение поможет людям в достижении успеха, предоставляя всю возможную информацию и расчеты. Приложение, основываясь на своей базе знаний, помогает человеку рассчитать коэффициент здоровья и выбрать оптимальную программу на определенный срок. Приложение разрабатывалась в среде Android Studio.

Проектирование и разработка базы знаний для мобильного приложения состоит из 5 этапов:

- постановка задачи;
- разработка базы знаний;
- разработка интерфейса приложения;
- тестирование на работоспособность;
- оформление и сдача отчета.

В таблице 4.1 приведен график реализации проекта.

Таблица 4.1 – Этапы реализации и проектирования

Перечень работ		Неделя от начала работ						
1 этап	Постановка задачи	■						
	Изучение литературы	■	■					
	Изучение методов разработки		■					
2 этап	Разработка базы знаний			■	■	■		
3 этап	Разработка интерфейса						■	■
4 этап	Тестирование приложения							■
5 этап	Оформление и							■



программные продукты:

- OS Windows 8.0– операционная система;
- Android Studio 2.1 – среда для разработки;

Программные продукты и соответствующие им стоимость приведены в таблице 4.4.

Таблица 4.4 – Перечень программного обеспечения приведены в таблице

Наименование	Количество	Стоимость за единицу
Windows 8.0	1	18394
Microsoft Office Professional Plus	1	Бесплатно
Android Studio 2.1	1	Бесплатно
Java Development Kit	1	Бесплатно
Итого:		18 94

#### 4.5 Расчет и определение затрат на создание и реализацию проекта

##### 4.5.1 Расходы на оплату труда основным разработчикам и программистам

Основные затраты на реализацию программного продукта ложатся на выплату заработной платы работникам, которые вычисляются по формуле:

$$\text{ФОТ} = \text{З}_{\text{осн}} + \text{З}_{\text{доп}} \quad (4.2)$$

где  $\text{З}_{\text{осн}}$  – основная заработная плата;

$\text{З}_{\text{доп}}$  – дополнительная плата.

Для высчитывания затрат на основную заработную плату используются числа о зарплате в среднем в день и фактическом времени работы каждого рабочего.

Средний заработок в день:

$$D = \frac{\text{ЗП}_m}{D_p} \quad (4.3)$$

где  $\text{ЗП}_m$  – ежемесячный размер заработной платы;

$D_p$  – количество рабочих дней в месяце (21 день).

Дизайнер:

$$D = \frac{50\,000}{21} = 2381 \text{ тенге/день}$$

Разработчик:

$$D = \frac{75\,000}{21} = 3571 \text{ тенге/день}$$

Один час работы рассчитывается по формуле:

$$H = \frac{D}{\text{Ч}_p} \quad (4.4)$$

где  $D$  – средний дневной заработок работника;

$\text{Ч}_p$  – количество часов рабочего дня (8 часов).

Дизайнер

$$H = \frac{2381}{8} = 298 \text{ тенге/час}$$

Разработчик

$$H = \frac{3571}{8} = 446 \text{ тенге/час}$$

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n * z * K} \quad (4.5)$$

где  $T$  – трудоемкость этапа, норма-час;

$q_n$  – количество исполнителей по этапу;

$z$  – продолжительность рабочего дня,  $z = 8$  часов;

$K$  – коэффициент выполнения норм времени,  $K = 1,1$ .

Полученную величину  $t_n$  округляем в большую сторону до целых дней.

$$t_1 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} - \text{Разработчик, постановка задачи};$$

$$t_2 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Дизайнер, изучение литературы};$$

$$t_3 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Разработчик, изучение литературы};$$

$$t_4 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Разработчик, выбор среды разработки};$$

$$t_5 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Разработчик, изучение «разделов упражнений»};$$

$$t_6 = \frac{44}{1 \cdot 8 \cdot 1,1} \approx 5 \text{ дней} \text{ Дизайнер, создание всех необходимых}$$

графических элементов;

$$t_7 = \frac{132}{1 \cdot 8 \cdot 1,1} \approx 15 \text{ дней} \text{ Разработчик, разработка базы знаний}$$

приложения;

$$t_8 = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1 \text{ дней} \text{ Разработчик, тестирование приложения};$$

$$t_9 = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1 \text{ дней} \text{ Разработчик, отладка приложения};$$

$$t_{10} = \frac{24}{1 \cdot 8 \cdot 1,1} \approx 3 \text{ дня} \text{ Разработчик; ввод ПП в эксплуатацию.}$$

Сводные результаты расчета затрат на основную заработную плату работников, задействованных в разработке ПП представлены в таблице 4.5

Таблица 4.5 - Сводные результаты расчета затрат на основную заработную плату

Наименование работ	Исполнитель	Трудоемкость		Длительность цикла, дни	Заработная плата за час работы, тенге	Сумма заработной платы
		Нормы-час	% от общей трудоемкости			
Постановка задачи	Разработчик	16	5,6	2	446	7136
Изучение литератур	Дизайнер	16	5,6	2	298	4768
Изучение литератур	Разработчик	16	5,6	2	446	7136
Выбор среды	Разработчик	8	2,8	2	446	3568
Изучение «требовании»	Разработчик	16	5,6	2	446	7136
Разработка интерфейс	Дизайнер	44	15,3	5	298	13110
Разработка базы знания	Разработчик	132	45,8	15	446	58870
Тестирование	Разработчик	8	2,8	1	446	3568
Отладка приложения	Разработчик	8	2,8	1	446	3568
Оформление и сдача	Разработчик	24	7,5	3	446	10700
Итого:		288	100	35		119600

Таким образом, по формуле рассчитанная основная заработная плата составляет 119600 тенге.

Дополнительная заработная плата рассчитывается по формуле:

$$З_{\text{доп}} = З_{\text{осн}} * 0,1 \quad (4.6)$$

и составит

$$З_{\text{доп}} = 119600 * 0,1 = 11960 \text{ тенге}$$

Фонд оплаты труда, согласно произведенным расчетам и формуле 4.2 составит:

$$\text{ФОТ} = 119600 + 11960 = 131600 \text{ тенге.}$$

#### 4.5.1 Расчет затрат по социальному налогу

Значение социального налога 11% от дохода работника и определяется по формуле:

$$С_{\text{н}} = (\text{ФОТ} - \text{ПО}) * 0,11 \quad (4.7)$$

где ПО – это пенсионные отчисления, составляющие 10% от ФОТ и не облагаются социальным налогом и определяются по формуле:

$$\begin{aligned} \text{ПО} &= \text{ФОТ} * 0,1, \\ \text{ПО} &= 131600 * 0,1 = 13160 \text{ тенге.} \end{aligned} \quad (4.8)$$

И так, социальный налог будет иметь значение:

$$С_{\text{н}} = (131600 - 13160) * 0,11 = 13030 \text{ тенге.}$$

#### 4.5.2 Расчет амортизационных отчислений

По данной формуле рассчитываются амортизационные отчисления:

$$A_j = \frac{H_A * C_{\text{ПЕР}} * N}{100 * 12 * n} \quad (4.9)$$

где  $H_A$  – норма амортизации;

$C_{\text{ПЕР}}$  – первоначальная стоимость оборудования;

$N$  – количество дней на выполнение работ;

$n$  – количество рабочих дней в одном месяце.

Таким образом, амортизационные отчисления по используемому техническому оборудованию и ПП, по формуле 4.9 будут иметь значение:

На оборудование:



$$A_1 = \frac{40 * 544990 * 35}{100 * 12 * 21} = 30280 \text{ тенге,}$$

$$A_2 = \frac{40 * 18394 * 35}{100 * 12 * 21} = 1022 \text{ тенге.}$$

Суммарные затраты на амортизацию рассчитываются по формуле:

$$A = A_1 + A_2. \quad (4.10)$$

Амортизационные отчисления составляют:

$$A = 30280 + 1022 = 31302 \text{ тенге.}$$

### 4.5.3 Расчет затрат на электроэнергию

В ходе разработки ПП используются электрооборудования, поэтому необходимо рассчитать затраты на пользование электроэнергией. Затраты на электроэнергию для производственных нужд включают в себя расходы на электроэнергию для оборудования и на дополнительные нужды, определяются по формуле 4.11.

$$\mathcal{E} = \mathcal{Z}_{\text{эл.эн.обор}} + \mathcal{Z}_{\text{доп.нуж}}, \quad (4.11)$$

где  $\mathcal{Z}_{\text{эл.эн.обор}}$  – затраты на электроэнергию оборудования;

$\mathcal{Z}_{\text{доп.нуж}}$  – затраты электроэнергии на дополнительные нужды.

Расходуемая электроэнергия на оборудование рассчитывается по формуле:

$$\mathcal{Z}_{\text{эл.эн.обор}} = W * T * S * K_{\text{исп}}, \quad (4.12)$$

где  $W$  – потребляемая мощность, Вт;

$T$  – количество часов работы оборудования;

$S$  – стоимость киловатт-часа электроэнергии (1кВтч=21 тнг);

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

Потребляемая мощность Acer Aspire v3-772g составляет 120 Вт.

Потребляемую мощность адаптеров для зарядки мобильного устройства не учитываем, т.к. они используются исключительно для тестирования и отладки приложения, при этом постоянно подключены к ноутбуку Acer Aspire v3-772g.

Время рассчитывается на основе количества рабочих дней в месяц и рабочих часов в день.

Сумма расходов на электроэнергия для ноутбуков составляет:

$$\mathcal{Z}_{\text{эл.эн.обор}} = (0,12 + 0,12) * (35 * 8) * 21 * 0,9 = 1270 \text{ тенге}$$

Расходы на дополнительные нужды учитываются по укрупненному показателю в размере 5% от затрат на техническое оборудование:

$$\begin{aligned} Z_{\text{доп.нуж}} &= 0,05 * Z_{\text{эл.эн.обо}}, \\ Z_{\text{доп.нуж}} &= 0,05 * 1270 = 63,5 \text{ тенге.} \end{aligned} \quad (4.13)$$

Суммарные расходы на электроэнергию составляют:

$$\text{Э} = 1270 + 63,5 = 1333,5 \text{ тенге.}$$

#### 4.5.4 Расчет прочих расходов

Так как в процессе разработки проекта нужен офис и доступ разработчиков в интернет, рассчитаем прочие расходы. Прочие расходы на разработку приложения, рассчитываются по формуле:

$$\text{Спр} = \text{Зар.кв} + \text{Зинт} \quad (4.14)$$

где Зар.кв – общая сумма затрат на аренду помещения;

Зинт – общая сумма затрат за интернет.

Месячная стоимость аренды в городе за 1 м<sup>2</sup> составило 2093 тенге, что соответствует средним показателем в городе. Площадь офиса составило 10 м<sup>2</sup>.

Затраты на аренду офиса рассчитывается по формуле:

$$\text{Зар.кв} = \text{Спл} * \text{Сцена.за.кв.м} * \text{Nмес} \quad (4.15)$$

где Спл – площадь арендуемого помещения;

Сцена.за.кв.м – стоимость аренды 1 м<sup>2</sup> в месяц;

Nмес – количество месяцев.

Так как проект выполнялся 35 дней, рабочее помещение бралось на 2 месяца. Затраты на аренду офиса составляют:

$$\text{Зар.кв} = 10 * 2093 * 2 = 41860 \text{ тенге.}$$

Размер ежемесячной платы по тарифному плану «ID Net Turbo» от провайдера «Казахтелеком» составила – 3600 тенге. Затраты на интернет рассчитываем по формуле:

$$\text{Зинт} = \text{Синт.мес} * \text{Nмес} \quad (4.16)$$

где Синт.мес – размер ежемесячной платы.

Суммарные затраты на интернет составляют:

$$\text{Зинт} = 3600 * 2 = 7200 \text{ тенге.}$$

Прочие расходы составляют:

$$C_{\text{пр}} = 41860 + 7200 = 49060 \text{ тенге.}$$

#### 4.5.5 Расчет накладных расходов

Накладные расходы рассчитываются как 30% от всех затрат и определяются по формуле:

$$H_p = (\text{ФОТ} + C_n + A + \text{Э} + C_{\text{пр}}) * 0,3 = (131600 + 13160 + 31302 + 1333,5 + 49060) * 0,3 = 67940 \text{ тенге} \quad (4.17) H_p$$

Таким образом, себестоимость разработки ПП, согласно формуле 4.1 составила:

$$C = 131600 + 13160 + 31302 + 1333,5 + 49060 + 67940 = 294\,400 \text{ тенге.}$$

Результаты расчета стоимости разработки базы знаний для мобильного приложения «ЗОЖ» на платформе Android представлены в таблице 4.6 и на рисунке 4.1.

Таблица 4.6 – Результирующая таблица себестоимости

Наименование статей затрат	Сумма, тенге	В процентах от общей суммы, %
ФОТ	131600	45
Затраты по социальному налогу	13160	4
Амортизационные отчисления	31302	11
Затраты на электроэнергию	1333,5	1
Накладные расходы	67940	22
Прочие расходы	49060	17
Итого	294400	



Рисунок 4.1 - Структура затрат на разработку приложения

#### 4.5.6 Цена реализации проекта

Цена данного проекта складывается из себестоимости и желаемого чистого дохода. Минимальная стоимость данного программного продукта рассчитывается по формуле:

$$Ц_{п} = C + П \quad (4.18)$$

где C – себестоимость разработки приложения;

П – чистый доход от программного продукта.

Для определения стартовой цены используется желаемый уровень рентабельности. Для данной отрасли он составляет 25%:

$$Ц_{п} = C * \left(1 + \frac{P}{100}\right) \quad (4.19)$$

где P - рентабельность.

Тогда по формуле 4.19:

$$Ц_{п} = 294400 * \left(1 + \frac{25}{100}\right) = 368000 \text{ тенге.}$$

Далее определяется цена реализации с учётом налога на добавленную стоимость (НДС), формула (4.17):

$$Ц_{р} = Ц_{п} + НДС \quad (4.20)$$

где НДС – налог на добавленную стоимость.

Ставка НДС на 2016 год, согласно пункту 1 статье 268 «Ставки налога

на добавленную стоимость» Налогового кодекса РК составляет 12 %.

В итоге получаем цену реализации проекта равной

$$Ц_p = 368000 + (368000 * 0,12) = 412200 \text{ тенге.}$$

Стоимость разработки базы знаний для мобильного приложения «Здоровый образ жизни» для операционной системы Android составила 412200 тенге, в которую заложены все возможные затраты при разработки программного продукта.

По анализу расходов на реализацию проекта основной статьей расходов является затраты на оплату труда, более 45%.

Стоимость разработки мобильного приложения включает больших интеллектуальных и финансовых затрат. Расчеты показали, что производство и реализация данного программного продукта можно считать экономически выгодным.

Проблема здорового образа жизни одна из самых актуальных в наши дни. Здоровый образ жизни предусматривает потребление достаточного количества белков, жиров, углеводов, витаминов и обеспечение организма достаточной физической нагрузкой для нормального функционирования организма в целом.

Приложение «Здоровый образ жизни» позволит следить за режимом дня, что позволит предотвратить множество болезней.

## 5 Безопасность жизнедеятельности

### 5.1 Анализ условий труда

Тема дипломной работы - «Разработка базы знания для мобильного приложения «ЗОЖ»». Приложение, разработанное в ходе выполнения дипломной работы, поможет пользователям в соблюдении здорового образа жизни.

Цель работы – разработка базы знаний, позволяющей осуществить оценку состояния здоровья пользователя по введенным данным и выводить индивидуальную программу тренировок.

Разработка базы знаний для мобильного приложения напрямую связана с работой на компьютере. В данной главе описаны основные вредные факторы при работе с компьютером.

В соответствии с санитарно-эпидемиологическими требованиями к условиям работы с источниками физических факторов, которые оказывают влияние на здоровье человека от 01.12.2011, оптимальными микроклиматическими условиями в легкой 1б категории работ являются следующие цифры:

- температура окружения 23 - 21°С - в холодный сезон, 22 - 24°С - в теплый;
- относительная влажность воздуха 40 - 60% в холоде и в теплое время года;
- скорость воздуха 0,2 м/с - в теплый сезон, 0,1 м/с - в холодный.

Вышеуказанные установленные нормы производственного микроклимата соответствуют климатическим условиям на рабочем месте.

При работе с компьютером, сотрудник может столкнуться такими опасными и вредными производственными факторами как: электромагнитное излучение, статическое электричество, поражение электрическим током, отсутствие освещенности рабочего пространства, снижение ионизации воздуха и др. Эти вредные производственные факторы могут влиять на здоровье сотрудника, который пользуется данной компьютерной техникой.

Компьютерная техника является потенциальным источником поражения электрическим током и возникновению пожара. Мерами обеспечения пожарной безопасности при работе с персональным компьютером является: правильное размещение оборудования и электрических кабелей. В виде простых мер необходимо использовать скрытую проводку, безопасные гнезда, рассчитывать нагрузку на сеть и соблюдать правила пожарной безопасности. Не стоит часто извлекать вилки из электрических розеток, чтобы избежать искривление.

Компьютеры при работе создают электростатические поля в зоне действия, которых попадают в различные предметы. Во время работы системы вентиляторов, гоняющих воздух в системных блоках, выдувают наружу электризованные пыли, оседающих не только на нашей коже, но и в

дыхательных путях. Поверхность монитора тоже накапливает статического электричества.

На сегодняшний день малоизученной областью является влияние статического электричества на организм человека. Большинство исследователей считают, что под воздействием статического электричества на коже человека, раздражаются нервные окончания, а также изменяется ионный состав тканей. Все эти изменения приводят к усталости, раздражительности и к плохому сну.

Меры защиты от статического электричества должна включать в себя: влажную уборку, дополнительное увлажнение воздуха в рабочем пространстве (влажность воздуха должна быть больше 50%), вентиляция помещений, заземление оборудования.

В дополнение к электрической опасности существует так же электромагнитное воздействие на пользователя компьютера. В качестве меры предосторожности против электромагнитных излучений, можно взять частые прогулки на свежем воздухе, вентилирующую помещения, активную физическую деятельность, соблюдение правил работы с ПК, выбор и использование хорошей аппаратурой, соответствующим стандартам безопасности и санитарным нормам.

Уровень шума компьютера зависит от его производительности. Шум имеет тенденции вызывать звуковые раздражения. Человек, подвергающийся частому акустическому воздействию, быстро становится уставшим, становится раздражительным, забывчивым. Источниками шума в компьютере может быть системный блок, устройства воздушного охлаждения, процессор. Нужно делать перерывы. При работе за компьютером освещение имеет огромное значение. Свет не должен быть ярким, лучшим вариантом при работе за современным компьютером является рассеянный приглушенный свет. Расположение компьютера относительно окна очень важно. Компьютер должен быть расположен таким образом, чтобы свет из окна не попал непосредственно на пользователя. Так как это может привести к усталости в глазах. Чтобы избежать этого, нужно приобрести жалюзи или плотные шторы, которые защищают от прямых солнечных лучей.

В дополнение к указанным выше вредным и опасным факторам, влияющих на сотрудника во время работы, необходимо отметить и другие опасные факторы, спровоцированных неправильной организацией работы на компьютере. Организации рабочего места нужно уделять максимум внимания, поскольку она влияет на здоровье сотрудника. Сама по себе сидячая работа является вредным для любого человека. Длительное пребывание в одном положении заставляет мышцы работать без остановки. Не активный образ жизни является главной проблемой пользователей ПК и программистов. За счет снижения физической активности, вызванной продолжительным сидением, растет риск заболевания как ожирение, геморрой, остеоартрит. При работе за компьютером, человек сидит в неправильном положении, сутулится или поддается вперед, его позвоночник деформируется и травмируются диски.

Вышеуказанные психофизиологические опасные и вредные факторы по характеру воздействия подразделяются на:

- физические, это статические и динамические;
- нервно-психиатрические, это психические перенапряжения, перенапряжение анализаторов, монотонность труда и эмоциональные перегрузки.

Пользователи и разработчик информационной системы наиболее подвержены опасности статических вредных (неизменное положение тела), перегрузку анализаторов.

Необходимо организовать надлежащее рабочее место, постоянно контролировать осанку и делать перерывы для отдыха и физических упражнений. Учитывая основные раскрытые вредные факторы, следует подчеркнуть требования по организации работы за компьютером:

- в рабочем офисе должны присутствовать естественное и искусственное освещение;
- помещение должно быть оборудовано системами кондиционирования воздуха или эффективной вентиляцией; помещение должно проветриваться ежедневно;
- в помещениях нужно проводить ежедневную влажную уборку;
- желательно закрыть окна шторами или жалюзи, чтобы избежать прямых попаданий лучей солнца;
- искусственное освещение должно быть равномерным и общим;
- на рабочем пространстве должны легко помещаться монитор, клавиатура, мышь, а также документы, книги, бумаги;
- стул должен регулироваться по высоте и углам наклона сиденья и спинки;
- монитора должен находиться на расстоянии не менее 55 – 60 см от глаз, и он должен быть установлен таким образом, чтобы не отражать посторонний свет. Необходимо выбрать правильную яркость экрана;
- рекомендуется по крайней мере, 2 раза в день выполнять упражнения для глаз;
- правильное положение рук во время работе с клавиатурой и мышью: локти лежат параллельно поверхности стола и под прямым углом к плечу;
- нужно постоянно контролировать положение тела в процессе, то есть за осанкой; время нахождения непосредственно перед компьютером не должно превышать шести часов в день, во время рабочего дня рекомендуется брать перерывы длительностью 10-20 минут, и выполнять комплексы физических упражнений для глаз;

Рассмотрим помещение в котором ведется разработка сайта (рисунок 5.2.1). Параметры рабочего помещения: длина - 4 м, ширина - 2,5 м, высота - 2,6 м. Площадь помещения составляет 10 м<sup>2</sup>, что дает нужную площадь на 1 человека. Офис расположен в здании на 4-м этаже. Согласно п.5 «Санитарно-эпидемиологические требования к размещению и эксплуатации ПК, ПлПК, ноутбуков и ВТ» РК от 25.04.2011 площадь на одно рабочее место пользователей ПК на базе плоских дискретных экранов (жидкокристаллические, плазменные) при любом положении должна составлять не менее 6 квадратных метров.



Рабочее помещение оснащено одним рабочим местом. Рабочее помещение находится в здании, вдали от железнодорожных дорог, крупных дорог, аэропорта и других источников шума, которые могут повлиять на процесс работы.

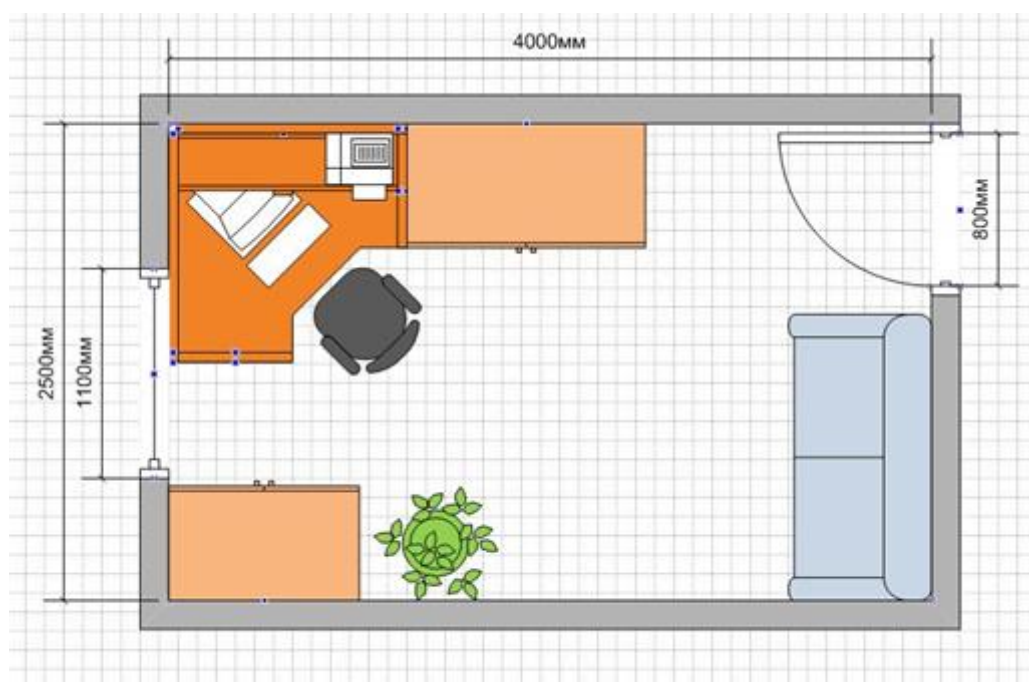


Рисунок 5.2.1 - План рабочего помещения

## 5.2 Расчет естественного освещения

Естественное освещение характеризуется тем, что уровень естественного освещения может резко измениться в течение очень короткого промежутка времени и в довольно в большом диапазоне. Таким образом, главной ценностью для оценки естественного освещения в помещении принят коэффициент естественной освещенности (КЕО).

Для V разряда зрительных работ (наименьший или эквивалентный размер объекта отличия от 1.0 до 5.0) нормированное значение КЕО,  $e_n$  для зданий, расположенных в разных районах должны определяться по формуле:

$$e_n = e_n \cdot m, \quad (5.5)$$

где  $e_n$  - значение КЕО по таблице 5.5.1;

$m$  - коэффициент светового климата, определяется по таблице 5.5.2 для ориентации световых проёмов С.

Таблица 5.5.1 - Нормы освещенности при естественном освещении

Характеристика зрительной работы	Эквивалент	Разряд зрительной работы	Естественное освещение	
			КЕО, $e_n$ , %	
			при верх. или комбин. освещ.	при бок. освещении
Мал. точности	Свыше 1.0 до 5.0	V		1,0

Таблица 5.5.2 - Значения коэффициента  $m$

Световые проемы	Ориентация световых проемов по сторонам горизонта	Коэффициент светового климата, $m$
		Номер группы административных районов
В наружных стенах Зданий	С	0,75
	СВ, СЗ	0,7
	З, В	0,65
	ЮВ, ЮЗ	0,65
	Ю	0,65

В нашем случае значение КЕО  $e_n=1,0$  и коэффициент светового климата  $m = 0,65$ .

Значение КЕО для этого помещения:

$$e_n = 1 \cdot 0,65 = 0,65\%$$

Расчитаем, достаточно ли естественного освещения для работы в офисе.

Значение КЕО данного помещения сравним с нормированным. Размеры офиса составляют 2,5x4x2,6 соответственно длина, ширина, высота. Окно находится на высоте 0,8 м от пола. Высота окна в точности 1,4 м.

Площадь окна расчитывается по формуле:

$$S_0 = \frac{S_n \times e_n \times \eta_0 \times K_{зд} \times K_3}{100 \times \tau_0 \times r_1}$$

Выразим  $e_n$  из формулы (5.6):

$$e_n = \frac{S_0 \times 100 \times \tau_0 \times r_1}{\eta_0 \times K_{зд} \times K_3 \times S_n}$$

где  $S_n$  - площадь помещения,  $S_n = 2,5 \cdot 4 = 10 \text{ м}^2$ ;

$S_0$  - площадь окна помещения (размеры окна 1,1x1,4),  $S_0 = 1,1 \cdot 1,4 = 1,54 \text{ м}^2$ ;

$K_3$  - коэффициент запаса,  $K_3 = 1,2$  (помещения общественных и жилых зданий);

$\eta_0$  - световая характеристика окон;

$\tau_0$  - общий коэффициент светопропускания;

$$\tau_0 = \tau_1 \times \tau_2 \times \tau_3 \times \tau_4 \quad (5.8)$$

конструкциях: железобетонные формы  $\tau_3 = 0,8$ ;

$\tau_4$  - коэффициент, учитывающий потери света в солнцезащитных устройствах: убирающиеся регулируемые жалюзи  $\tau_4 = 1,2$ . Тогда:

$$\tau_0 = 0,8 \cdot 0,6 \cdot 0,8 \cdot 1,2 = 0,5184.$$

Отношение длины помещения к его глубине:

$$\frac{2,5}{4/2} = 1,25.$$

Отношение ширины помещения к высоте от уровня рабочей поверхности до верха окна:

$$\frac{4}{1,4} = 2,86.$$

Отсюда  $\eta_0 = 15$ ;

$r_1$  - коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхностей помещения и подстилающего слоя, прилегающего к зданию:

$$\frac{B}{h_1} = \frac{4}{1,4} = 2,86;$$

$$\frac{H}{B} = \frac{2,6}{4} = 0,65;$$

$$\frac{L}{B} = \frac{2,5}{4} = 0,625;$$

$$\frac{P_{\text{пот}} + P_{\text{ст}} + P_{\text{пол}}}{3} = \frac{70 + 50 + 30}{3} = 50\%;$$

Отсюда  $r_1 = 1,75$ .

$$\frac{P}{H_{\text{зд}}} = \frac{20}{4,2} = 4,76.$$

$K_{\text{зд}}$  - коэффициент, учитывающий затенение окон противостоящими зданиями смотрим по таблице 5.5.3.

Таблица 5.5.3 - Значения коэффициента  $K_{\text{зд}}$

$P/H_{\text{зд}}$	0,5	1	1,5	2	3 и более
$K_{\text{зд}}$	1,7	1,4	1,2	1,1	1

Отсюда видим, что  $K_{\text{зд}} = 1$ .

Подставим все значения в расчетную формулу (5.7):

$$e_n = \frac{1,54 \times 100 \times 0,5184 \times 1,75}{15 \times 1 \times 1,2 \times 10} = 0,78\%.$$

Нормируемое КЕО равно  $e_n = 0,65\%$ , а фактическое значение КЕО  $e_n = 0,78\%$ . Это означает, что помещение соответствует нормативам освещения рабочего помещения.

### 5.3 Расчет освещения

Повышение освещенности следует предусматривать также в помещениях с недостаточным по нормам естественным светом, который при боковом освещении составляет менее 80% нормируемого значения, а при верхнем менее 60%. При этом освещенность должна быть 200 Лк при лампах накаливания малой точности.

В таблице 5.4.1 приведены нормы освещенности при искусственном освещении.

Таблица 5.4.1 - Нормы освещенности при искусственном освещении

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм.	Разряд Зрительной работы	Освещенность, лк
			Искусственное освещение
Малой точности	От 1.0 до 5.0	V	200

Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Мы же будем использовать лампы накаливания.

Расчет освещения производится для комнаты, длина которой 2,5 м, ширина - 4 м.

Найдем значение освещенности данного помещения и сравним его с нормированным значением, для того, чтобы определить достаточность текущего искусственного освещения для работы в помещении.

Вычислим высоту подвеса светильника над рабочей поверхностью

$$H = h - h_p - h_c, \quad (5.1)$$

где  $h_c$  - расстояние от светильника до перекрытия,  $h_c = 0,1$  м;

$h_p$  - высота рабочей поверхности над полом,  $h_p = 0,8$  м;

$h$  - высота помещения,  $h = 2,6$  м.

$$H = 2,6 - 0,8 - 0,1 = 1,7 \text{ м.}$$

Наиболее выгодное расстояние между светильниками определяется как:

$$L = \lambda \cdot H, \quad (5.2)$$

где  $\lambda$  - относительные светотехнические и энергетические наивыгоднейшие расстояния между светильниками ( $\lambda = 1,2 \div 1,4$ ).

$$L = 1,2 \cdot 1,7 = 2,04 \text{ м.}$$

Определяем индекс помещения:

$$i = \frac{S}{H \times (A + B)}$$

где  $S$  - площадь помещения,  $S = 10 \text{ м}^2$ ;

$H$  - расчетная высота подвеса,  $H = 1,7$  м;

$A$  - ширина помещения,  $A = 4$  м;

$B$  - длина помещения,  $B = 2,5$  м.

Подставив значения получим:

$$i = \frac{10}{1,7 \times (4 + 2,5)} = 0,9.$$

Понятие коэффициента отражения поверхности используется при планировании и расчете освещения. От него зависят характеристики осветительных приборов, которые будут использоваться для обеспечения заданной степени освещенности помещения.

Чем выше коэффициент отражения потолка, стен и пола в комнате, тем меньшей мощностью люстры можно обойтись.

В таблице 5.4.2 приведены коэффициенты отражения различных поверхностей.

Таблица 5.4.2 - Коэффициенты отражения

Характеристика поверхности	Коэффициент отражения, %
Белая поверхность (побелка, белые обои, белая плитка)	70
Светлая поверхность (светлые обои, светлый потолок, светлая краска)	50
Поверхность средней светлости (светлый паркет, светлый линолеум, несветлая краска, несветлые обои)	30
Темная поверхность (темный паркет, темная краска, темный линолеум, темные обои)	10

В данном случае коэффициенты отражения потолка, стен и пола равны:  
 $R_{\text{пот}} = 70\%$ ;  $R_{\text{ст}} = 50\%$ ;  $R_{\text{пол}} = 30\%$ .

Зная индекс помещения  $i$ , находим коэффициент использования, используя таблицу 5.4.3.

Таблица 5.4.3 - Значение коэффициента использования для светильников

$i$	$R_{\text{пот}}, \% 70$ $R_{\text{ст}}, \% 50$ $R_{\text{пол}}, \% 30$	$R_{\text{пот}}, \% 50$ $R_{\text{ст}}, \% 30$ $R_{\text{пол}}, \% 10$	$R_{\text{пот}}, \% 30$ $R_{\text{ст}}, \% 10$ $R_{\text{пол}}, \% 10$
0,5	28	21	18
1,0	49	40	36
3,0	73	61	58
5,0	80	67	65

В нашем случае  $= 49\%$ ;

Также, в нашем случае  $K_3=1,2$ .

Формула для расчета освещенности:

$$E = \frac{N \times \Phi_{\text{л}} \times \eta \times n}{S \times K_3 \times z}$$

где  $E$  - требуемая освещенность горизонтальной плоскости, лк;  $N$  - количество светильников;

$\Phi_{\text{л}}$  - световой поток одной лампы, лм;  $\eta$  - коэффициент использования осветительной установки;

$n$  - число ламп в одном светильнике;

$S$  - площадь помещения,  $\text{м}^2$ ;

$K_3$  - коэффициент запаса;

$z$  - коэффициент неравномерности освещения (отношение средней освещенности к минимальной,  $z = 1,1$ ).

Расчетные значения светового потока рассмотрены в таблице 5.4.5.

Таблица 5.4.5 - Расчетные значения светового потока наиболее распространенных источников света  $\Phi_{л}$

Тип лампы(мощность)	$\Phi_{л}$ , лм	Тип лампы	$\Phi_{л}$ , лм
200	3100	25	230
150	2200	15	90
100	1360		
75	940		
60	720		
40	420		

Для освещения выбираем лампы накаливания, световой поток которых  $\Phi_{л} = 1360$  Лк.

Подставим все значения в формулу (5.4) и получим:

$$E = \frac{4 \times 1360 \times 0,49 \times 1}{10 \times 1,2 \times 1,1} = 201,94 \text{ Лк.}$$

Нормируемая освещенность составляет 200 Лк, а освещенность в помещении 201,94 Лк. Это означает, что искусственного освещения в помещении достаточно.

В данном разделе был произведен анализ условий труда в данном помещении, также, расчет естественного и искусственного освещения.

Расчет показал, что для естественного освещения достаточно одного окна площадью 1,54 м<sup>2</sup>.

Искусственного освещения на рабочем месте было достаточно. Таким образом, система искусственного освещения включает в себя 4 светильника по 1 лампе со световым потоком излучения 1360 Лм каждая, поэтому в этом помещении можно работать и в темное время суток. Была описана организация безопасности жизнедеятельности в процессе проектирования, разработки и реализации дипломной работы. В этом разделе был произведен анализ условий труда в рабочем помещении, в частности, расчет естественного и освещения.

## Заключение

Разработка базы знаний для мобильного приложения - интересный и сложный процесс, выполнение которой может быть более чем оправдано. Она требует логического мышления и структурированного подхода. Она включает в себя различные задачи, непосредственно не связанных с изучением языков кода. Это всегда новые знания, которые могут быть приобретены, новые требования, которые требуют исследований и новые технологии, с которыми стоит экспериментировать. Наградой является своего рода запуск окончательной версии приложения и осознание собственного вклада в достижение данной цели. Программное обеспечение определяет форму современного мира, а также тот, кто работает над его созданием, может внести свой вклад в образ будущего.

Во время выполнения этого дипломного проекта были изучены основные принципы разработки мобильного приложения, а также требований, предъявляемых к нему. Мы изучили принципы построения и эксплуатации программного продукта для операционной системы Android. Кроме того, было также установлено, что изучаемая тема является в наивысшей степени актуальной в реалиях современного рынка мобильных устройств.

Была, также, рассчитана общая стоимость продукта, на основании суммирования различных видов расходов, возможные при разработке. Этот расчет соответствует любому виду реализации продукта, как разработка для заказчика, и как самостоятельная разработка, для последующей публикации в интернет магазинах мобильного приложения. Изучение подобных продуктов на рынке, показало, что стоимость разработки полностью соответствует качеству разрабатываемого ПП и позволяет получить большое количество прибыли.

В разделе «Безопасность жизнедеятельности» были рассчитаны необходимые условия для организации соответствующего всем установленным нормам, рабочего места для сотрудников, участвующих в создании проекта. Рассчитано требуемое искусственное освещение, необходимое для комфортной работы без вреда для здоровья сотрудников.



## Список используемой литературы

- 1 Гаврилова – Базы знаний интеллектуальных систем: Пособие. 2003. – С 234.
- 2 Лисп. Русскоязычное общество программистов [lisp.ru](http://lisp.ru)
- 3 Ахметова М. -Системы искусственного интеллекта: Конспект лекций. 2013. – 46 с.
- 4 Fukuoka, Y., Kamitani, E., Bonnet, K., & Lindgren, T. (2011). Real-time social support through a mobile virtual community to improve healthy behavior in overweight and sedentary adults: A focus group analysis. *Journal of Medical Internet Research*.
- 5 Fukuoka, Y., Lindgren, T., & Jong, S., 2012. Qualitative Exploration of the acceptability of a Mobile Phone and Pedometer-Based Physical Activity Program in a Diverse Sample of Sedentary Women. *Public Health Nursing*, – 232-240 с.
- 6 Kreps, G., & Neuhauser, L., 2010. New directions in eHealth communication: opportunities and challenges. *Patient Education & Counseling*, – 329-336 стр.
- 7 Нильсон Н. - Искусственный интеллект. Методы поиска решений, 1973 – 202с
- 8 .Заенцев И. В. Нейронные сети: основные модели И. В. Заенцев. — Воронеж: Изд-во Воронежского госуд. ун-та, 1999. — 76 с.
- 9 Хорошевский В, Базы знаний интеллектуальных систем. Учебник. — СПб.: Питер, 2000, - 225 с.
- 10 Джонс М. Т. Программирование искусственного интеллекта в приложениях М. Тим Джонс; Пер. с англ. Осипов А. И. — М.: ДМК Пресс, 2006. — 312 с.
- 11 Варакин, М.В. Разработка мобильных приложений под Android: учебное пособие / М.В. Варакин; Центр компьютерного обучение "Специалист" при МГТУ им. Н.Э. Баумана – Москва, 2012 – 128с
- 12 *Android Application Development / Rick Rogers, John Lombardo, Zigurd Mednieks, and Blake Meike.*– 399 с
- 13 Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. З.Д. Еркешева, Г.Ш. Боканова Алматы: АУЭС, 2013 – 40 с.
- 14 ГОСТ 12.1.005-88 ОБЩИЕ САНИТАРНО-ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ВОЗДУХУ РАБОЧЕЙ ЗОНЫ
- 15 ГОСТ 30494-96 ПАРАМЕТРЫ МИКРОКЛИМАТА В ПОМЕЩЕНИЯХ
- 16 [habrahabr.ru](http://habrahabr.ru)
- 17 [fandroid.info](http://fandroid.info)
- 18 [developer.android.com](http://developer.android.com)

## Приложение А

### Листинг программной части

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cse280.HealthyLifeStyle" >

    <!-- To auto-complete the email text field in the login form with the user's emails -->
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.READ_PROFILE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity
            android:name="cse280.HealthyLifeStyle.LoginActivity"
            android:label="@string/title_activity_login"
            android:theme="@style/AppTheme.NoActionBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Dash"
            android:label="@string/title_activity_dash"
            android:parentActivityName="cse280.HealthyLifeStyle.LoginActivity"
            android:theme="@style/AppTheme.NoActionBar" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="cse280.HealthyLifeStyle.LoginActivity" /><!-- ATTENTION: This
intent was auto-generated. Follow instructions at
https://g.co/AppIndexing/AndroidStudio to publish your URLs. -->
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />

                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
                <!-- ATTENTION: This data URL was auto-generated. We recommend that you use the
HTTP scheme.
                TODO: Change the host or pathPrefix as necessary. -->
            <data
                android:host="HealthyLifeStyle.cse280"
                android:pathPrefix="/dash"
                android:scheme="http" />
```

```

    </intent-filter>
</activity>
<activity
    android:name=".SettingsActivity"
    android:label="@string/title_activity_settings"
    android:parentActivityName=".Dash" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.Dash" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.CalorieInput"
    android:label="@string/title_activity_calorie_input"
    android:parentActivityName="cse280.HealthyLifeStyle.DashBoard"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.DashBoard" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.DashBoard"
    android:label="@string/title_activity_dash_board"
    android:parentActivityName="cse280.HealthyLifeStyle.LoginActivity"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.LoginActivity" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.Stats"
    android:label="@string/title_activity_stats"
    android:parentActivityName="cse280.HealthyLifeStyle.DashBoard"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.DashBoard" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.Workout"
    android:label="@string/title_activity_workout"
    android:parentActivityName="cse280.HealthyLifeStyle.DashBoard"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.DashBoard" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.History"
    android:label="@string/title_activity_history"
    android:parentActivityName="cse280.HealthyLifeStyle.DashBoard"
    android:theme="@style/AppTheme.NoActionBar" >

```

```

    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.DashBoard" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.CreateAccount"
    android:label="@string/title_activity_create_account"
    android:parentActivityName="cse280.HealthyLifeStyle.LoginActivity"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.LoginActivity" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.Cardio"
    android:label="@string/title_activity_cardio"
    android:parentActivityName="cse280.HealthyLifeStyle.Workout"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.Workout" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.BodyWeightExercise"
    android:label="@string/title_activity_body_weight_exercise"
    android:parentActivityName="cse280.HealthyLifeStyle.Workout"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.Workout" />
</activity>
<activity
    android:name="cse280.HealthyLifeStyle.Weights"
    android:label="@string/title_activity_weights"
    android:parentActivityName="cse280.HealthyLifeStyle.Workout"
    android:theme="@style/AppTheme.NoActionBar" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="cse280.HealthyLifeStyle.Workout" />
</activity>
</application>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"

```

```
tools:showIn="@layout/activity_body_weight_exercise"  
tools:context="cse280.HealthyLifeStyle.Cardio"  
android:orientation="vertical">
```

```
<Spinner  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/spinner"  
    android:layout_weight="0.00"  
    android:spinnerMode="dropdown" />
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceLarge"  
    android:text="Повторения:"  
    android:id="@+id/textView31"  
    android:layout_gravity="center_horizontal"  
    android:layout_weight="0.00" />
```

```
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:inputType="number"  
    android:ems="10"  
    android:id="@+id/edit_time"  
    android:layout_gravity="center_horizontal"  
    android:layout_weight="0.00"  
    android:hint="Кол-во повторений" />
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceLarge"  
    android:text="Подходы:"  
    android:id="@+id/textView19"  
    android:layout_gravity="center_horizontal"  
    android:layout_weight="0.00" />
```

```
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:inputType="number"  
    android:ems="10"  
    android:id="@+id/edit_distance"  
    android:layout_gravity="center_horizontal"  
    android:layout_weight="0.00"  
    android:hint="Кол-во Подходов" />
```

```
<TextView  
    android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="@string/calorie_burn"
android:id="@+id/textView32"
android:layout_gravity="center_horizontal"
android:layout_weight="0.00" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:id="@+id/calories_burned_display"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.00" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ввод"
    android:id="@+id/button5"
    android:layout_weight="0.00"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:layout_gravity="center_horizontal"
    android:onClick="calcBurned" />
```

```
<Button
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Назад К упражнениям"
    android:id="@+id/button11"
    android:layout_weight="0.00"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:layout_gravity="center_horizontal"
    android:onClick="back" />
```

```
</LinearLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

```

xmlns:app="http://schemas.android.com/apk/res-auto" android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:showIn="@layout/activity_calorie_input"
tools:context="cse280.HealthyLifeStyle.CalorieInput"
android:orientation="vertical"
android:weightSum="1"
android:background="@drawable/background">

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/enter_calories_eaten"
    android:id="@+id/textView16"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="42dp" />

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="62dp"
    android:layout_weight="0.09"
    android:layout_marginTop="10dp"
    android:weightSum="1">

```

```

<EditText
    android:layout_width="114dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/calories_eaten"
    android:layout_alignBottom="@+id/button7"
    android:layout_toLeftOf="@+id/button7"
    android:layout_toStartOf="@+id/button7"
    android:layout_weight="0.80"
    android:singleLine="true" />

```

```

<Button
    android:layout_width="84dp"
    android:layout_height="wrap_content"
    android:text="@string/enter"
    android:id="@+id/button7"
    android:layout_gravity="center_horizontal"
    android:background="@drawable/buttonshape"

```

```
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:onClick="addTotal"
    android:layout_weight="0.20" />
```

```
</LinearLayout>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/today_calories_eaten"
    android:id="@+id/textView18"
    android:layout_below="@+id/button7"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/zero"
    android:id="@+id/calories_eaten_display"
    android:layout_below="@+id/textView18"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_horizontal" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/back_to_dashboard"
    android:id="@+id/button8"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:layout_marginTop="40dp"
    android:onClick="toDash"
    android:layout_gravity="center_horizontal" />
```

```
</LinearLayout>
```



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_cardio" tools:context="cse280.HealthyLifeStyle.Cardio"
    android:orientation="vertical">
```

```
<Spinner
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/spinner"
    android:layout_weight="0.00"
    android:spinnerMode="dropdown"/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Время:"
    android:id="@+id/textView31"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.00" />
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:ems="10"
    android:id="@+id/edit_time"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.00"
    android:hint="В Минутах" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Дистанция:"
    android:id="@+id/textView19"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.00" />
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:inputType="numberDecimal"
android:ems="10"
android:id="@+id/edit_distance"
android:layout_gravity="center_horizontal"
android:layout_weight="0.00"
android:hint="В км" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/calorie_burn"
    android:id="@+id/textView32"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.00" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:id="@+id/calories_burned_display"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.00" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ввод"
    android:id="@+id/button5"
    android:layout_weight="0.00"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:layout_gravity="center_horizontal"
    android:onClick="calcBurned" />
```

```
<Button
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Назад К Упражнениям"
    android:id="@+id/button11"
    android:layout_weight="0.00"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
```

```

        android:shadowRadius="5"
        android:textColor="#FFFFFF"
        android:textSize="20sp"
        android:layout_gravity="center_horizontal"
        android:onClick="back" />
</LinearLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="cse280.HealthyLifeStyle.CreateAccount"
    tools:showIn="@layout/activity_create_account"
    android:orientation="vertical"
    android:weightSum="1"
    android:background="@drawable/background2">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="@string/name"
            android:textColor="@color/colorText"
            android:id="@+id/textView20" />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPersonName"
            android:ems="10"
            android:id="@+id/set_name"
            android:singleLine="true"
            android:hint="@string/hint_name" />
    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="46dp">

        <TextView
            android:layout_width="wrap_content"

```

```
android:layout_height="wrap_content"  
android:textAppearance="?android:attr/textAppearanceMedium"  
android:text="@string/password"  
android:textColor="@color/colorText"  
android:id="@+id/asdsad" />
```

```
<EditText  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:password="true"  
  android:id="@+id/set_password"  
  android:layout_weight="1"  
  android:singleLine="true"  
  android:hint="@string/hint_pass" />
```

```
</LinearLayout>
```

```
<LinearLayout  
  android:orientation="horizontal"  
  android:layout_width="match_parent"  
  android:layout_height="wrap_content"  
  android:layout_gravity="right"  
  android:weightSum="1">
```

```
<TextView  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:textAppearance="?android:attr/textAppearanceMedium"  
  android:text="@string/age"  
  android:textColor="@color/colorText"  
  android:id="@+id/textView22" />
```

```
<EditText  
  android:layout_width="111dp"  
  android:layout_height="wrap_content"  
  android:inputType="number"  
  android:ems="10"  
  android:id="@+id/set_age"  
  android:hint="@string/hint_age"  
  android:singleLine="true" />
```

```
</LinearLayout>
```

```
<LinearLayout  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:orientation="horizontal"  
  android:layout_weight="0.49">
```

```
<TextView  
  android:layout_width="78dp"  
  android:layout_height="wrap_content"
```

```
android:textColor="@color/colorText"
android:textAppearance="?android:attr/textAppearanceMedium"
android:text="@string/gender"
android:id="@+id/textView45" />
```

```
<RadioGroup
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
```

```
<RadioButton
```

```
android:layout_width="99dp"
android:layout_height="wrap_content"
android:text="@string/male"
android:textColor="@color/colorText"
android:id="@+id/male"
android:checked="false"
android:onClick="onRadioButtonClicked" />
```

```
<RadioButton
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/female"
android:textColor="@color/colorText"
android:id="@+id/female"
android:checked="false"
android:onClick="onRadioButtonClicked" />
```

```
</RadioGroup>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
android:orientation="horizontal"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:weightSum="1">
```

```
<TextView
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceMedium"
android:text="@string/height"
android:textColor="@color/colorText"
android:id="@+id/textView43"
android:layout_weight="0.46"
android:paddingRight="90dp" />
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/weight"
        android:textColor="@color/colorText"
        android:id="@+id/textView44"
        android:layout_weight="0.49" />
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
<EditText
    android:layout_width="111dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/set_height"
    android:layout_marginRight="25dp"
    android:hint="@string/hint_inches"
    android:singleLine="true" />
```

```
<EditText
    android:layout_width="149dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/set_weight"
    android:hint="@string/hint_pounds"
    android:singleLine="true" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:weightSum="1">
```

```
<TextView
    android:layout_width="110dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/target_weight"
    android:textColor="@color/colorText"
    android:id="@+id/textView463"
    android:layout_weight="2.04"
    android:paddingRight="5dp" />
```

```
<TextView
    android:layout_width="175dp"
```

```
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/desired_weekly_workouts"
        android:textColor="@color/colorText"
        android:id="@+id/textView454" />
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
<EditText
    android:layout_width="111dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/set_goal"
    android:layout_marginRight="25dp"
    android:hint="@string/hint_pounds"
    android:singleLine="true" />
```

```
<EditText
    android:layout_width="149dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/set_desired_workouts"
    android:hint="@string/hint_days"
    android:singleLine="true"
    android:autoText="false"
    android:password="false"/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:weightSum="1">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/activity_level"
    android:id="@+id/textView434"
    android:textColor="@color/colorText"
    android:layout_weight="0.46" />
```

```
<TextView
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/time_frame"
        android:textColor="@color/colorText"
        android:id="@+id/textView445"
        android:layout_weight="0.49" />
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:weightSum="1"
    android:layout_weight="0.09">
```

```
<EditText
    android:layout_width="139dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/set_activity_level"
    android:hint="@string/hint_activity"
    android:singleLine="true" />
```

```
<RadioGroup
    android:layout_width="177dp"
    android:layout_height="83dp">
```

```
<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/time_under_3"
    android:id="@+id/one"
    android:textColor="@color/colorText"
    android:checked="false"
    android:onClick="onRadioButtonClicked" />
```

```
<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/time_under_6"
    android:id="@+id/three"
    android:checked="false"
    android:textColor="@color/colorText"
    android:onClick="onRadioButtonClicked" />
```

```
<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/time_over_6"
    android:textColor="@color/colorText"
```



```
    android:id="@+id/six"
    android:checked="false"
    android:onClick="onRadioButtonClicked" />
```

```
</RadioGroup>
</LinearLayout>
```

```
<Button
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/done"
    android:id="@+id/done"
    android:layout_gravity="center_horizontal"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:onClick="done" />
```

```
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_dash_board"
    tools:context="cse280.HealthyLifeStyle.DashBoard"
    android:background="@drawable/background2">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:weightSum="1"
        android:orientation="vertical">
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="250dp"
            android:background="@color/colorBack"
            android:layout_height="wrap_content"
            android:layout_weight="0.05">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/name"
    android:layout_weight="0.33"
    android:textColor="@color/colorText"
    android:id="@+id/textView"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:textStyle="bold|italic" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/weight"
    android:layout_weight="0.33"
    android:textColor="@color/colorText"
    android:id="@+id/textView5"
    android:layout_above="@+id/lost_gained"
    android:layout_alignRight="@+id/textView2"
    android:layout_alignEnd="@+id/textView2"
    android:textStyle="bold|italic" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/bmi"
    android:layout_weight="0.33"
    android:textColor="@color/colorText"
    android:id="@+id/textView6"
    android:layout_alignTop="@+id/textView5"
    android:layout_alignLeft="@+id/textView3"
    android:layout_alignStart="@+id/textView3"
    android:textStyle="bold|italic" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.09"
    android:background="@color/colorBack">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:id="@+id/name_display"
    android:layout_weight="0.33"
```

```
android:textColor="@color/colorPrimaryDark"  
android:layout_below="@+id/textView"  
android:layout_alignParentLeft="true"  
android:layout_alignParentStart="true"  
android:text="Димаш"  
android:textStyle="bold|italic"  
android:textSize="20dp"  
android:autoText="false" />
```

```
<TextView  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:textAppearance="?android:attr/textAppearanceSmall"  
  android:id="@+id/weight_display"  
  android:textColor="@color/colorPrimaryDark"  
  android:layout_weight="0.33"  
  android:layout_below="@+id/textView5"  
  android:layout_alignLeft="@+id/textView5"  
  android:layout_alignStart="@+id/textView5"  
  android:text="150"  
  android:textStyle="normal|bold|italic"  
  android:textSize="20dp" />
```

```
<TextView  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:textAppearance="?android:attr/textAppearanceSmall"  
  android:id="@+id/bmi_display"  
  android:textColor="@color/colorPrimaryDark"  
  android:layout_weight="0.33"  
  android:layout_below="@+id/textView6"  
  android:layout_toRightOf="@+id/button"  
  android:layout_toEndOf="@+id/button"  
  android:text="30"  
  android:textStyle="normal|bold|italic"  
  android:textSize="20dp" />
```

```
</LinearLayout>
```

```
<LinearLayout  
  android:orientation="horizontal"  
  android:layout_width="250dp"  
  android:layout_height="wrap_content"  
  android:layout_weight="0.05">
```

```
<TextView  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Потеряно кг"  
  android:textColor="@color/colorText"  
  android:id="@+id/lost_gained"
```

```
android:layout_below="@+id/textView4"
android:layout_toRightOf="@+id/textView10"
android:layout_toEndOf="@+id/textView10"
android:layout_weight="0.33"
android:paddingRight="10dp"
android:password="false"
android:phoneNumber="false"
android:textStyle="bold|italic" />
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/pounds_left"
  android:textColor="@color/colorText"
  android:id="@+id/textView9"
  android:layout_above="@+id/textView10"
  android:layout_alignLeft="@+id/textView3"
  android:layout_alignStart="@+id/textView3"
  android:layout_weight="0.33"
  android:paddingRight="5dp"
  android:textStyle="bold|italic" />
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/pounds_goal"
  android:textColor="@color/colorText"
  android:id="@+id/textView8"
  android:layout_above="@+id/textView10"
  android:layout_alignLeft="@+id/textView15"
  android:layout_alignStart="@+id/textView15"
  android:layout_weight="0.33"
  android:textStyle="bold|italic" />
```

```
</LinearLayout>
```

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="200dp"
  android:layout_height="wrap_content"
  android:layout_weight="0.09">
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceSmall"
  android:id="@+id/lbs_lost"
  android:layout_alignTop="@+id/lbs_left"
  android:layout_toRightOf="@+id/button"
  android:layout_toEndOf="@+id/button"
  android:layout_weight="0.4"
```

```
android:textColor="@color/colorPrimaryDark"  
android:text="15"  
android:textStyle="normal|bold|italic"  
android:textSize="20dp" />
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceSmall"  
    android:id="@+id/lbs_left"  
    android:layout_alignLeft="@+id/weight_display"  
    android:layout_alignStart="@+id/weight_display"  
    android:layout_below="@+id/lost_gained"  
    android:layout_weight="0.4"  
    android:textColor="@color/colorPrimaryDark"  
    android:text="20"  
    android:textSize="20dp"  
    android:textStyle="normal|bold|italic" />
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceSmall"  
    android:text="175"  
    android:id="@+id/goal_weight_display"  
    android:layout_below="@+id/textView8"  
    android:textColor="@color/colorPrimaryDark"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:layout_weight="0.2"  
    android:textSize="20dp"  
    android:textStyle="normal|bold|italic" />
```

```
</LinearLayout>
```

```
<LinearLayout  
    android:orientation="horizontal"  
    android:layout_width="240dp"  
    android:layout_height="wrap_content"  
    android:layout_weight="0.05"  
    android:weightSum="1">
```

```
<TextView  
    android:layout_width="70dp"  
    android:layout_height="wrap_content"  
    android:text="@string/maintenance"  
    android:textColor="@color/colorText"  
    android:id="@+id/textView10"  
    android:layout_below="@+id/lost_gained"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"
```

```
android:layout_weight="0.3"
android:textStyle="bold|italic" />
```

```
<TextView
  android:layout_width="70dp"
  android:layout_height="wrap_content"
  android:text="@string/daily"
  android:textColor="@color/colorText"
  android:id="@+id/textView11"
  android:layout_alignTop="@+id/textView10"
  android:layout_centerHorizontal="true"
  android:password="false"
  android:phoneNumber="false"
  android:layout_weight="0.2"
  android:textStyle="bold|italic" />
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/target_diff"
  android:textColor="@color/colorText"
  android:id="@+id/textView12"
  android:layout_alignTop="@+id/textView11"
  android:layout_toRightOf="@+id/textView6"
  android:layout_toEndOf="@+id/textView6"
  android:layout_weight="0.3"
  android:textStyle="bold|italic" />
```

```
</LinearLayout>
```

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="240dp"
  android:layout_height="wrap_content"
  android:layout_weight="0.20">
```

```
<TextView
  android:layout_width="60dp"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceSmall"
  android:text="2000"
  android:id="@+id/natural_display"
  android:layout_below="@+id/textView10"
  android:layout_alignParentLeft="true"
  android:layout_alignParentStart="true"
  android:layout_weight="0.15"
  android:textColor="@color/colorPrimaryDark"
  android:textSize="20dp"
  android:textStyle="normal|bold|italic" />
```

```
<TextView
```

```
android:layout_width="20dp"
android:layout_height="106dp"
android:textAppearance="?android:attr/textAppearanceSmall"
android:text="1800"
android:layout_weight="0.7"
android:paddingRight="1dp"
android:paddingLeft="1dp"
android:id="@+id/target_display"
android:layout_alignTop="@+id/natural_display"
android:layout_toRightOf="@+id/textView13"
android:layout_toEndOf="@+id/textView13"
android:textColor="@color/colorPrimaryDark"
android:textSize="20dp"
android:textStyle="normal|bold|italic" />
```

```
<TextView
    android:layout_width="40dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:text="-200"
    android:id="@+id/difference_display"
    android:layout_weight="0.15"
    android:layout_alignTop="@+id/target_display"
    android:layout_alignRight="@+id/textView12"
    android:layout_alignEnd="@+id/textView12"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="20dp"
    android:textStyle="normal|bold|italic" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.04">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/calorie_intake"
    android:id="@+id/textView13"
    android:paddingRight="40dp"
    android:layout_below="@+id/textView10"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:textStyle="normal|bold|italic" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```

        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="0"
        android:id="@+id/textView30"
        android:layout_alignBottom="@+id/textView15"
        android:layout_below="@+id/textView14"
        android:layout_alignLeft="@+id/textView29"
        android:layout_alignStart="@+id/textView29"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20dp"
        android:textStyle="normal|bold|italic" />
</LinearLayout>

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.04"
    android:weightSum="1">

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/calorie_burn"
    android:id="@+id/textView14"
    android:layout_below="@+id/textView13"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:paddingRight="75dp"
    android:textStyle="bold|italic" />

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:id="@+id/textView29"
    android:layout_below="@+id/remaining_cals"
    android:layout_alignLeft="@+id/difference_display"
    android:layout_alignStart="@+id/difference_display"
    android:text="0"
    android:textStyle="normal|bold|italic"
    android:textSize="20dp"
    android:textColor="@color/colorPrimaryDark"
    android:textColorHighlight="@color/colorPrimaryDark" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```



```
android:layout_weight="0.09"  
android:weightSum="1">
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceLarge"  
    android:text="@string/calorie_remain"  
    android:id="@+id/textView15"  
    android:layout_below="@+id/textView14"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:paddingRight="165dp"  
    android:textStyle="bold|italic" />
```

```
<TextView  
    android:layout_width="52dp"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceMedium"  
    android:text="0"  
    android:id="@+id/remaining_cals"  
    android:layout_above="@+id/textView14"  
    android:layout_alignParentRight="true"  
    android:layout_alignParentEnd="true"  
    android:layout_alignTop="@+id/textView13"  
    android:layout_alignLeft="@+id/textView29"  
    android:layout_alignStart="@+id/textView29"  
    android:textColor="@color/colorPrimaryDark"  
    android:textStyle="normal|bold|italic"  
    android:textSize="20dp" />
```

```
</LinearLayout>
```

```
<LinearLayout  
    android:orientation="horizontal"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="0.05">
```

```
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/action_food"  
        android:id="@+id/button3"  
        android:onClick="openCalorieInput"  
        android:background="@drawable/buttonshape"  
        android:shadowColor="#A8A8A8"  
        android:shadowDx="0"  
        android:shadowDy="0"  
        android:shadowRadius="5"  
        android:textColor="#FFFFFF"  
        android:textSize="15sp"
```

```

android:layout_weight="0.5"
android:layout_marginRight="6dp"/>

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/action_stats"
android:id="@+id/button"
android:layout_centerHorizontal="true"
android:onClick="openStats"
android:background="@drawable/buttonshape"
android:shadowColor="#A8A8A8"
android:shadowDx="0"
android:shadowDy="0"
android:shadowRadius="5"
android:textColor="#FFFFFF"
android:textSize="15sp"
android:layout_weight="0.5"/>
</LinearLayout>

```

```

<LinearLayout
android:orientation="horizontal"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="0.05">

```

```

<Button
android:layout_marginRight="6dp"
android:layout_marginTop="0dp"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/action_workout"
android:id="@+id/button2"
android:onClick="openWorkout"
android:background="@drawable/buttonshape"
android:shadowColor="#A8A8A8"
android:shadowDx="0"
android:shadowDy="0"
android:shadowRadius="5"
android:textColor="#FFFFFF"
android:textSize="15sp"
android:layout_weight="0.5"/>

```

```

<Button
android:layout_marginTop="0dp"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Сброс"
android:id="@+id/button4"
android:onClick="openHistory"
android:background="@drawable/buttonshape"

```

```

        android:shadowColor="#A8A8A8"
        android:shadowDx="0"
        android:shadowDy="0"
        android:shadowRadius="5"
        android:textColor="#FFFFFF"
        android:textSize="15sp"
        android:layout_weight="0.5"/>
    </LinearLayout>
</LinearLayout>
</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_weights" tools:context="cse280.HealthyLifeStyle.Cardio"
    android:orientation="vertical">

    <Spinner
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/spinner"
        android:layout_weight="0.00"
        android:spinnerMode="dropdown" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Бес:"
        android:id="@+id/textView31"
        android:layout_gravity="center_horizontal"
        android:layout_weight="0.00" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:ems="10"
        android:id="@+id/edit_weight"
        android:layout_gravity="center_horizontal"
        android:layout_weight="0.00"
        android:hint="Прибавлено отягащений" />

    <TextView

```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="Повторения:"
android:id="@+id/dasdas"
android:layout_gravity="center_horizontal"
android:layout_weight="0.00" />
```

```
<EditText
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:inputType="number"
  android:ems="10"
  android:id="@+id/edit_time"
  android:layout_gravity="center_horizontal"
  android:layout_weight="0.00"
  android:hint="Кол-во повторений" />
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceLarge"
  android:text="Подходы:"
  android:id="@+id/textView19"
  android:layout_gravity="center_horizontal"
  android:layout_weight="0.00" />
```

```
<EditText
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:inputType="number"
  android:ems="10"
  android:id="@+id/edit_distance"
  android:layout_gravity="center_horizontal"
  android:layout_weight="0.00"
  android:hint="Кол-во подходов" />
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceLarge"
  android:text="@string/calorie_burn"
  android:id="@+id/textView32"
  android:layout_gravity="center_horizontal"
  android:layout_weight="0.00" />
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceLarge"
  android:id="@+id/calories_burned_display"
```

```
android:layout_gravity="center_horizontal"
android:layout_weight="0.00" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ввод"
    android:id="@+id/button5"
    android:layout_weight="0.00"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:layout_gravity="center_horizontal"
    android:onClick="calcBurned" />
```

```
<Button
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Назад К УПРАЖНЕНИЯМ"
    android:id="@+id/button11"
    android:layout_weight="0.00"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:layout_gravity="center_horizontal"
    android:onClick="back" />
```

```
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_workout" tools:context="cse280.HealthyLifeStyle.Workout"
    android:orientation="vertical"
    android:weightSum="1"
    android:background="@drawable/background">
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Кардио"
    android:id="@+id/cardio_button"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="15sp"
    android:layout_gravity="center_horizontal"
    android:onClick="cardio"
    android:layout_weight="0"/>
```

```
<Button
    android:layout_marginTop="10dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Собственный вес"
    android:id="@+id/button6"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="15sp"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0"
    android:onClick="bodyWeight"/>
```

```
<Button
    android:layout_marginTop="10dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Отягощение"
    android:id="@+id/button10"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="15sp"
    android:layout_gravity="center_horizontal"
    android:onClick="weights"
```

```
android:layout_weight="0"/>
```

```
<TextView  
    android:layout_width="274dp"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceMedium"  
    android:text="Или введите вручную"  
    android:id="@+id/textView17"  
    android:layout_marginLeft="50dp"  
    android:layout_marginTop="10dp"/>
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/enter_calories_burned"  
    android:id="@+id/d"  
    android:layout_alignParentTop="true"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:layout_marginTop="10dp"  
    android:textSize="20dp" />
```

```
<LinearLayout  
    android:orientation="horizontal"  
    android:layout_width="match_parent"  
    android:layout_height="70dp">
```

```
<EditText  
    android:layout_width="wrap_content"  
    android:layout_height="65dp"  
    android:inputType="number"  
    android:ems="10"  
    android:id="@+id/calories_burned"  
    android:singleLine="true" />
```

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/enter"  
    android:id="@+id/updateburn"  
    android:background="@drawable/buttonshape"  
    android:shadowColor="#A8A8A8"  
    android:shadowDx="0"  
    android:shadowDy="0"  
    android:shadowRadius="5"  
    android:textColor="#FFFFFF"  
    android:textSize="15sp"  
    android:layout_gravity="center_horizontal"
```

```
        android:onClick="addTotal"
        android:layout_weight=".20" />
</LinearLayout>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/today_calories_burned"
    android:id="@+id/total_calories_burned"
    android:layout_below="@+id/updateburn"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/zero"
    android:id="@+id/calories_burned_today"
    android:layout_below="@+id/total_calories_burned"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="15dp"
    android:layout_gravity="center_horizontal" />
```

```
<Button
    android:layout_width="202dp"
    android:layout_height="wrap_content"
    android:text="@string/back_to_dashboard"
    android:id="@+id/button9"
    android:background="@drawable/buttonshape"
    android:shadowColor="#A8A8A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:textColor="#FFFFFF"
    android:textSize="15sp"
    android:layout_gravity="center_horizontal"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:onClick="toDashboard" />
```

```
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
    <color name="colorPrimary">#942171</color>
    <color name="colorPrimaryDark">#6c1b58</color>
    <color name="colorAccent">#FF4081</color>
    <color name="colorText">#0a0909</color>
    <color name="colorTextB">#43528f</color>
```



```
<color name="colorTextL">#9fd3d9</color>
<color name="colorBack">#f9fdfd</color>
</resources>
```

```
<resources>
  <!-- Default screen margins, per the Android Design guidelines. -->
  <dimen name="activity_horizontal_margin">10dp</dimen>
  <dimen name="activity_vertical_margin">5dp</dimen>
  <!-- Default screen margins, per the Android Design guidelines. -->
  <dimen name="nav_header_vertical_spacing">16dp</dimen>
  <dimen name="nav_header_height">160dp</dimen>
  <dimen name="fab_margin">16dp</dimen>
</resources>
```

```
<resources>
  <string name="app_name">HealthyLifeStyle</string>
```

```
<!-- Activity Title Strings -->
```

```
<string name="title_activity_dash">Путь к здоровью</string>
```

```
<string name="title_activity_login">HLS</string>
```

```
<string name="title_activity_calorie_input">Ввод калории</string>
```

```
<string name="title_activity_dash_board">Путь к здоровью</string>
```

```
<string name="title_activity_stats">Данные пользователя</string>
```

```
<string name="title_activity_workout">Упражнения</string>
```

```
<string name="title_activity_history">История</string>
```

```
<string name="title_activity_create_account">Регистрация</string>
```

```
<!-- Strings related to Login -->
```

```
<string name="hint_username">пользователь</string>
```

```
<string name="hint_password">пароль</string>
```

```
<string name="login">Вход</string>
```

```
<string name="create_account">Создать аккаунт</string>
```

```
<string name="sign_in">регистрация</string>
```

```
<string name="email">почта</string>
```

```
<!-- Strings related to Workout -->
```

```
<string name="enter_calories_burned">кол-во сожженных калорий:</string>
```

```
<string name="today_calories_burned">сегодня сожжено:</string>
```

```
<string name="back_to_dashboard">На Главную</string>
```

```
<!-- Strings related to Food -->
```

```
<string name="enter_calories_eaten">Введите Употребленные Калории:</string>
```

```
<string name="today_calories_eaten">Сегодня Употреблено:</string>
```

```
<!-- Strings related to Food and Workout -->
<string name="enter">ВВОД</string>
<string name="zero">0</string>
```

```
<!-- Strings related to Stats, Account, and DashBoard -->
<string name="name">Имя:</string>
<string name="hint_name">10 char max</string>
<string name="hint_pass">16 char max</string>
<string name="hint_age">1-120</string>
<string name="age">Возраст:</string>
<string name="gender">Пол:</string>
<string name="height">Рост:</string>
<string name="weight">Вес:</string>
```

```
<!-- Strings related to Stats and Account-->
<string name="password">Пароль:</string>
<string name="male">Мужчина</string>
<string name="female">Женщина</string>
<string name="target_weight">Намеченный вес:</string>
<string name="desired_weekly_workouts">Желательное # Занятия В Неделю: </string>
<string name="activity_level">Ур. активности:</string> <!-- Consider array for list-->
<string name="activity_low">Низкий</string>
<string name="activity_medium">Средний</string>
<string name="activity_high">Высокий</string>
<string name="time_frame">временные рамки:</string><!-- Consider date input -->
<string name="time_under_3">1-3 Месяцев</string>
<string name="time_under_6">3-6 Месяцев</string>
<string name="time_over_6">6+ Месяцев</string>
<string name="done">Завершить</string>
```

```
<!-- Strings related to Account -->
<string name="date_of_birth">Дата рождения:</string>
<string name="hint_inches">В См</string>
<string name="hint_pounds">В кг</string>
<string name="hint_days">1-14</string>
<string name="hint_activity">1(низ) - 5(выс)</string>
```

```
<!-- Strings related to DashBoard -->
<string name="bmi">ИМТ:</string>
<string name="pounds_lost">Потеряно кг:</string>
<string name="pounds_left">Осталось кг:</string>
<string name="pounds_goal">Намеченный вес:</string>
<string name="daily_cals">Намеченные калорий:</string>
<string name="today_total">Общее сегодня:</string>
<string name="target_diff">Разница:</string>
<string name="calorie_intake">Сегодня употреблено</string>
```

```
<string name="calorie_burn">Сегодня сожжено</string>
<string name="calorie_remain">Осталось</string>
<string name="action_food">Прием пищи</string>
<string name="action_workout">Упражнения</string>
<string name="action_stats">Редактировать</string>
<string name="action_progress">След прогресса</string>
<!-- Try to replace the following two with better descriptions -->
<string name="maintenance">Кал-ий в среднем:</string>
<string name="daily">Суточная норма:</string>
```

```
<!-- Strings related to Navigation Drawer -->
<string name="navigation_drawer_open">Open navigation drawer</string>
<string name="navigation_drawer_close">Close navigation drawer</string>
```

```
<!-- Strings related to Settings -->
<string name="action_settings">Настройки</string>
<string name="title_activity_settings">Настройки</string>
<string name="pref_title_system_sync_settings">System sync settings</string>
<string name="pref_header_general">Общее</string>
<string name="pref_title_social_recommendations">Enable social recommendations</string>
<string name="pref_description_social_recommendations">Recommendations for people to
contact
based on your message history
</string>
```

```
<!-- Strings based on the user -->
<string name="pref_title_display_name">Display name</string>
<string name="pref_default_display_name">John Smith</string>
```

```
<string name="pref_title_add_friends_to_messages">Add friends to messages</string>
<string-array name="pref_example_list_titles">
<item>Always</item>
<item>When possible</item>
<item>Never</item>
</string-array>
<string-array name="pref_example_list_values">
<item>1</item>
<item>0</item>
<item>-1</item>
</string-array>
```

```
<!-- Example settings for Data & Sync -->
<string name="pref_header_data_sync">Data & sync</string>

<string name="pref_title_sync_frequency">Sync frequency</string>
<string-array name="pref_sync_frequency_titles">
<item>15 minutes</item>
```

```
<item>30 minutes</item>
<item>1 hour</item>
<item>3 hours</item>
<item>6 hours</item>
<item>Never</item>
</string-array>
<string-array name="pref_sync_frequency_values">
  <item>15</item>
  <item>30</item>
  <item>60</item>
  <item>180</item>
  <item>360</item>
  <item>-1</item>
</string-array>
```

```
<!-- Strings related to Errors / Permissions -->
<string name="error_invalid_email">Неправильный адрес почты</string>
<string name="error_invalid_password">Пароль слишком короткий</string>
<string name="error_incorrect_password">Неверный пароль</string>
<string name="error_field_required">This field is required</string>
<string name="permission_rationale">"Contacts permissions are needed for providing email
completions."</string>
```

```
<!-- Example settings for Notifications -->
<string name="pref_header_notifications">Подсказки</string>
<string name="pref_title_new_message_notifications">New message notifications</string>
<string name="pref_title_ringtone">Ringtone</string>
<string name="pref_ringtone_silent">Silent</string>
<string name="pref_title_vibrate">Vibrate</string>
<string name="title_activity_cardio">Кардио</string>
```

```
<!-- Exercises -->
<string-array name="cardio_workouts">
  <item>Ходьба</item>
  <item>Бег</item>
</string-array>
```

```
<string-array name="body_weight_workouts">
  <item>Отжимания</item>
  <item>Приседания</item>
</string-array>
```

```
<string-array name="weights_workouts">
  <item>Подъем Гантели На Бицепс</item>
  <item>Подъем Туловища</item>
  <item>Приседания</item>
  <item>Становая тяга</item>
</string-array>
```

```
<string name="title_activity_body_weight_exercise">С собственным весом</string>
<string name="title_activity_weights">С отягощениями</string>
```

```
</resources>
```

```
<resources>
```

```
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
    <style name="AppTheme.NoActionBar">
        <item name="windowActionBar">>false</item>
        <item name="windowNoTitle">>true</item>
    </style>
    <style name="AppTheme.AppBarOverlay"
parent="ThemeOverlay.AppCompat.Dark.ActionBar" />
    <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
```

```
</resources>
```

```
import java.util.*;
/*
unused class, used for documentation and reference
*/
public class WorkoutDB
{
    private static Map<String,Integer> idList = new HashMap<>();
    private static Map<Integer,Double> coList = new HashMap<>();

    public WorkoutDB()
    {
        super();
        loadDB();
    }

    public static void loadDB()
    {
        idList.clear();
        //cardio
        idList.put("run",0);
        idList.put("walk",1);

        //bodyweight
        idList.put("pushups",10);
```

```

idList.put("situps",11);

//weightlifting
idList.put("curls",100);
idList.put("benchpress",101);
idList.put("squats",102);
idList.put("deadlifts",103);

coList.clear();
coList.put(0,0.0124);
coList.put(1,0.0085);

coList.put(10,0.0009); // 0.036 per min *assume 40/min, 0.0009 cal/pushup
coList.put(11,0.000655); // 0.0262 per min *assume 40/min, 0.000655 cal/pushup

coList.put(100,0.005832861);
coList.put(101,0.00661057); // cal/pound/rep //physics/anatomy calculation
coList.put(102,0.00972); //assume distance lifted is 1.7ft
coList.put(103,0.00972); //assume distance lifted is 1.7ft
}

public static Integer getId(String name)
{
    if(idList.containsKey(name)){
        return idList.get(name);
    }
    return -1;
}

public static double getCoefficient(Integer id)
{
    if(coList.containsKey(id)){
        return coList.get(id);
    }
    return -1.0;
}
}

```