

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа ядерных технологий
 Направление подготовки 01.03.02 «Прикладная математика и информатика»
 Отделение школы (НОЦ) Отделение экспериментальной физики

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
Разработка мобильного приложения для самодиагностики невусов

УДК 004.415:004.451:616.5-006.81-071.2

Обучающийся

Группа	ФИО	Подпись	Дата
0В92	Бутковская Светлана Владимировна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Лисок Александр Леонидович	к. ф.-м. н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кащук Ирина Вадимовна	к. т. н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич Ольга Алексеевна	к. б. н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП/ОПОП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Крицкий Олег Леонидович	к. ф.-м. н.		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа ядерных технологий
 Направление подготовки 01.03.02 «Прикладная математика и информатика»
 Отделение школы (НОЦ) Отделение экспериментальной физики

УТВЕРЖДАЮ:
 Руководитель ООП/ОПОП

 (Подпись) (Дата) (ФИО)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

Обучающийся:

Группа	ФИО
0B92	Бутковская Светлана Владимировна

Тема работы:

Разработка мобильного приложения для самодиагностики невусов
<i>Утверждена приказом директора (дата, номер)</i>

Срок сдачи обучающимся выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к функционированию (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.)</i></p>	<p>В ходе выполнения выпускной квалификационной работы необходимо разработать мобильное приложение на основе OS IOS, обеспечивающее пользователям возможность мониторинга изменения невусов, контроль внешних признаков невусов для самодиагностики. Предоставление пользователю информации о погоде. Возможность пользователю выбрать нужный фототип кожи и после этого получить информацию о необходимых мерах защиты в выбранном городе в зависимости от значения УФ-индекса.</p>
<p>Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке <i>(аналитический обзор литературных источников с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе)</i></p>	<ol style="list-style-type: none"> 1. литературный обзор; 2. анализ медицинских критериев диагностики невусов; 3. выбор и обоснование архитектуры приложения; 4. верстка приложения, написание программного кода; <p>В качестве дополнительных разделов: – Социальная ответственность;</p>

	<ul style="list-style-type: none"> – Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; – Заключение
--	--

Перечень графического материала

(с точным указанием обязательных чертежей)

Консультанты по разделам выпускной квалификационной работы

(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Кащук Ирина Вадимовна
Социальная ответственность	Антоневич Ольга Алексеевна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	06.02.2023
---	------------

Задание выдал руководитель / консультант (при наличии):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Лисок Александр Леонидович	к.ф.-м.н, доцент		06.02.2023

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
0В92	Бутковская Светлана Владимировна		06.02.2023

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
0B92	Бутковская Светлана Владимировна

Школа	ИШНКБ	Отделение	ОЭИ
Уровень образования	Бакалавриат	Направление/специальность	01.03.02 Прикладная математика и информатика

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Стоимость материальных ресурсов и специального оборудования определены в соответствии с рыночными ценами г.Томска Тарифные ставки исполнителей определены штатным расписанием НИ ТПУ.
2. Нормы и нормативы расходования ресурсов	Норма амортизационных отчислений на специальное оборудование
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Отчисления во внебюджетные фонды 30,2%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Выполнение SWOT-анализа проекта Анализ конкурентоспособности
2. Определение возможных альтернатив проведения научных исследований	Определение целей и ожиданий, требований проекта. Определение заинтересованных сторон и их ожиданий.
3. Планирование процесса управления НИ: структура и график проведения, бюджет, риски и организация закупок	Составление календарного плана проекта. Определение бюджета НИ
4. Определение ресурсной, финансовой, экономической эффективности	Интегральный финансовый показатель. Интегральный показатель ресурсоэффективности. Интегральный показатель эффективности

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. График проведения НИ
4. Определение бюджета НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	13.02.2023
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Кашук Ирина Вадимовна	к.т.н		13.02.2023

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0B92	Бутковская Светлана Владимировна		13.02.2023

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
0B92	Бутковская Светлана Владимировна

Школа	ИШНКБ	Отделение (НОЦ)	ОЭИ
Уровень образования	Бакалавриат	Направление/специальность	01.03.02 Прикладная математика и информатика

Тема ВКР:

Разработка мобильного приложения самодиагностики невусов	
Исходные данные к разделу «Социальная ответственность»:	
<p>Введение</p> <ul style="list-style-type: none"> - Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения. - Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации. 	<p>Объект исследования: мобильное приложение для самодиагностики невусов.</p> <p>Область применения: медицина</p> <p>Рабочая зона: офис.</p> <p>Размеры помещения: 20 м².</p> <p>Количество и наименование оборудования рабочей зоны: ноутбук на операционной системе macOS.</p> <p>Рабочие процессы, связанные с объектом исследования, осуществляемые в рабочей зоне: алгоритмическая и программная разработка с использованием ноутбука.</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</p> <ul style="list-style-type: none"> - специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; - организационные мероприятия при компоновке рабочей зоны. 	<ul style="list-style-type: none"> - Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032-78 «Рабочее место при выполнении работ сидя» - Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиН 2.2.4.3359-16 Санитарно-эпидемиологические требования к физическим факторам на рабочих местах - Трудовой кодекс Российской Федерации: Федеральный Закон от 30 дек. 2001 г. №197-ФЗ Раздел 10
<p>2. Производственная безопасность при разработке проектного решения:</p> <ul style="list-style-type: none"> - Анализ выявленных вредных и опасных производственных факторов - Расчет уровня опасного или вредного производственного фактора 	<p>Вредные факторы:</p> <ul style="list-style-type: none"> - Отклонение показателей микроклимата - Недостаточная освещенность рабочей зоны - Пониженная световая и цветовая контрастность - Повышенный уровень шума на рабочем месте - Повышенный уровень статического электричества <p>Опасные факторы:</p> <ul style="list-style-type: none"> - Опасность поражения электрическим током <p>Расчет: расчет системы искусственного освещения.</p>
<p>3. Экологическая безопасность при разработке проектного решения:</p>	<p>Воздействия на селитебную зону не происходит.</p> <p>Анализ воздействия на литосферу:</p> <ul style="list-style-type: none"> - Утилизация компьютеров, оргтехники и бумаги

	Анализ воздействия на гидросферу: – Производство компьютерной техники Воздействия на атмосферу не происходит
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения:	Возможные ЧС: - пожар Наиболее типичная ЧС – пожар

Дата выдачи задания для раздела по линейному графику	22.02.2023
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич Ольга Алексеевна	к.б.н		22.02.2023

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0В92	Бутковская Светлана Владимировна		22.02.2023

РЕФЕРАТ

Выпускная квалификационная работа содержит - 113 страниц, 19 - рисунков, 28 - таблиц, 25 - литературных источников и 8 приложений.

Ключевые слова: мобильное приложение, IOS, операционная система, SPF, UV-index, невус.

Цель работы: разработка мобильного приложения на базе операционной системы IOS для самодиагностики невусов.

Результатом работы является мобильное приложение на основе OS IOS, обеспечивающее пользователям возможность мониторинга изменения невусов, контроль внешних признаков невусов для самодиагностики. Предоставление пользователю информации о погоде. Возможность пользователю выбрать нужный фототип кожи и после этого получить информацию о необходимых мерах защиты в выбранном городе в зависимости от значения УФ-индекса.

В соответствии с целью определены следующие задачи:

- проанализировать аналогичные мобильные приложения в Интернете;
- проанализировать влияние УФ-индекса на организм и критерии диагностики невусов;
- изучить особенности разработки приложений для ОС IOS;
- определить требования к программе;
- спроектировать архитектуру разрабатываемого приложения;
- разработать мобильное приложение;
- выполнить тестирование.

СОДЕРЖАНИЕ

Список терминов и сокращений	10
Введение	11
Глава 1. Теоретическая часть	16
1.1 Анализ аналогичных мобильных приложений	16
1.2 Изучение влияния УФ-индекса на организм и критерии диагностики невусов	18
Глава 2. Практическая часть	20
2.1 Описание предметной области	20
2.2 Описание возможностей пользователей приложения	21
2.3 Паттерн проектирования MVP	22
2.4 Чистая архитектура	25
2.5 Внедрение зависимостей	26
2.6 Паттерн проектирования «Фасад»	27
2.7 Работа с сетью	28
2.8 Хранение данных пользователя с использованием Core Data	31
2.9 Детектирование злокачественных и доброкачественных новообразований при помощи сверточных нейронных сетей	38
2.10 Верстка	41
2.11 Тестирование	42
2.12 Описание и демонстрация функционала разработанного мобильного приложения	43
2.13 Демонстрация возможностей пользователей мобильного приложения	43
2.13.1 Изменение языка интерфейса мобильного приложения	44
2.13.2 Экран просмотра погоды	45
2.13.3 Экран выбора фототипа	46
2.13.4 Экран для динамического наблюдения за родинками и их самодиагностики	48
2.13.5 Экран для получения рекомендаций по защите кожи от UV лучей	49
2.14 Выводы по главе	51
Глава 3. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	52

3.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.	52
3.2 Планирование научно-исследовательских работ	55
3.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	70
Выводы по разделу	73
Глава 4. Социальная ответственность	74
Введение	74
4.1 Правовые и организационные вопросы обеспечения безопасности	74
4.2 Производственная безопасность	76
4.3 Экологическая безопасность	82
4.4 Безопасность в чрезвычайных ситуациях	84
Заключение	87
Список использованных источников	88
Приложение 1. Внедрение зависимостей	92
Приложение 2. Паттерн проектирования «Фасад»	94
Приложение 3. Модель данных на текущий день и на следующие пять дней	95
Приложение 4. Получение данных о погоде на текущий день в формате JSON, конвертация полученных данных.	96
Приложение 5. CoreData	100
Приложение 6. Предсказательная модель для определения злокачественных образований	101
Приложение 7. Верстка	104
Приложение 8. Тестирование	111

СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ

API – Application Programming Interface – программный интерфейс приложения;

JSON – формат обмена данными, запись которых идентична представлению объектов в JavaScript;

UI (User Interface или Пользовательский интерфейс) – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы;

URL – унифицированный указатель ресурса, определяющий его конкретное местоположение;

XML – «расширяемый язык разметки»;

iOS – мобильная операционная система для продуктов компании Apple;

БД – база данных;

ОС – операционная система;

ПО – программное обеспечение;

УФ-индекс (UV-index) – Ультрафиолетовый индекс;

Фреймворк – программное обеспечение, представляющее шаблон для программной платформы и облегчающее процесс разработки описание способов, которыми одна программа может взаимодействовать с другой программой.

ВВЕДЕНИЕ

Современные смартфоны стали незаменимыми гаджетами для большинства людей, а наличие персонального компьютера становится всё реже необходимым условием для связи с миром. Согласно исследованию Gartner, [1] продажи смартфонов в мире в 2017 году достигли почти 1,5 млрд штук, что на 100 млн больше, чем годом ранее. Каждый день создается огромное количество мобильных приложений, чтобы удовлетворить потребности пользователей, давая им возможность использовать смартфоны для выполнения наиболее разнообразных задач.

Сегодня все больше людей предпочитают быстрый и удобный доступ к информации и выбирают мобильные версии сайтов для поиска и получения нужного контента. В России, согласно статистике с сайта Cossa.ru, [2] 98% населения использует мобильные телефоны. Современные технологии позволяют создавать программное обеспечение для мобильных устройств, имеющее такой же функционал и характеристики, как и громоздкие компьютеры. В настоящее время практически все пользуются мобильными устройствами в повседневной жизни, их популярность обусловлена функциональностью, небольшим размером и доступом к каждому виду связи. Мобильные устройства заменяют настольный компьютер дома, так как все функции доступны на устройстве во время поездок и в других условиях. На сегодняшний день, основную классификацию мобильных телефонов делят на: BlackBerry, Symbian OS, Windows Mobile, Android, Mac OS и иные.

Главные преимущества iPhone — качественная оптимизация, отсутствие багов, высокая степень надежности. Правило Apple — конфиденциальность и безопасность пользователей имеют первостепенное значение. Поэтому с самого начала IOS была закрытой системой и более защищенной от кибератак по сравнению с Android.

Злокачественные новообразования кожи являются одной из самых распространенных патологий в мире. Согласно данным за 2020 год, [3] в России рак кожи занимает у женщин второе место по частоте встречаемости

— 14,5%, находясь после рака молочной железы (21,7%); у мужчин он занимает третье место - 10,6%, после рака трахеи, бронхов и легкого (16,5%) и рака предстательной железы (14,9%). Хотя единственной причины возникновения рака кожи установить невозможно, доказанным и наиболее распространенным фактором риска является длительное воздействие ультрафиолетовых лучей.

Солнечное облучение фактор, который при наличии генетической предрасположенности (светлая кожа с большим количеством родинок) провоцирует самообновление клеток. При постоянных повреждениях клеток, организм теряет контроль над процессами регенерации, что является начальным этапом онкологического процесса [3].

Злокачественные опухоли кожи могут возникнуть на месте родинки, которая раньше не доставляла никаких проблем в течение многих лет. Несмотря на очевидный кожный дефект, пациенты, столкнувшиеся с начальной стадией заболевания, зачастую стараются откладывать визит к врачу из-за страха услышать диагноз рак кожи. Однако, своевременное обращение к медицинским специалистам является крайне важным, поскольку прогноз заболевания в значительной степени зависит от того, насколько быстро было начато лечение.

Количество злокачественных заболеваний кожи непрерывно растет. Рак кожи стал самым часто диагностируемым онкологическим заболеванием в России в 2017 году, следует из данных Минздрава (рис. 1) [4].

Самые распространенные онкологические заболевания в России

Указано число злокачественных опухолей, диагностированных в 2017 году, тыс. штук



Источник: Минздрав

© РБК, 2018

Рисунок 1 - Статистика распространенности онкологических заболеваний [4]

Человек не всегда может вовремя распознать грозящую ему опасность. Обращение к квалифицированному врачу может быть затруднено из-за удаленности, а также по финансовым или другим причинам.

В связи с этим было решено разработать мобильное приложение, которое бы позволяло выбрать из галереи или сделать камерой смартфона фотографию подозрительной «родинки» и указать дополнительные данные о ее локализации на теле и типе кожи. По результатам самоанализа пользователь принимает решение о необходимости визита к врачу. Также отслеживание «родинки» во времени позволяет пользователю заметить изменения в цвете, размере, форме и вовремя обратиться к врачу.

Объектом выпускной квалификационной работы является предоставление пользователю функционала мобильного приложения для самодиагностики невусов и формирование рекомендаций по защите кожи в зависимости от значения УФ-индекса и фототипа пользователя.

Предметом квалификационной работы является мобильное приложение для самодиагностики невусов.

Целью выпускной квалификационной работы является разработка мобильного приложения на базе операционной системы IOS. В соответствии

с целью определены следующие задачи:

- проанализировать аналогичные мобильные приложения в Интернете;
- проанализировать влияние УФ-индекса на организм и критерии диагностики невусов;
- изучить особенности разработки мобильных приложений для ОС IOS;
- определить требования к программе;
- спроектировать архитектуру разрабатываемого приложения;
- разработать мобильное приложение;
- выполнить тестирование.

С целью определения заинтересованности потенциальных пользователей в приложении для самодиагностики невусов был проведен опрос, результаты которого представлены на рисунках ниже.

Таблица 1- Результаты опроса потенциальных пользователей приложения

Интересует ли Вас состоянием своего здоровья?	Как много времени Вы проводите на улице?	Следите ли Вы за состоянием своей кожи?	Как часто Вы посещаете дерматолога?	Было бы Вам полезно контролировать новообразования на Вашем теле?	Были ли у Ваших родственников онкологические заболевания кожи?	Интересует ли Вас приложение для самодиагностики невусов (родинок) на теле?
Да	От одного до трех часов	Скорее да	Раз в год	Скорее да	Скорее нет	Да
Да	Более трех часов	Скорее да	Раз в полгода	Скорее да	Нет	Да
Да	Час или менее	Да	Раз в полгода	Скорее да	Нет	Скорее да
Скорее нет	Час или менее	Скорее да	Раз в год	Скорее нет	Нет	Скорее нет
Да	От одного до трех часов	Да	Раз в год	Да	Нет	Скорее нет
Да	От одного до трех часов	Скорее нет	Раз в год	Скорее да	Нет	Скорее да

Да	От одного до трех часов	Скорее да	Раз в год	Скорее нет	Нет	Нет
Да	От одного до трех часов	Да	Раз в год	Да	Скорее нет	Да
Скорее да	Час или менее	Нет	Раз в год	Нет	Нет	Скорее да
Скорее да	От одного до трех часов	Скорее нет	Раз в год	Скорее нет	Нет	Нет
Да	Час или менее	Скорее нет	Раз в год	Скорее нет	Нет	Нет
Да	От одного до трех часов	Да	Раз в два месяца	Да	Нет	Нет

По результатам опроса можно сделать вывод, что 25% от общего числа опрошенных на вопрос «Интересует ли Вас приложение для самодиагностики невусов (родинок) на теле?» ответили «Да» и 25% опрошенных на этот же вопрос отвечают «Скорее да». Основываясь на этом, можно сделать вывод о заинтересованности потенциальных пользователей в разрабатываемом приложении.

ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 АНАЛИЗ АНАЛОГИЧНЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Для определения требований к мобильному приложению стоит выполнить анализ конкурирующих решений, представленных на рынке. Для анализа были выбраны следующие решения:

- «ProRodinki»
- «MoleScope»
- «Локализованный УФ-индекс»

Данные решения являются одними из самых популярных в российском сегменте рынка приложений, например, «ProRodinki» имеет более 100 тысяч установок на устройства.

Анализ данных решений позволит выявить требования для приложения, проектируемого и разрабатываемого в рамках данной выпускной квалификационной работы. В анализе (табл. 2) важно учесть недостатки и недоработки существующих решений, сделать это возможно с помощью их непосредственного использования или предоставленных пользовательских отзывов в магазине приложений для устройств под управлением операционной системы IOS – «App Store».

Таблица 2 - Анализ функционала аналогичных мобильных приложений

	ProRodinki	MoleScope	Локализованный УФ-индекс	Разрабатываемое приложение
Использует обученную нейросеть для оценивания необходимости посещения врача	+	-	-	-
Позволяет сохранять данные о родинках, наблюдать,	+	+	-	+

анализировать их				
Позволяет пользователю определить тип кожи сновываясь на шкале Фитцпатрика, которая классифицирует людей в зависимости от реакции их кожи на воздействие солнца	-	-	+	+
Позволяет пользователю получить персонализированные значения максимального пребывания на солнце, и других мер защиты от УФ излучения	-	-	+	+
Не требует отдельного устройства, присоединяющегося к камере	+	-	+	+
Предоставление пользователю информации о погоде	-	-	-	+

Основываясь на проведенном анализе (табл. 1) конкурентных решений, были выявлены и составлены требования к мобильному приложению, разрабатываемому в данной работе.

- Позволяет сохранять данные о родинках для мониторинга изменения и контроля внешних признаков для самодиагностики;

- Предоставляет пользователю информации о погоде;
- Дает пользователю возможность определить тип кожи основываясь на шкале Фитцпатрика, которая классифицирует людей в зависимости от реакции их кожи на воздействие солнца;
- Позволяет пользователю получить персонализированные значения максимального пребывания на солнце, и других мер защиты от УФ-излучения.

1.2 ИЗУЧЕНИЕ ВЛИЯНИЯ УФ-ИНДЕКСА НА ОРГАНИЗМ И КРИТЕРИИ ДИАГНОСТИКИ НЕВУСОВ

Родинки — это скопления пигмента в коже, которые могут быть плоскими или объемными: все зависит от того, насколько близко к поверхности кожи расположены пигментные клетки меланоциты [5].

Критерии диагностики родинок могут варьироваться в зависимости от типа родинки, но общие критерии включают:

1. Асимметрия: если одна часть родинки выглядит иначе, чем другая, это может быть признаком злокачественности.

2. Края: если края родинки неровные, зазубренные, размытые или нечеткие, это может быть признаком злокачественной родинки.

3. Цвет: если родинка имеет необычный цвет или цветовую шкалу, это может быть признаком злокачественности. Меланомы могут содержать несколько цветов, таких как черный, коричневый, красный, белый и синий.

4. Диаметр: если диаметр родинки большой (больше 6 мм), это может быть признаком злокачественности.

5. Изменения: любые изменения по размеру, форме, цвету или текстуре могут быть признаком злокачественной родинки.

Для оценки интенсивности повреждающего воздействия ультрафиолета дерматолог Томас Фитцпатрик, доктор медицинских наук – в 1975 году разработал классификацию фототипов, которой активно пользуются современные дерматологи (табл. 3).

Таблица 3 - Классификация фототипов дерматолога Томаса Фитцпатрика [6]

	I тип	II тип	III тип	IV тип	V тип	VI тип
Цвет кожи	светло-розовая, слоновая кость, молочно-белая	персиковая, светло-бежевая	светло-коричневая, светло-бежевая	оливковая, бронзовая	очень смуглая, желто-коричневая	темно-коричневая, черная
Цвет глаз	голубые, зеленые, светло-серые	голубые, серые, синие, зеленые, светло-карие	голубые, синие, серые, зеленые, карие	карие, черные	карие, черные	карие, черные
Цвет волос	блонд, рыжий, светло-русый	рыжий, все оттенки русого	русый, каштановый	каштановый, черный	каштановый, черный	черный
Солнечный ожог	всегда сильный	в большинстве случаев сильный	умеренно сильный	редко и не сильный	очень редко, выражен слабо	практически не бывает
Безопасное время на активном солнце, мин.	5-7	10	20	30	40	50

Прежде всего, свой фототип нужно знать, чтобы правильно оценить свой уровень чувствительности кожи в отношении к солнечному свету. Владея информацией об уровне чувствительности кожи можно обезопасить себя от рисков, связанных с ультрафиолетовым излучением.

ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметная область: контроль за состоянием родинок.

В данной предметной области существует проблема отслеживания состояния родинок, ведь любые их изменения служат поводом для обращения к врачу. Обычно контроль ведется с помощью мобильного телефона с помощью которого человек делает фотографию родинки либо отслеживание осуществляется в виде дневника на бумажном носителе куда человек записывает характеристики своей родинки. В дневник записываются ее характеристики, обычно, каждая запись включает: цвет, размер, местоположение и форму родинки.

Если записи ведутся на бумажном носителе, то в какой-то момент бумажных носителей станет несколько, поскольку в предыдущем может закончиться место для записей, бумажный носитель может прийти в негодность или человек может попросту его потерять. Если контроль осуществляется с помощью мобильного телефона, то фотографии со временем могут затеряться среди других, или человек может случайно удалить нужную фотографию, также анализ состояния родинок очень затруднен, так как обработка информации может занимать достаточно большое количество времени.

2.2 ОПИСАНИЕ ВОЗМОЖНОСТЕЙ ПОЛЬЗОВАТЕЛЕЙ

ПРИЛОЖЕНИЯ

Основной фактор риска развития рака кожи — УФ-излучение, которое повреждает ДНК клеток кожи. Очевидно, что пользователи приложения могут находиться в разных городах или даже странах. Поэтому необходимо иметь функционал выбора интересующего местоположения.

OpenWeatherMap — это онлайн-сервис, который предоставляет специальный API для получения информации о погоде. Для сбора и анализа

данных используются как официальные метеорологические службы, так и данные, полученные с метеостанций аэропортов и частных метеостанций во всем мире [7]. Информация обрабатывается OpenWeatherMap, после чего, на основе данных строится прогноз погоды и погодные карты, например карты облачности и осадков.

Каждому человеку важно знать свой фототип кожи, поскольку от него зависит сила влияния ультрафиолетового излучения, что в свою очередь приводит к возникновению различных опухолей, в том числе злокачественных. В зависимости от фототипа пользователя ему будут рекомендованы свои меры защиты от излучения. Для определения фототипа кожи будем использовать шкалу типов кожи по Фитцпатрику.

Родинки появляются на коже регулярно, это естественный процесс, которого не следует бояться. Однако, важно следить за их размером, цветом, формой и состоянием. Например, рекомендуется фотографировать невусы каждые полгода, чтобы отслеживать изменения и обсудить их со специалистом. Поэтому есть необходимость разработать отдельный функционал, который бы предлагал выбрать из галереи или сделать камерой смартфона фотографию подозрительной "родинки" и указать дополнительные данные о ее локализации на теле и типе кожи. По результатам самоанализа пользователь принимает решение о необходимости визита врачу.

Рак кожи занимает 3 место в мире по общей заболеваемости злокачественными опухолями и 2 место по онкологической заболеваемости у женщин [3]. Заболеваемость данной патологией растет. Важно, чтобы люди со всего мира отдавали себе отчет в опасности обгореть - на солнце или в солярии. Поэтому приложение будет поддерживать не только русский язык, но и английский.

Локализация приложений на разные языки важна по нескольким причинам. Она позволяет расширить аудиторию пользователей, давая возможность пользоваться приложением людям, говорящим на разных языках. Также это повышает уровень комфорта пользователей, так как они

могут пользоваться приложением на своём родном языке.

2.3 ПАТТЕРН ПРОЕКТИРОВАНИЯ MVP

Перед началом разработки следует определить функционал приложения. В зависимости от функционала можно выбрать паттерн проектирования, он определяет архитектуру приложения, задает его логику: на какие компоненты/модули будет делиться приложение и каким образом они взаимодействуют, и помогают разработчику понять, как устроен продукт внутри.

MVP имеет небольшое преимущество перед MVC в том, что он может разбить наше приложение на модули. Таким образом, мы можем избежать необходимости постоянно создавать представления. Другими словами, MVP может помочь сделать наши представления многократно [8].

С целью сокращения времени, затрачиваемого на разработку сложных программных продуктов, рекомендуется использовать готовые унифицированные решения. Это связано с тем, что стандартизация процессов упрощает коммуникацию между разработчиками, позволяет использовать общепринятые конструкции и значительно снижает число ошибок.

MVP является паттерном программирования графических интерфейсов. В этом паттерне приложение разбивается на три компонента:

- Model работает с данными, выполняет вычисления и управляет бизнес-процессами.
- View отображает пользовательский интерфейс и данные из модели.
- Presenter действует как посредник между моделью и видом.



Рисунок 2 – Схема паттерна MVP [8]

На схеме MVP (рис. 2) видно, что приложение, созданное с использованием MVP-архитектуры, работает посредством взаимодействия между моделью, видом и представителем. Процесс работы выглядит следующим образом:

- Вид строит интерфейс и добавляет в него данные из модели.
- Пользователь видит информацию и взаимодействует с интерфейсом.
- Вид перехватывает события и передает их представителю.
- Представитель обрабатывает данные и передает их модели.
- Модель выполняет операции и обновляется, изменяя свои свойства.
- Представитель получает обновленную модель и передает ее виду.
- Вид строит интерфейс с новыми данными.

Модель — это часть приложения, которая содержит бизнес-логику. Модель должна быть независима от других частей продукта, не иметь знаний об элементах дизайна и способах их отображения. Такой подход позволяет менять представление данных, их отображение, не затрагивая модель. Модель

обладает следующими особенностями:

- Модель — это бизнес-логика приложения.
- Модель не знает о других компонентах (контроллерах и представлениях).
- Для некоторых проектов модель — это слой данных (DAO, база данных, XML-файл).
- Для других проектов модель — это менеджер базы данных, набор объектов или просто логика приложения.

Представление отображает данные, полученные от модели, но не может напрямую влиять на модель. Таким образом, представление имеет доступ только на чтение к данным. Оно обладает следующими свойствами:

- Представление отображает данные, полученные от модели любым способом.
- В некоторых случаях представление может иметь код, который реализует бизнес-логику.

Представитель является прослойкой между видом и моделью. Он обрабатывает события, передаваемые видом, и при необходимости обращается к модели. Затем представитель возвращает данные на отрисовку виду [9].

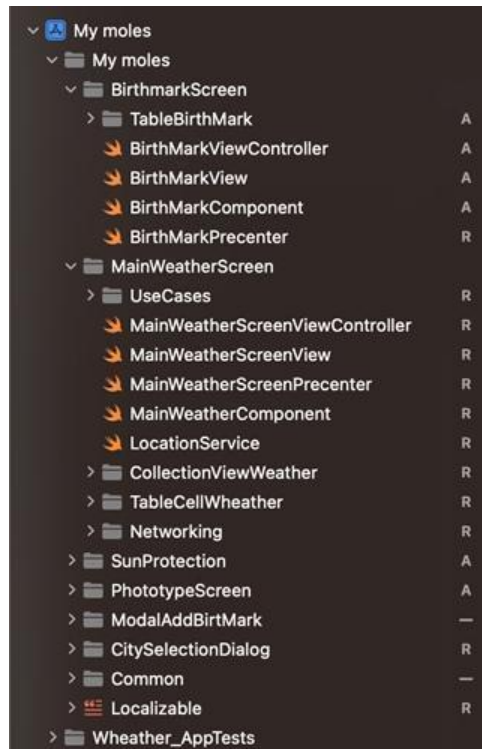


Рисунок 3 – Структура проекта

2.4 ЧИСТАЯ АРХИТЕКТУРА

Чистая архитектура представляет собой способ организации кода, который обеспечивает чёткое разделение ответственности между компонентами приложения. Приложение разбивается на независимые функциональные компоненты, которые взаимодействуют друг с другом определённым способом, передавая только те ресурсы, которые необходимы для выполнения задачи. Это позволяет уменьшить сложность каждого компонента, снизить вероятность ошибок и ускорить их исправление при обнаружении.

В чистой архитектуре сценарии приложения (use case) описываются отдельно, именно они определяют, какие внешние сервисы понадобятся в процессе работы приложения (рис. 4). Благодаря этому мы получаем больше свободы в выборе инструментов и можем приспособливать внешний мир под свои потребности, а не наоборот.



Рисунок 4 - Представление слоев чистой архитектуры [10]

Каждый круг изображает части ПО. Внешние круги описывают механизмы взаимодействия, а внутренние — правила взаимодействия. Названия секторов на иллюстрациях примерные и не обязательно соответствуют именам, используемым в конкретной системе.

По сути слои можно поделить следующим образом:

1. Сущности (предметная область и логика);
2. Сценарии использования (логика приложения);
3. Адаптеры интерфейсов (контроллеры, презентеры — всё, что помогает внешним фреймворкам взаимодействовать с приложением);
4. Внешние фреймворки (и/или устройства).

Чем ближе к центру круга, тем выше уровень описания системы. Таким образом, слой "сущности" полностью оторван от любой реализации системы и описывает предметную область такой, какой она есть [10].

2.5 ВНЕДРЕНИЕ ЗАВИСИМОСТЕЙ

Техника внедрения зависимостей (Dependency Injection, DI) широко используется в программировании как метод, способствующий разработке эффективных приложений. При использовании принципов DI вы создаете основу для хорошей архитектуры приложения, что приводит к следующим

преимуществам:

- возможность повторного использования кода;
- упрощение процесса рефакторинга;
- упрощение процесса тестирования.

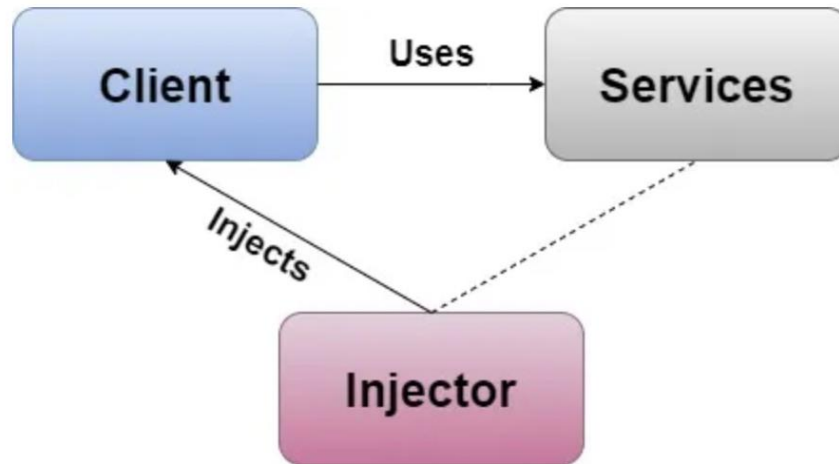


Рисунок 5 – Схема DI

Принимающий объект называется клиентом, а переданный (внедренный) объект называется службой. Код, который передает службу клиенту, может быть самым разным и называется инжектором. Вместо того, чтобы клиент указывал, какую службу он будет использовать, инжектор сообщает клиенту, какую службу использовать. «Внедрение» относится к передаче зависимости (службы) объекту (клиенту), который будет ее использовать [11]. Внедрение зависимостей для разрабатываемого приложения представлены в приложении 1.

2.6 ПАТТЕРН ПРОЕКТИРОВАНИЯ «ФАСАД»

Паттерны проектирования являются неотъемлемой частью программирования и необходимы для быстрого решения типовых задач. В процессе разработки постоянно встречаются одни и те же проблемы, и набор готовых шаблонов (паттернов) помогает их избежать.

Фасад - это паттерн проектирования, который позволяет снизить сложность и минимизировать зависимость между подсистемами,

предоставляя единый упрощенный интерфейс для набора подсистем. Фасад обеспечивает высокоуровневый интерфейс, который можно использовать для упрощения взаимодействия между подсистемами при проектировании сложных систем. Применение паттерна фасад позволяет уменьшить связность между компонентами, а также сократить объем передаваемой между ними информации.

Принцип декомпозиции является широко используемым при разработке сложных систем и заключается в разбиении системы на меньшие и более простые подсистемы. Высокая связность между модулями системы является очевидной проблемой, которую можно решить с помощью паттерна фасад. Применение паттерна для работы с сетью представлено в приложении 2.

В приложении есть слой, который занимается сетевыми вызовами. Для работы ему понадобится JSONDecoder, также в нем будет происходить обработка потоков. В реализуемом приложении выполняется два сетевых запроса на сервер погоды «OpenWeather». Использование паттерна «Фасад» для работы с сетевыми запросами позволит облегчит процесс тестирования, а также упростит реализацию запросов.

2.7 РАБОТА С СЕТЬЮ

Работа в сети является обязательным требованием для большинства современных приложений IOS.

OpenWeather - это онлайн сервис, предоставляющий API для доступа к различным данным о погоде, включая текущую погоду и прогнозы. Для получения информации он использует данные от официальных метеорологических служб, метеостанций в аэропортах и частных метеостанций, таким образом, обеспечивая точность и надежность данных. Сервис OpenWeather становится все более популярным среди разработчиков, так как он позволяет быстро получать актуальные данные о погоде в нужном формате и использовать их в своих приложениях и проектах.

В ходе разработки будем использовать данные на текущий день, а также

на 5 дней, они доступны в любой точке земного шара. Данные включают в себя данные прогноза погоды с 3-часовым шагом. Прогноз доступен в формате JSON или XML.

Сначала нам нужно создать модель, в которой мы будем хранить данные, декодируя данные ответа от API.

Эта модель будет соответствовать декодируемому протоколу. Поскольку нам не нужно что-то кодировать в API, мы создадим его только как декодируемый. Модель данных разрабатываемого приложения представлена в приложении 3.

Любые данные, которые мы видим в приложениях, которые мы используем, поступают из Интернета. Здесь на помощь приходят сеть и API. Этот обмен данными между устройством и сервером называется сетью. REST — это в основном тип архитектуры, который позволяет нам создавать службы и приложения, которые могут использоваться любым устройством или клиентом, понимающим HTTP. Например, в нашем случае мы делаем запрос на получение данных о погоде, и реализуемый запрос вернет нам данные в виде объекта JSON.

Для создания сетевого HTTP-запроса используется класс `URLSession`. Класс `URLSession` на самом деле является частью группы классов, которые работают вместе, чтобы создавать HTTP-запросы и отвечать на них.

`URLSession` включает ряд основных классов. Вот как работает среда:

1. Мы используем `URLSession` для создания сеанса. Сеанс можно представить, как открытую вкладку или окно в веб-браузере, которое объединяет множество HTTP-запросов при последующих посещениях веб-сайта.

2. `URLSession` используется для создания `URLSessionTask` экземпляров, которые могут извлекать данные и возвращать их в ваше приложение, а также загружать и загрузить файлы в веб-службы.

3. Мы настраиваем сеанс с `URLSessionConfiguration` объектом. Эта конфигурация управляет кэшированием, файлами cookie, подключением и

учетными данными.

4. Чтобы сделать запрос, мы создам задачу данных класса `URLSessionDataTask` и предоставляете ей URL-адрес и обработчик завершения. Это закрытие, которое выполняется, когда ответ на запрос возвращается вашему приложению.

5. Когда обработчик завершения выполняется, вы можете проверить возвращенные данные и предпринять соответствующие действия, например, загрузить данные в `TableView`, как мы будем делать с данными о погоде.

Следующим шагом для получения данных о погоде будет написание функции для выполнения сетевого запроса. Для большего понимания написанного кода следует пояснить некоторые конструкции языка и методы.

Замыкания — это автономные блоки функций, которые можно передавать и использовать в вашем коде. Это означает, что вы можете передавать функции в качестве аргумента другим функциям. Функция, которую вы передаете, может быть использована в какой-то момент функцией, которой она передана. Их также можно рассматривать как вложенные функции.

Обработчики завершения можно рассматривать как замыкания, которые выполняются после выполнения тела функции, т.е. всякий раз, когда завершается функция с обработчиком завершения (свойство «`isFinished`» функции изменяется на `true`), выполняется код обработчика завершения. Это очень полезно при работе с функциями, которые могут занять довольно много времени, например, с сетевыми вызовами.

`Async` означает «асинхронный». Этот термин можно рассматривать как атрибут метода, показывающий, что сам метод выполняет асинхронную работу. `Await` — это ключевое слово, которое будет использоваться для вызова асинхронных методов.

Асинхронность функции подразумевает под собой то, что переход на следующую операцию не осуществится до тех пор, пока не будет ответа от функции. При этом такое поведение не означает, что тред будет заблокирован

на период выполнения. Напротив, во время ожидания ответа он будет свободно выполнять другие действия.

Наконец мы можем вызвать реализованный метод запроса и получить данные в формате JSON, а далее конвертировать их в более удобный формат для дальнейшего отображения в таблице (приложение 4).

2.8 ХРАНЕНИЕ ДАННЫХ ПОЛЬЗОВАТЕЛЯ С ИСПОЛЬЗОВАНИЕМ CORE DATA

Для функционирования приложения необходимо хранить множество данных, например, фотографии, загруженные пользователем или его фототип. Чтобы помочь в сложной задаче управления и поиска сохраненных данных, Apple разработала целую структуру под названием Core Data. Core Data позволяет нам хранить данные на устройстве и снова делать их доступными даже после закрытия приложения.

Идея Core Data заключается в том, что ваши данные организованы и хранятся в базе данных с помощью так называемых схем. Схемы сообщают Core Data, как структурированы ваши данные, и могут быть разработаны с помощью визуального редактора, который является частью Xcode. На рисунке 6 показан визуальный редактор файла схемы Core Data, в котором вы можете создавать объекты. Сущность похожа на таблицу в базе данных, где вы можете определить столбцы и их типы данных.

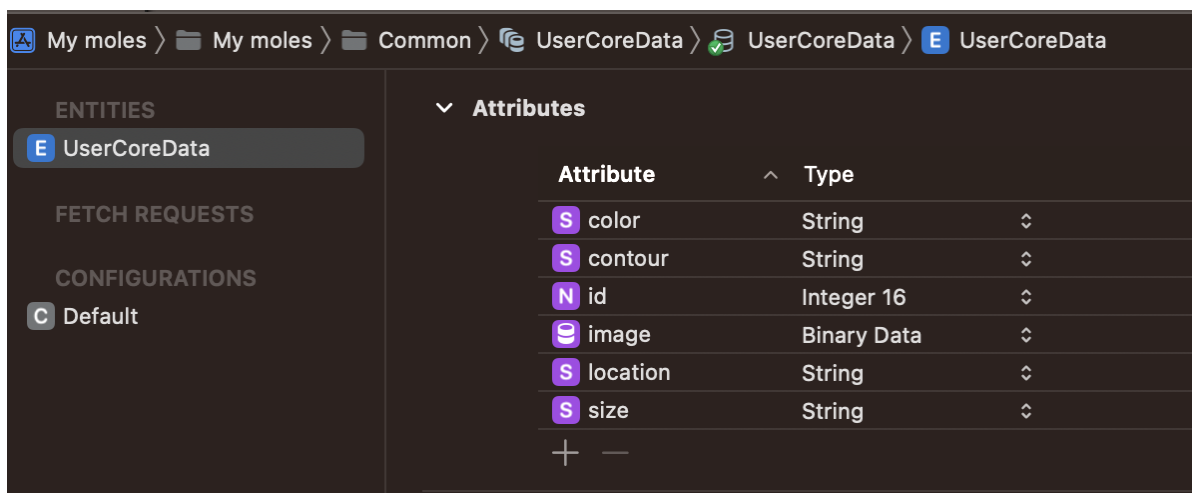


Рисунок 6 - Визуальный редактор файла схемы Core Data

Сущности (Entities) — таблицы, в которых будут «отображаться» наши Swift объекты,

Атрибуты (Attributes) — это колонки в таблице базы данных. В нашем swift - это соответствует свойствам (properties) объектов.

Взаимосвязи (Relationships) — это также свойства объектов, но они являются указателями на другие объекты в базе данных, или указателями на ряд других объектов.

Запросы в качестве свойств (Fetch properties) — это «вычисляемый» способ получить указатель на некоторые другие свойства. Если Взаимосвязи ссылаются напрямую на конечные объекты, то запросы ссылаются на объекты, выбранные указанным предикатом.

Core Data способен сортировать и фильтровать наши данные и может работать с гораздо большими данными — фактически нет ограничений на объем данных, которые он может хранить. Более того, Core Data реализует всевозможные расширенные функции, когда действительно нужно на них опереться: проверка данных, отложенная загрузка данных, отмена и повтор и многое другое (приложение 5).

Core Data описывает и управляет данными, хранящимися в приложении. Код на Objective-C или Swift может манипулировать данными для сохранения или чтения из приложения. Модель базы данных создается в Interface Builder, а Core Data организован в большие классы:

- Managed Object Context, который отвечает за взаимодействие с данными при сохранении или перезаписи.
- Persistent Store Coordinator управляет хранением данных.
- Managed Object Model определяет формат базы данных.
- Persistent Store представляет собой репозиторий, в котором хранятся данные.

Суть работы Core Data заключается в создании модели данных, а затем сохранении данных в модель с помощью метода `saveContext()`. Приложение не может работать без базы данных, поскольку данные будут потеряны при

выходе из него. Это и является основной задачей Core Data - обеспечение хранения данных.

Core Data может конвертировать данные в XML, бинарный код и SQLite для хранения. Он имеет стандартизированные схемы, а также существуют аналоги, такие как Realm и SQLite. Чтобы контролировать в процессе разработки сохраняемые данные используем приложение «DB Browser for SQLite» (Рисунок 7).

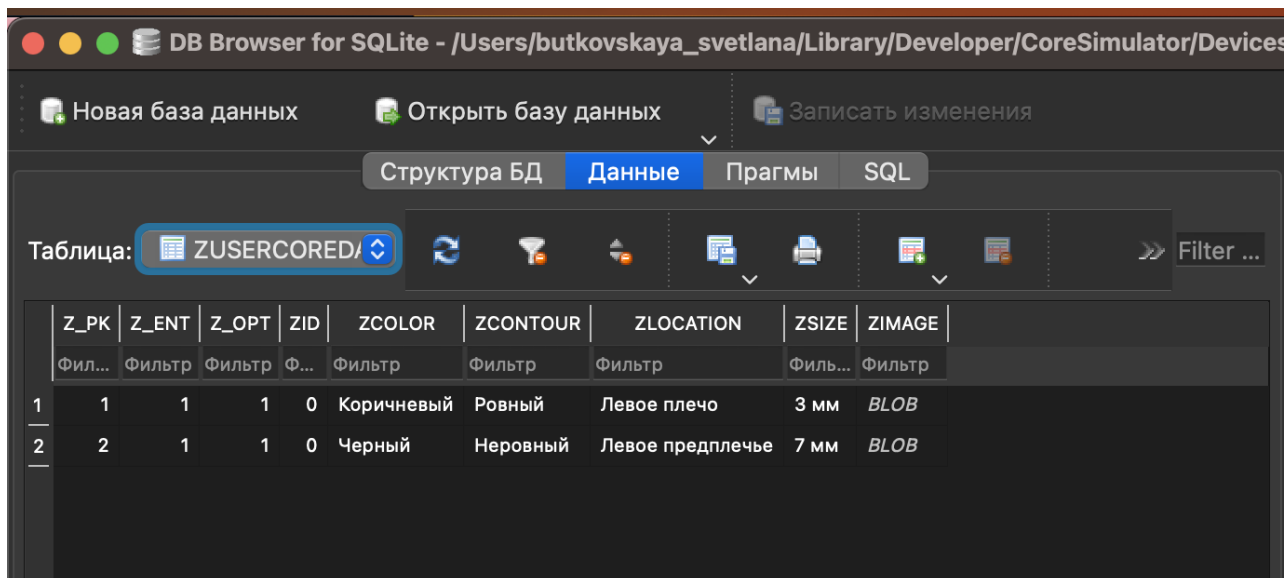


Рисунок 7 - Интерфейс программы «DB Browser for SQLite»

В предметной области приложения можно выделить 2 объекта.

User: Пользователь приложения, который добавляет информацию о невусах. У пользователя есть код, город и фототип.

Mole: Невус. У невуса есть размер, форма, местоположение, цвет, информация о том, кровоточит или нет.

Выполнив анализ предметной области, выделим сущности и укажем ключи сущностей, используемые для идентификации экземпляров сущности:

- Пользователь (Ключ - код пользователя)

Определим связи между указанными сущностями:

- Пользователь ИМЕЕТ невус

Степень связи ИМЕЕТ – 1:М (1 ко многим), поскольку у одного

пользователя может быть сразу несколько невусов. Класс принадлежности связи: Н-О, так как пользователь может не иметь невусов, но если невуc существует то он обязательно относится к какому-то пользователю.

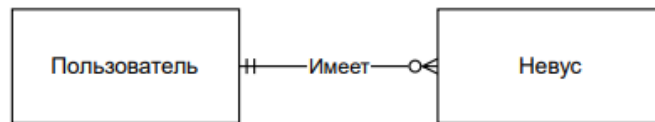


Рисунок 8 – Диаграмма инфологической модели

Проведём формирование набора предварительных отношений для каждой связи с указанием ключевых атрибутов таблиц. Первичные ключи для каждой таблицы выделены жирным шрифтом, внешние ключи выделены курсивом.

Связь ИМЕЕТ

- Пользователь (**Код пользователя (id)**)
- Невус (*Код пользователя (user_id)*)

Определим не ключевые атрибуты для каждой таблицы, укажем краткую характеристику каждого атрибута.

Таблица Пользователь (user) содержит следующие не ключевые атрибуты:

- Название города (city)
- Фототип (phototype)

Таблица Невус (mole) содержит следующие не ключевые атрибуты:

- Размер (size)
- Цвет (color)
- Местоположение (place)
- Форма (shape)
- Кровоточит или нет (bleeding)

Схема «сущность-связь» (также ERD или ER-диаграмма) — это разновидность блок-схемы, где показано, как разные «сущности» связаны между собой внутри системы.

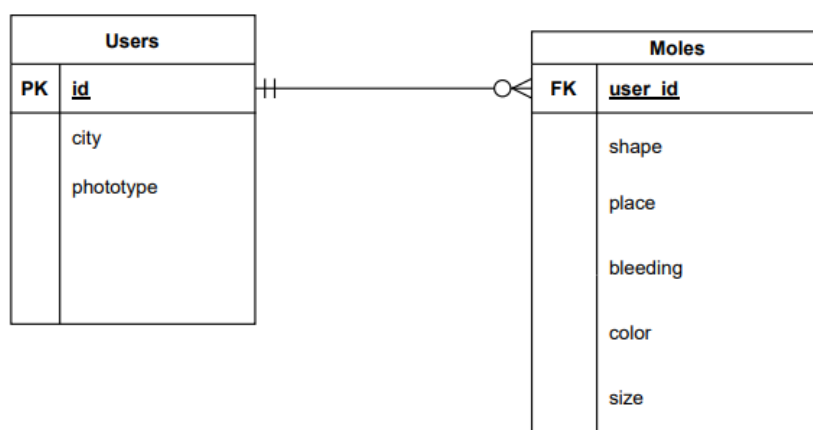


Рисунок 9 – Диаграмма связей таблиц в базе данных

Все отношения находятся в первой нормальной форме, поскольку все атрибуты отношений атомарны.

Все отношения приведены ко второй нормальной форме, потому что значение каждого не ключевого атрибута в каждом отношении можно однозначно определить только с помощью всего первичного ключа. Все таблицы автоматически приведены ко второй нормальной форме, так как у всех таблиц есть только простые ключи, состоящие из одного столбца.

Также можно сказать, что все таблицы в БД приведены к 3НФ, поскольку между не ключевыми атрибутами в таблице отсутствует функциональная зависимость.

Представим имеющуюся в виде ориентированного графа. Данный граф является ориентированным, так как на нем имеются направленные стрелки, указывающие на отдельные вершины графа. Каждая стрелка указывает направление движения от одной вершины к другой, что говорит о наличии определенной связи между ними. Таким образом, ориентированный граф позволяет задавать отношения, которые являются направленными, и если проверяется наличие связи между вершинами, то учитывается не только сам факт существования ребра, но и направление от одной вершины к другой.

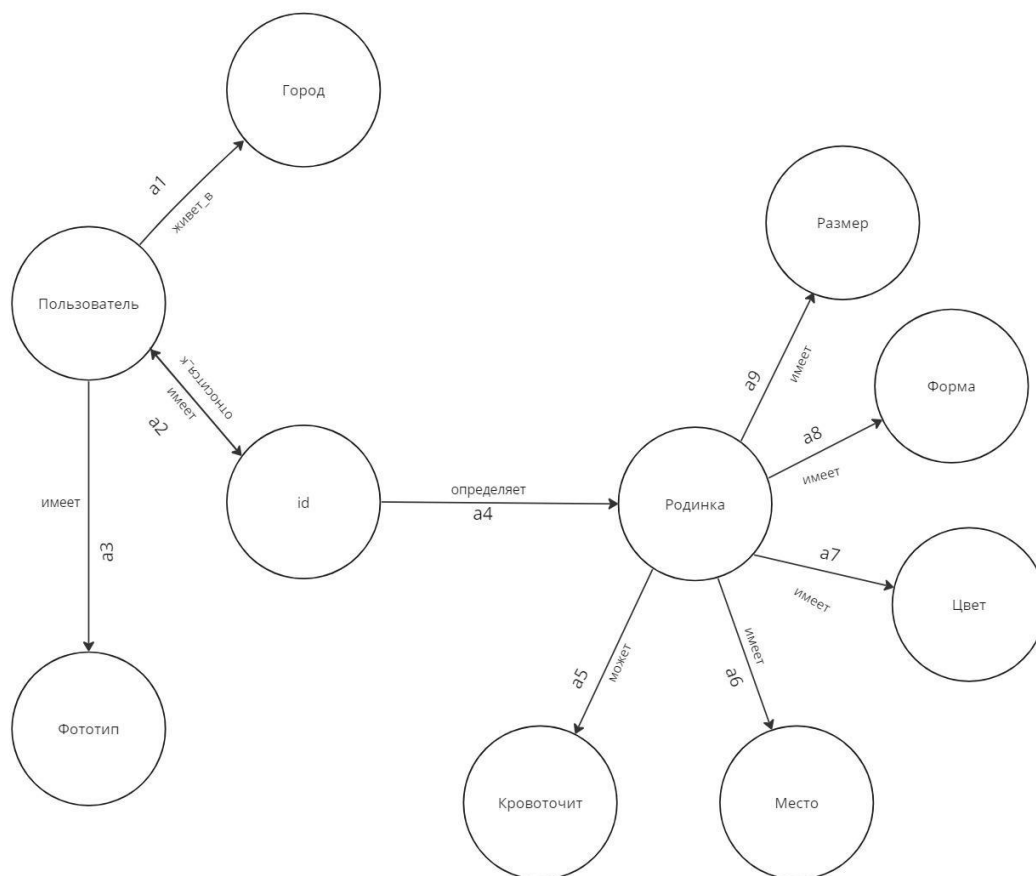


Рисунок 10 – представление базы данных в графовом виде

Теперь определим характеристики графа, составив матрицы смежности и инцидентности.

Таблица 4 - Матрица смежности

	Пользователь	Город	id	фототип	родинка	размер	форма	цвет	место	кровоточит
Пользователь	0	1	1	1	0	0	0	0	0	0
Город	0	0	0	0	0	0	0	0	0	0
id	0	0	0	0	1	0	0	0	0	0
фототип	0	0	0	0	0	0	0	0	0	0
родинка	0	0	0	0	0	1	1	1	1	1
размер	0	0	0	0	0	0	0	0	0	0
форма	0	0	0	0	0	0	0	0	0	0
цвет	0	0	0	0	0	0	0	0	0	0
место	0	0	0	0	0	0	0	0	0	0
кровоточит	0	0	0	0	0	0	0	0	0	0

Таблица 5 – Матрица инцидентности

	a1	a2	a3	a4	a5	a6	a7	a8	a9
Пользователь	-1	-1	-1	0	0	0	0	0	0
Город	1	0	0	0	0	0	0	0	0
id	0	1	0	-1	0	0	0	0	0

фототип	0	0	1	0	0	0	0	0	0
родинка	0	0	0	1	-1	-1	-1	-1	-1
размер	0	0	0	0	0	0	0	0	1
форма	0	0	0	0	0	0	0	1	0
цвет	0	0	0	0	0	0	1	0	0
место	0	0	0	0	0	1	0	0	0
кровоочит	0	0	0	0	1	0	0	0	0

Проверим граф на сильно-связность. Сильно связный граф - это ориентированный граф, в котором существует путь из каждой вершины в каждую другую вершину. Другими словами, для любых двух вершин U и V существует одновременно путь от U к V и от V к U .

Для проверки графа на сильно-связность рассмотрим ребро a_1 (рис. 10). Поскольку ребро a_1 имеет направление из вершины «Пользователь» в вершину «Город» и не имеет при этом обратного направления, делаем вывод, что граф не является сильно-связным.

Теперь найдем в графе гамильтоновы циклы. Если граф имеет простой цикл, содержащий все вершины графа по одному разу, то такой цикл называется гамильтоновым циклом, а граф называется гамильтоновым графом. Граф, который содержит простой путь, проходящий через каждую его вершину, называется полугамильтоновым. Это определение можно распространить на ориентированные графы, если путь считать ориентированным. Гамильтонов цикл не обязательно содержит все ребра графа. Ясно, что гамильтоновым может быть только связный граф. Поскольку в нашем случае граф не является связным, то в нем отсутствуют гамильтоновы циклы.

2.9 ДЕТЕКТИРОВАНИЕ ЗЛОКАЧЕСТВЕННЫХ И ДОБРОКАЧЕСТВЕННЫХ НОВООБРАЗОВАНИЙ ПРИ ПОМОЩИ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

В будущем в приложение планируется внедрить возможность предсказания для пользователя вероятности того, является ли его невос злокачественным или доброкачественным новообразованием.

Поскольку основной задачей детектирования является определение злокачественного новообразования с наибольшей точностью, за главную метрику будет взята полнота (recall). Для обучения нейронной сети было взято 2640 изображений злокачественных и доброкачественных образований. В качестве тестового набора данных взято 660 изображений злокачественных и доброкачественных образований. Соотношение одного класса к другому в обеих выборках примерно одинаковое. Архитектура нейронной сети, организованная при помощи библиотеки Keras, выглядит следующим образом:

- аугментация данных для увеличения датасета;
- два сверточных слоя с выходом 16 каналов, фильтр размером 3x3
- сверточный слой MaxPooling;
- два сверточных слоя с выходом 32 канала, фильтр размером 5x5;
- сверточный слой MaxPooling;
- два сверточных слоя с выходом 16 каналов, фильтр размером 3x3;
- сверточный слой MaxPooling;
- векторизация при помощи Flatten;
- Полносвязный слой, принимающий на вход вектор из 128 значений;
- Полносвязный слой с сигмоидой в качестве функции активации.

В качестве функционала ошибки используется бинарная кросс-энтропия для избежания затухания градиента на концах сигмоиды, оптимизатор был выбран ADAM. Рассматриваемые метрики: recall, precision, accuracy.

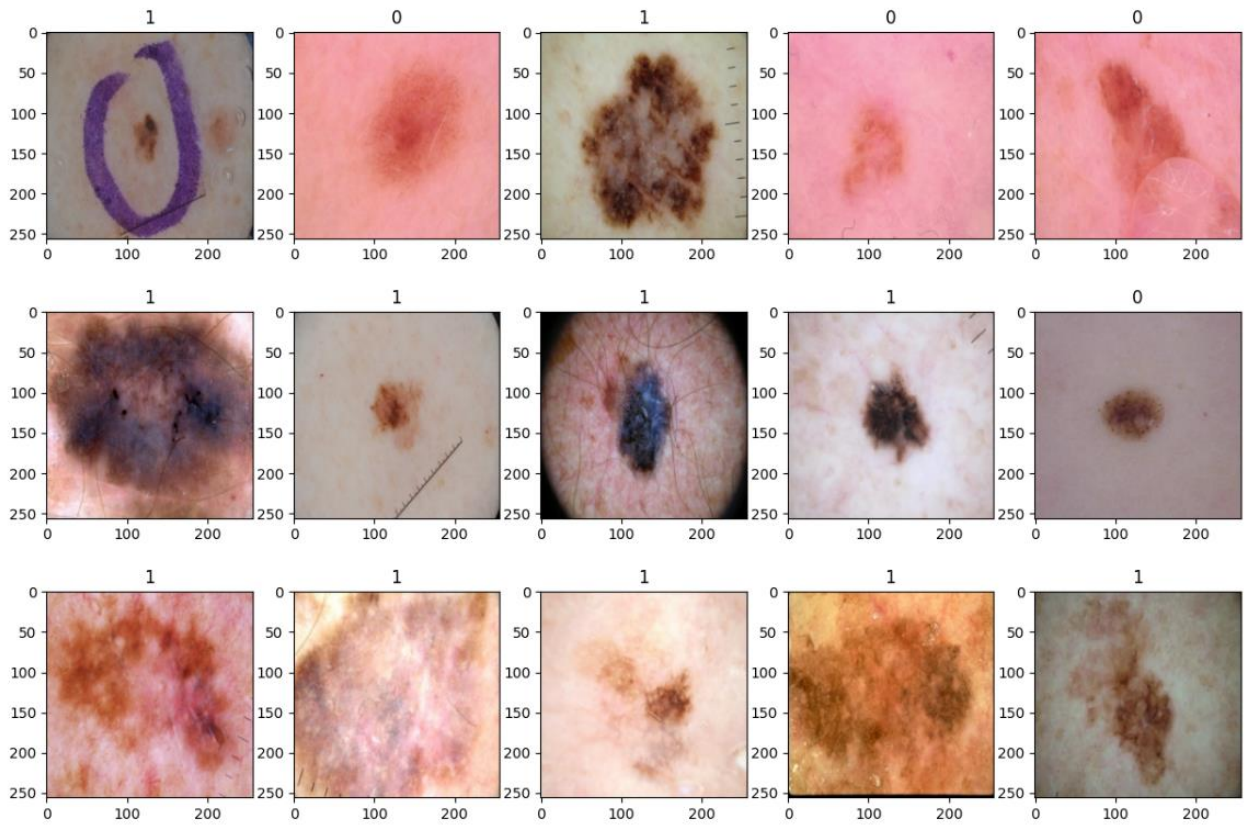


Рисунок 11 - пример 15 изображений невусов из тренировочной выборки (0 - доброкачественное образование, 1 - злокачественное образование)

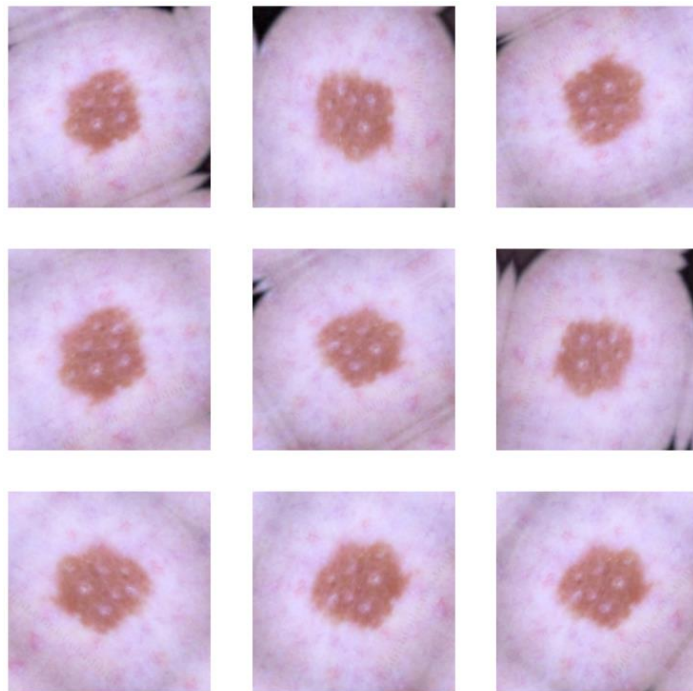


Рисунок 12 - аугментация невуса из тренировочной выборки

Фрагмент кода, отвечающий за вывод информации о новообразовании из тестовой выборки. Тестовая выборка использовалась в качестве валидационной.

```
img, label = batch[0][15], batch[1][15]
plt.imshow(img)
if label==1:
    plt.title("Malignant")
else:
    plt.title("Benign")
plt.show()
```

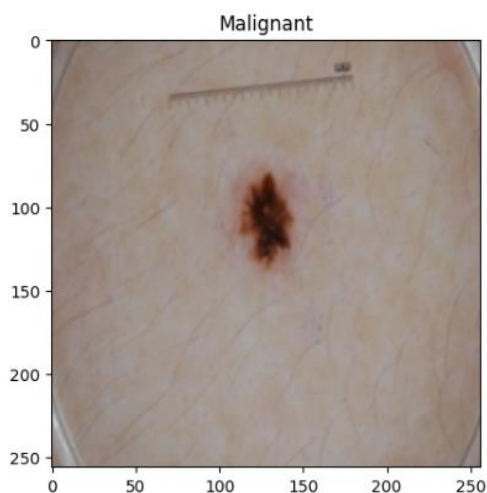


Рисунок 13 – результат предсказания модели для одного из экземпляров тестовой выборки

Для данного новообразования уверенность модели в том, что оно является злокачественным, составляет 91,6 %. Полноценный код, написанный на языке Python, представлен в приложении 6.

2.10 ВЕРСТКА

Большинство iOS-проектов в настоящее время постепенно начинают с удаления Storyboard и Xib. Разработчики предпочитают управлять пользовательским интерфейсом и макетами, записывая ограничения программно. Это стало одним из лучших решений, позволяющих избежать слияния конфликтов, особенно при работе в команде. Поэтому отрисовывать элементы приложения будем с помощью SnapKit.

SnapKit — это мощная библиотека автоматической компоновки с минимальным объемом кода, позволяющая свести к минимуму работу разработчиков по управлению ограничениями. Она является типобезопасным по своей конструкции и, следовательно, помогает уменьшить количество ошибок при создании надежных приложений (приложение 7).

Также в процессе разработки будем использовать R.Swift. R.swift — это инструмент для получения строго типизированных ресурсов с автозаполнением, таких как изображения, шрифты и переходы в проектах Swift. R.Swift помогает разработчики избежать ряда проблем и имеет ряд преимуществ:

- никогда больше не вводятся строковые идентификаторы;
- поддерживает изображения, шрифты, раскадровки, переходы, идентификаторы повторного использования и многое другое;
- проверки и ошибки во время компиляции вместо сбоев во время выполнения.

```
findPhototypeView.setTitleColor(R.color.white(), for: .normal)
R.string.localizable.feelsLike()
R.image.sunHeader().withRenderingMode(.alwaysOriginal)
labelTextWheather.font = R.font.manropeMedium(size: 14)
```

Рисунок 14 - Примеры использования R.Swift

2.11 ТЕСТИРОВАНИЕ

При написании тестов учитываются все возможные варианты и ситуации, которые могут произойти в приложении. Это помогает тестировать все эти подходы и исключать возможные ошибки. В момент написания нового функционала проще всего запустить тесты и убедиться в том, что старый функционал по-прежнему отрабатывает корректно. Этим можно сэкономить время в будущем, особенно если проект амбициозный и с долгосрочными планами. Модульное тестирование (Unit test) - это короткие тесты, которые проверяют работу конкретного куска кода. Цель состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет быстро проверить, не привело ли очередное изменение кода к появлению ошибок в уже протестированных местах программы, а также облегчает обнаружение и устранение таких ошибок [12].

Аббревиатура FIRST описывает критерии для эффективных unit тестов. Они включают следующее: быстрота (Fast), независимость/изолированность (Independent/Isolated), повторяемость (Repeatable), самостоятельная проверка (Self-validating) и своевременность (Timely). Для юнит-тестирования используются тестовые фреймворки - они позволяют имитировать функции. В коде больших проектов много зависимостей, и функции вызывают друг друга, влияя на разные части программы. Но, как правило, достаточно проверить функции «в вакууме», отдельно от остального кода.

Unit tests облегчают работу программиста и позволяют находить ошибки, которые могут быть не замечены в процессе работы приложения. Они также позволяют документировать код и упрощают работу с ним другим разработчикам. Если не понятно, как работает та или иная функция, можно перейти к коду юнит-теста. По нему сразу становится понятно, какие параметры принимает функция и что она возвращает после выполнения. Это упрощает жизнь тем, кто работает со сложным кодом. Кроме того, юнит-тесты могут помочь избежать ошибок при рефакторинге и упрощают в целом

процесс разработки. Примеры тестирования методов разрабатываемого приложения предоставлены в приложении 8.

2.12 ОПИСАНИЕ И ДЕМОНСТРАЦИЯ ФУНКЦИОНАЛА РАЗРАБОТАННОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ

По вышеописанным алгоритмам, разработано мобильное приложение под OS IOS и заполнена тестовыми данными база данных. В соответствии с заявленными возможностями, которые должно предоставлять мобильное приложение проводится демонстрация возможностей пользователя.

2.13 ДЕМОНСТРАЦИЯ ВОЗМОЖНОСТЕЙ ПОЛЬЗОВАТЕЛЕЙ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Каждый пользователь приложения может выполнять следующие действия:

- изменить язык пользовательского интерфейса (с русского на английский и обратно);
- посмотреть погоду на текущий и следующие 5 дней в любом городе мира;
- выбрать свой фототип;
- загружать в приложение фотографии родинок с их описанием для самодиагностики;
- получить рекомендации по защите кожи от UV лучей в выбранном городе с учетом фототипа пользователя.

2.13.1 ИЗМЕНЕНИЕ ЯЗЫКА ИНТЕРФЕЙСА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Мобильное приложение позволяет пользователю просматривать информацию на двух языках: на русском и английском. Это относится к тексту на кнопках, элементах меню и прочим надписям в мобильном приложении (рис. 12). Язык выбирается автоматически в зависимости от языка установленного на самом устройстве пользователя (русский или английский). Если у пользователя стоит другой язык, то по умолчанию используется версия с английским языком.

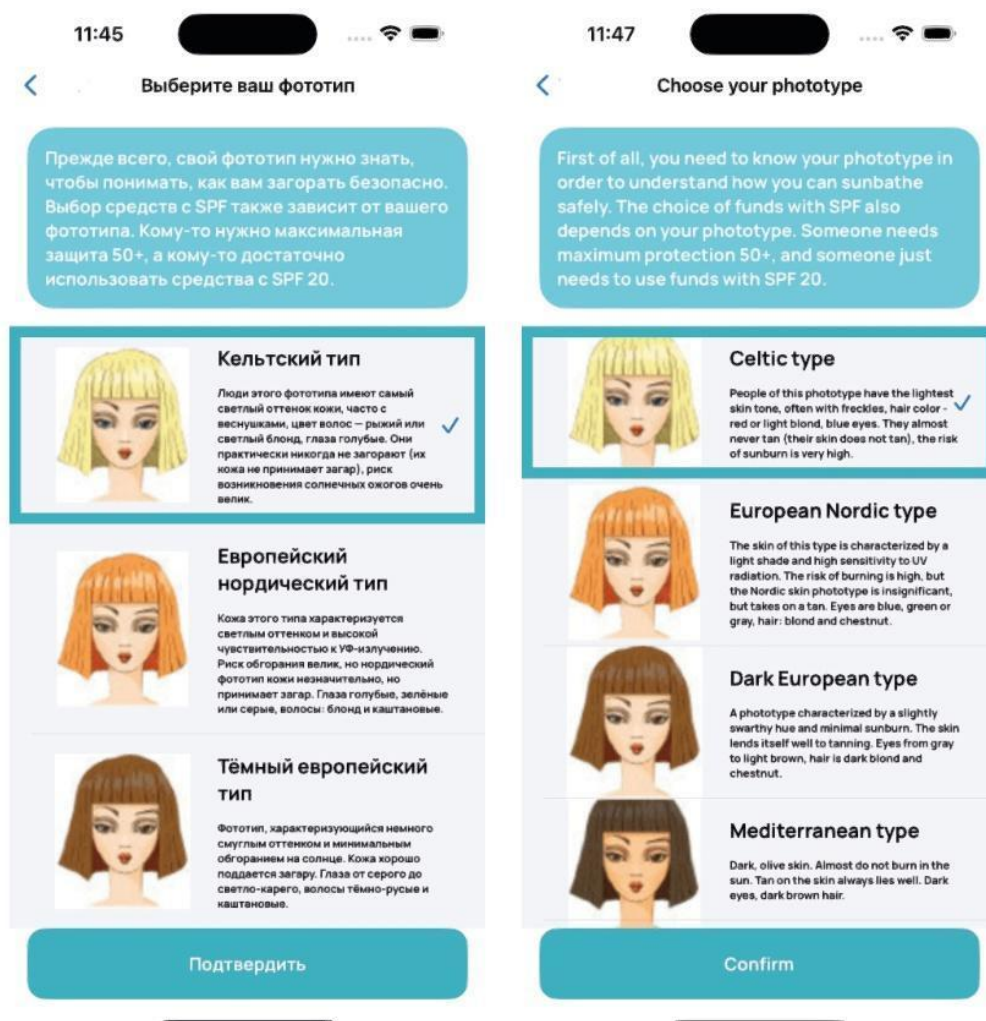


Рисунок 15 - Демонстрация смены языка на экране выбора фототип

Как видно на приведенных выше рисунках, при смене на соответствующий язык текст на кнопках, а также текст элементов меняет свой перевод в соответствии с языком.

2.13.2 ЭКРАН ПРОСМОТРА ПОГОДЫ

Прогноз погоды - это предсказание предстоящего состояния погоды, основанное на научном анализе её изменений. Прогноз погоды необходим для планирования работы аэропортов, производства сельхозпродукции и конце концов для организации работ коммунальных хозяйств. Также прогноз погоды нужен в первую очередь тем профессиям, от которых зависят жизни людей. Это диспетчер полетов авиакомпаний, пилоты самолетов, рыбаки, спасатели, агрономы, представители экстремальных профессий, водители особенно дальних перевозок, военные. Также прогноз погоды необходим метеочувствительным людям. В нашем приложении пользователь может посмотреть прогноз на текущий день и на следующие пять дней с трехчасовым интервалом (рис. 13).

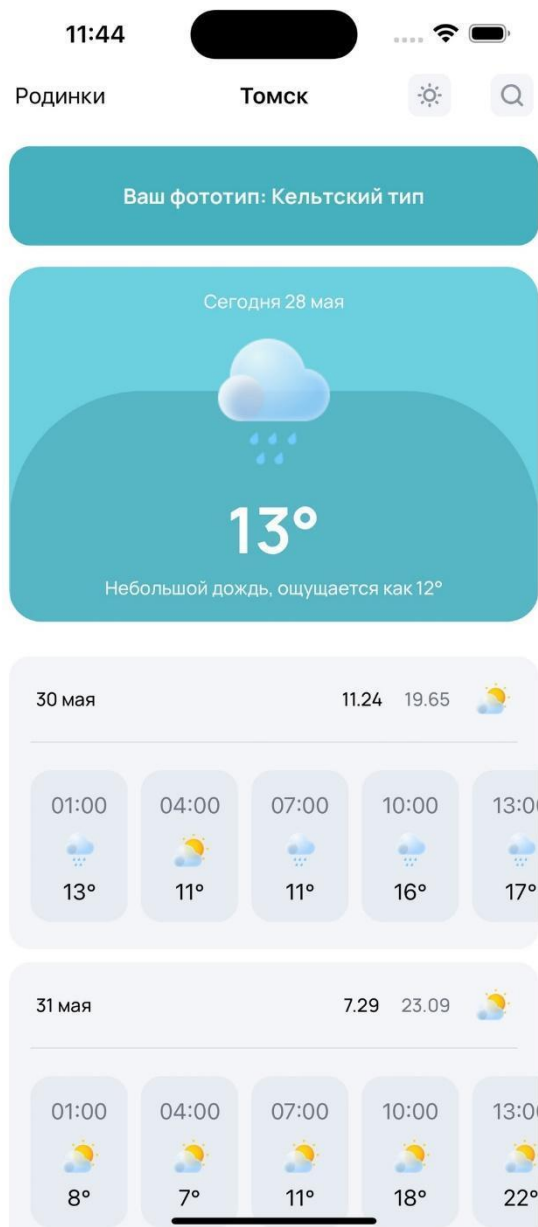


Рисунок 16 - Экран просмотра погоды

2.13.3 ЭКРАН ВЫБОРА ФОТОТИПА

Знание своего фототипа кожи важно для правильной оценки склонности к ультрафиолетовому излучению и для выбора соответствующих мер защиты от вредного воздействия света на кожу. Поэтому был разработан функционал, с помощью которого пользователь может определить свой фототип (рис. 14). Знание фототипа пользователя поможет приложению выбрать оптимальный защитный фактор солнцезащитного средства, определить продолжительность пребывания на солнце, определить необходимость использования

специализированной защитной одежды, шляп и очков.

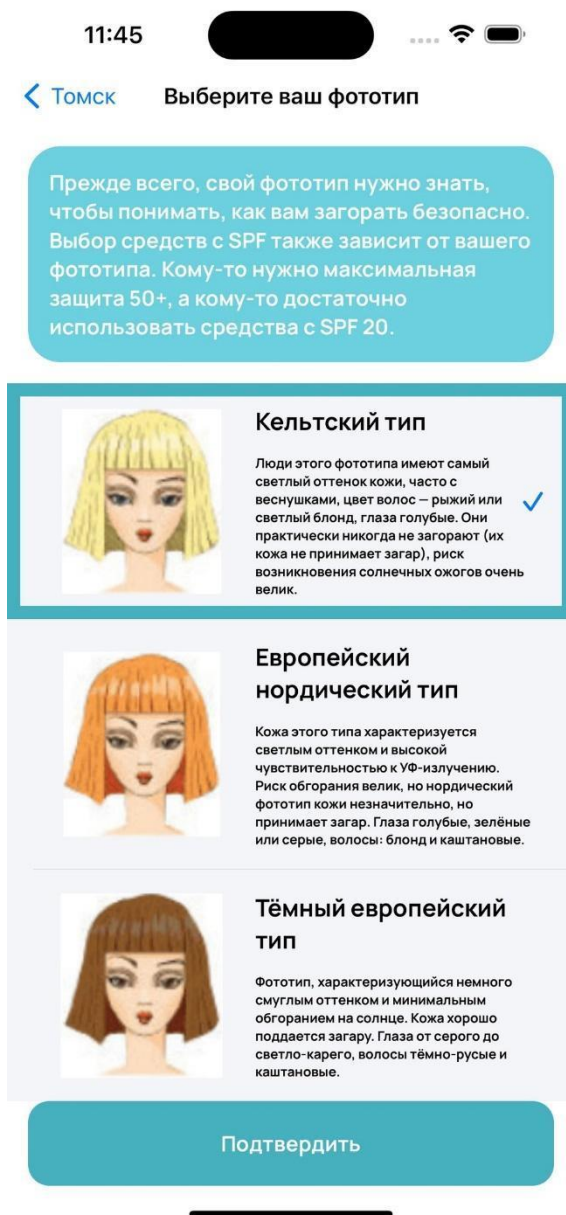


Рисунок 17 - Экран выбора фототипа

2.13.4 ЭКРАН ДЛЯ ДИНАМИЧЕСКОГО НАБЛЮДЕНИЯ ЗА РОДИНКАМИ И ИХ САМОДИАГНОСТИКИ

Следить за своими родинками очень важно в связи с возможностью развития рака кожи. Родинки могут менять свой цвет, форму, размер, структуру или выступать - это могут быть признаки того, что в их области началось злокачественное развитие клеток. Поэтому был разработан функционал, с помощью которого пользователь может отслеживать состояние родинок и проводить самодиагностику (рис. 15).

Следовательно, следить за своими родинками имеет решающее значение для раннего выявления меланомы и других видов рака кожи. Хотя не все родинки являются раковыми, их контроль поможет идентифицировать факторы риска или симптомы ранних стадий рака кожи, что повышает шансы на успешное вмешательство и излечение.

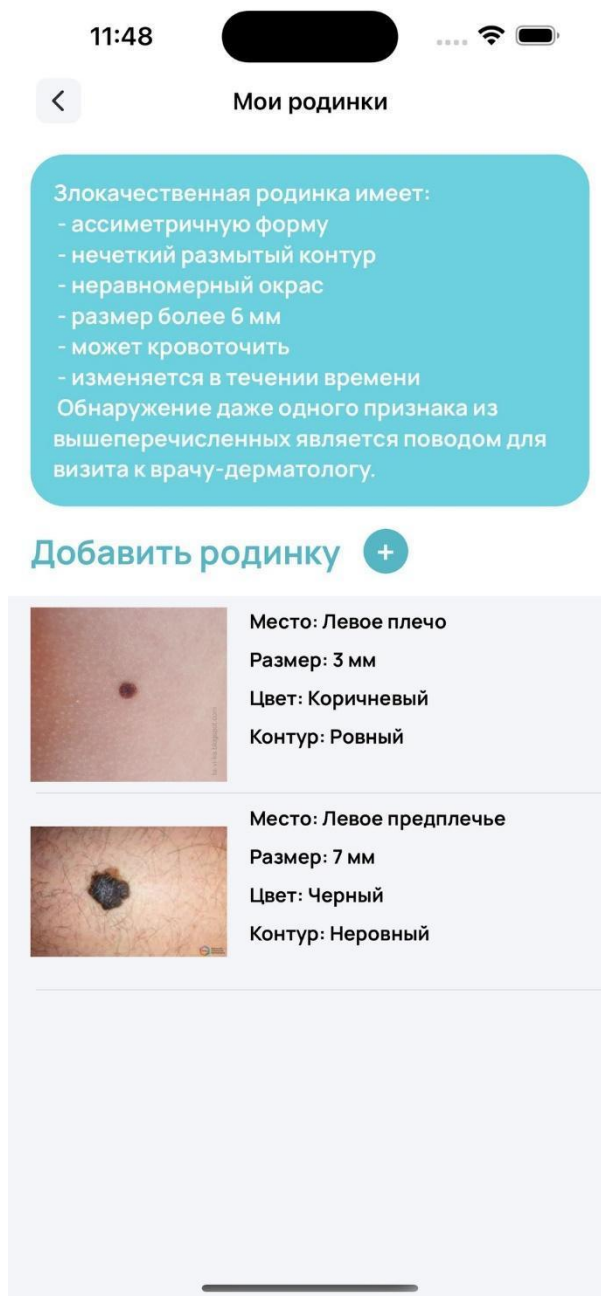


Рисунок 18 - Экран для динамического отслеживания изменений родинок и из самодиагностики

2.13.5 ЭКРАН ДЛЯ ПОЛУЧЕНИЯ РЕКОМЕНДАЦИЙ ПО ЗАЩИТЕ КОЖИ ОТ UV ЛУЧЕЙ

Ультрафиолетовые (УФ) лучи, которые относятся к части спектра солнечного света, могут привести к различным проблемам кожи, включая солнечные ожоги, преждевременное старение, эритему (покраснение кожи) и повышенный риск развития различных видов рака кожи. Поэтому был

разработан функционал, с помощью которого пользователь может получить рекомендации по защите сформированные по значению УФ-индекса и фототипа пользователя (рис. 16).

Защита от солнечных лучей может быть достигнута несколькими способами, такими как использование солнцезащитных кремов с более высоким фактором защиты, также можно использовать защитную одежду, головные уборы и очки. Благодаря этим методам УФ лучи не оказывают пагубного воздействия на кожу.

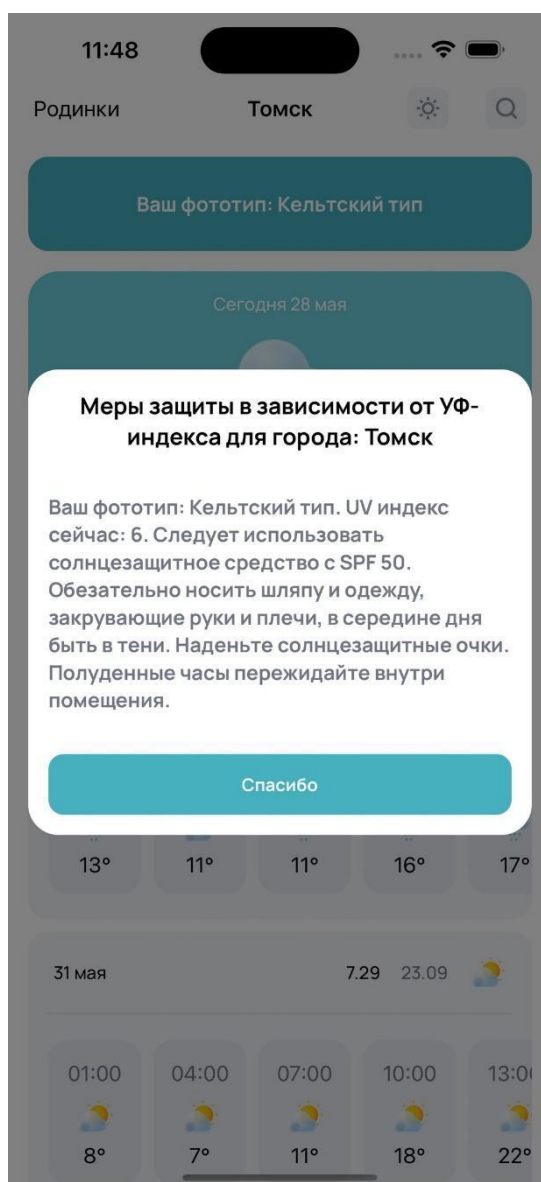


Рисунок 19 - Экран для получения рекомендаций по защите кожи от UV лучей

2.14 ВЫВОДЫ ПО ГЛАВЕ

В данной главе подробно рассмотрены все разработанные модули в мобильном приложении, которые обеспечивает пользователям возможность мониторинга изменения невусов, контроль внешних признаков невусов для самодиагностики. Предоставление пользователю информации о погоде. Возможность пользователю выбрать нужный фототип кожи и получить информацию о необходимых мерах защиты в выбранном городе в зависимости от значения УФ-индекса.

ГЛАВА 3. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

3.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.

Цель экономического раздела – провести детальный анализ проекта по критериям конкурентоспособности и ресурсоэффективности. Количество злокачественных заболеваний кожи непрерывно растет. Многие злокачественные заболевания кожи тем лучше поддаются лечению, чем раньше они обнаружены. Сам человек не всегда может вовремя распознать грозящую ему опасность. Обращение к квалифицированному дерматологу могут себе позволить не все граждане из-за своего финансового положения.

В связи с этим было решено разработать мобильное приложение, которое бы позволяло выбрать из галереи или сделать камерой смартфона фотографию подозрительной "родинки" и указать дополнительные данные о ее локализации на теле и типе кожи. По результатам анализа формулируется рекомендация пользователю о необходимости визита к врачу.

3.1.1 Потенциальные потребители результатов исследования

В ходе ВКР были проведены исследования для получения оценки потребности рынка в приложении для самодиагностики невусов. Для того, чтобы определить потенциальных потребителей, необходимо определить целевой рынок и произвести его сегментирование. Целевым рынком являются медицинские организации, заинтересованные в своевременной диагностике симптомов и грамотном выстраивании лечения своих пациентов. Конечными потребителями являются люди, следящие за состоянием своего здоровья в целом, а также те, кто проживает в местности с большим значением УФ-индекса или склонные к дерматологическим заболеваниям.

3.1.2 Анализ конкурентных технических решений

Позиция разработки и конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum V_i \cdot B_i, \quad (1)$$

где K – конкурентоспособность научной разработки или конкурента;

V_i – вес показателя (в долях единицы);

B_i – балл i -го показателя.

В таблице 6 приведена оценочная карта, включающая конкурентные технические разработки в области производства разрабатываемого приложения.

B_0 – продукт проведенной работы;

B_1 MoleScore – приложение для самодиагностики невусов;

B_2 UMSkinCheck.

По результатам проведенного анализа видно, что разрабатываемое приложение для самодиагностики невусов по выделенным критериям способно конкурировать и может найти свое место на рынке подобных приложений и быть востребованным.

Таблица 6 - Оценочная карта для сравнения конкурентных технических разработок

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		B_0	B_1	B_2	B_0	B_1	B_2
1	2	3	4	5	6	7	8
Технические критерии обогащаемого материала							
1. Получение личных рекомендаций, исходя из прогноза погоды с учетом фототипа кожи человека	0,3	5	3	3	1,5	0,9	0,9

2. Площадь поверхности кожи для анализа	0,1	3	3	3	0,3	0,3	0,3
3. Количество параметров оценки "родинки"	0,1	4	3	4	0,4	0,3	0,4
4. Требуется отдельного устройства, присоединяющегося к камере	0,1	5	5	3	0,5	0,5	0,3
5. Получение прогноза погоды и значения УФ-индекса	0,1	5	2	2	0,5	0,2	0,2
Экономические критерии оценки эффективности							
8. Конкурентоспособность продукта	0,2	4	4	5	0,8	0,8	1
9. Финансирование научной разработки	0,1	5	5	5	0,5	0,5	0,5
Итого:	1				4,5	3,5	3,6

3.1.3 SWOT – анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Он проводится в несколько этапов.

Таблица 7 - SWOT-анализ

	<p>Сильные стороны научно-исследовательского проекта:</p> <p>С1 Простой и понятный пользовательский интерфейс</p> <p>С2 Приложение подстраивается под фототип человека по Фитцпатрику</p> <p>С3 Приложение является бесплатным для установки</p>	<p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл1 Отсутствие многих функций</p> <p>Сл2 Отсутствие медицинской сертификации</p> <p>Сл3 Отсутствие бюджетного финансирования.</p>
<p>Возможности:</p> <p>В1 Использование инновационной инфраструктуры ТПУ</p> <p>В2 Высокий спрос среди пользователей смартфонов, желающих отслеживать динамику изменения невусов</p> <p>В3</p>	<p>Простота и интуитивно понятный интерфейс приложения приведут к высокому спросу среди людей, которые заинтересованы в мониторинге состояния своего здоровья, имея при себе всего лишь телефон с доступом в Интернет.</p>	<p>Использование опыта и инфраструктуры ТПУ может помочь в кратчайшие сроки пройти испытания и получить государственную сертификацию, что повысит спрос на продукт несмотря на слабость функционала по сравнению с конкурентами.</p> <p>Использование инфраструктуры ТПУ может дать необходимое</p>

		финансирование для проекта.
Угрозы: У1 Недостаточное финансирование проекта У2 Развитая конкуренция У3 Сложность в получении медицинской сертификации	Проста реализации приложения и возможность бесплатной установки позволят преодолеть барьер конкуренции и занять свое место на рынке.	Необходимо получить финансирование на разработку прототипа и конечного продукта. Необходимо сертифицировать приложение для того, чтобы имелась возможность постановки предварительного диагноза через приложение

3.2 Планирование научно-исследовательских работ

3.2.1 Структура работ в рамках научного исследования

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

Для выполнения научных исследований формируется рабочая группа, в чей состав входят: бакалавр, научный руководитель, консультант по части социальной ответственности (СО) и консультант по экономической части (ЭЧ) выпускной квалификационной работы. Составим перечень этапов и работ в рамках проведения научного исследования и проведем распределение исполнителей по видам работ (табл. 8).

Таблица 8 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Выбор темы ВКР	1	Постановка задачи	Научный руководитель, инженер
Разработка технического задания	2	Составление и утверждение технического задания	Научный руководитель,

			консультант ЭЧ, СО, инженер
	3	Календарное планирование работ по теме	Научный руководитель, инженер
Подбор и изучение материалов по теме	4	Изучение медицинских основ диагностики невусов	Инженер
	5	Обзор существующих приложений для самодиагностики невусов	Инженер
	6	Изучение принципов реализации мобильных приложений	Инженер
	7	Изучение основ языка Swift	Инженер
Теоретические исследования	8	Архитектура современных мобильных приложений на IOS	Инженер
	9	Изучение влияния УФ-индекса на состояние кожи человека	Инженер

Обобщение и оценка результатов исследования	10	Выбор наилучшей концепции приложения	Научный руководитель, бакалавр
	11	Определение целесообразности проведения НИР	Научный руководитель, инженер
Проведение НИР			
Разработка приложения	12	Изучение методов и технологий для решения задач	Научный руководитель, инженер
	13	Выбор оптимальных методов решений	Научный руководитель, инженер
	14	Выбор архитектуры	Инженер
	15	Вёрстка приложения	Инженер
Доработка приложения, исправление ошибок и тестирование	16	Подключение базы данных	Инженер
	17	Настройка логики между интерфейсами и написание методов	Научный руководитель, инженер
	18	Тестирование	Научный руководитель, инженер
Оформление комплекта документации по НИР	19	Оценка эффективности производства и применения разработки	Инженер, консультант по ЭЧ
	20	Разработка социальной ответственности по теме	Инженер, консультант СО

	21	Составление пояснительной записки	Инженер
--	----	-----------------------------------	---------

3.2.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожі}$ используется следующая формула (2):

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5} \quad (2)$$

Где: $t_{ожі}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\min i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{\max i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ несколькими исполнителями:

$$T_{pi} = \frac{t_{ожі}}{ч_i}, \quad (3)$$

где T_{pi} – продолжительность одной работы, раб.дн.;

$t_{ожі}$ – ожидаемая трудоемкость выполнения одной работы, чел. – дн;

$ч_i$ – численность исполнителей, выполняющих одновременно одну и

ту же работу на данном этапе, чел.

3.2.3 Разработка графика проведения научного исследования

При выполнении дипломных работ студенты в основном становятся участниками сравнительно небольших по объему научных тем. Поэтому наиболее удобным и наглядным является построение ленточного графика проведения научных работ в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (4)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле (5):

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}, \quad (5)$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

Таким образом:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}} = \frac{365}{365 - 96 - 14} = 1,48 \quad (6)$$

Все рассчитанные значения необходимо свести в таблицу 9.
и построить диаграмму Ганта (табл. 10).

Таблица 9 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях T_{pi}	Длительность работ в календарных днях T_{ki}
	t_{min} , чел-дни		t_{max} , чел-дни		$t_{ожг}$, чел-дни			
	И с п . 1	И с п . 2	И с п . 1	И с п . 2	И с п . 1	И с п . 2		
1	2	3	4	5	6	7	8	9
1. Постановка задачи	2	2	3	3	2,4	2,4	2,4	3
2. Составление и утверждение технического задания	2	-	4	-	2,8	-	2,8	4
3. Календарное планирование работ по теме	1	1	2	2	1,4	1,4	1,4	2
4. Изучение медицинских основ диагностики невусов	-	2	-	4	-	2,8	2,8	4
5. Обзор существующих приложений для самодиагностики невусов	-	4	-	6	-	4,8	4,8	7
6. Изучение принципов реализации мобильных приложений	-	5	-	7	-	5,8	5,8	8
7. Изучение основ языка Swift	-	5	-	7	-	5,8	5,8	8
8. Архитектура современных мобильных приложений на IOS	-	2	-	3	-	2,4	2,4	3
9. Изучение влияния УФ-индекса на состояние кожи человека	-	2	-	3	-	2,4	2,4	3
10. Выбор наилучшей концепции приложения	3	3	4	4	3,4	3,4	3,4	5

11. Определение целесообразности проведения ВКР	2	2	3	3	2,4	2,4	2,4	3
12. Изучение методов и технологий для решения задач	-	3	-	5	-	3,8	3,8	12
13. Выбор оптимальных методов решений	-	5	-	8	-	6,2	6,2	9
14. Выбор архитектуры	-	3	-	6	-	4,2	4,2	12
15. Вёрстка приложения	-	5	-	8	-	6,2	6,2	9
16. Подключение базы данных	-	2	-	4	-	2,8	2,8	4
17. Настройка логики между интерфейсами и написание методов	-	6	-	9	-	7,2	7,2	10
18. Тестирование	-	3	-	5	-	3,8	3,8	5
19. Оценка эффективности производства и применения разработки	-	3	-	5	-	3,8	3,8	5
20. Разработка социальной ответственности по теме	-	4	-	5	-	4,4	4,4	10
21. Составление пояснительной записки	-	4	-	5	-	4,4	4,4	6
Итого:	10	76	16	107	12,4	94	83,2	123

Примечание: Исп. 1 – научный руководитель, Исп. 2 – инженер.


Таблица 10 – Диаграмма Ганта

№	Исп	T_{ki} , кал. дн.	Продолжительность работ																			
			Февраль				Март					Апрель				Май						
			1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4	5		
1	Исп1	3	■																			
	Исп2	3	■																			
2	Исп1	4		■																		
	Исп2	4		■																		
3	Исп1	2			■																	
	Исп2	2			■																	
4	Исп2	4			■																	
5	Исп2	7			■	■																
6	Исп2	8				■	■															
7	Исп2	8					■	■														
8	Исп2	3						■														
9	Исп2	3							■													
10	Исп1	5								■												
	Исп2	5									■											
11	Исп1	3										■										
	Исп2	3											■									
12	Исп1	12												■	■							
	Исп2	12													■	■						
13	Исп1	9														■	■					
	Исп2	9														■	■					
14	Исп2	12															■	■				
15	Исп2	9																■	■			
16	Исп2	4																	■	■		
17	Исп1	10																		■		
	Исп2	10																		■		

Продолжение таблицы 10

№	Исполнитель	Длительность, дн	Продолжительность работ																	
			Февраль				Март					Апрель				Май				
			1,2	8,2	15,2	22,2	1,3	8,3	15,3	22,3	29,3	5,4	12,4	19,4	26,4	3,5	10,5	17,5	24,5	31,5
18	Исп1	5																		
	Исп2	5																		
19	Исп2	5																		
20	Исп2	10																		
21	Исп2	6																		

Примечание:

 – Исп. 1 (научный руководитель),  – Исп. 2 (инженер)

3.2.4 Бюджет научно-технического исследования (НИР)

В процессе формирования бюджета НИР используется следующая группировка затрат по статьям:

- материальные затраты НИР;
- затраты на основное оборудование;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

3.2.5 Расчет материальных затрат НИР

В таблице 11 представлены материальные затраты данного НИР.

Таблица 11 - Материальные затраты

Наименование	Количество, шт	Цена за ед., руб.	Сумма, руб.
Комплекс канцелярских принадлежностей	4	340	1200
Картридж для лазерного принтера	1	3490	3490
Итого			8290

3.2.6 Расчет амортизации специального оборудования

Для реализации проекта необходимо наличие как минимум одного ноутбука на операционной системе macOS, поскольку разработка мобильного приложения на языке Swift подразумевает работу именно с техникой компании Apple. Всё необходимое ПО имело либо полностью бесплатную лицензию, либо бесплатную версию с уменьшенным функционалом, которого для данного проекта вполне достаточно.

При выполнении научно-исследовательского проекта использовался ПЭВМ – MacBook Air. Срок полезного использования данного ноутбука по паспорту составляет 3 года.

Таблица 12 - Затраты на оборудование

№	Наименование оборудования	Кол-во, шт.	Срок полезного использования, лет	Цены единицы оборудования, тыс. руб.	Общая стоимость оборудования, тыс. руб.
1	ПЭВМ	1	3	65	65
Итого		65 тыс. руб.			

Расчет амортизации проводится следующим образом:

Норма амортизации определяется по следующей формуле:

$$H_A = \frac{1}{n}, \quad (7)$$

где n – срок полезного использования в годах.

Амортизация определяется по следующей формуле:

$$A = \frac{H_A \cdot I}{12} \cdot m, \quad (8)$$

где I – итоговая сумма, тыс. руб.;

m – время использования, мес.

Рассчитаем норму амортизации для ноутбука, с учётом того, что срок полезного использования составляет 3 года:

$$H_A = \frac{1}{n} = \frac{1}{3} = 0,33. \quad (9)$$

Общую сумму амортизационных отчислений находим следующим образом:

$$A = \frac{0,33 \cdot 65000}{12} \cdot 3 = 5362,5 \text{ руб} \quad (10)$$

3.2.7 Основная заработная плата исполнителей темы

Статья включает основную заработную плату работников, непосредственно занятых выполнением НИР, (включая премии и доплаты) и дополнительную заработную плату. Также включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20 –30 % от тарифа или оклада:

$$Z_{\text{зп}} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (11)$$

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

$Z_{\text{доп}}$ – дополнительная заработная плата (12 – 20 % от $Z_{\text{осн}}$).

Основная заработная плата ($Z_{\text{осн}}$) научного руководителя рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_p, \quad (12)$$

где $Z_{\text{осн}}$ –основная заработная плата одного работника;

$Z_{\text{дн}}$ – среднедневная заработная плата работника, руб;

T_p – продолжительность работ, выполняемых научно – техническим работником, раб.дн.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_m \cdot M}{F_d}, \quad (13)$$

где Z_m –месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года;

F_d – действительный годовой фонд рабочего времени научно технического персонала, раб.дн.

В таблице 13 приведен баланс рабочего времени каждого работника НИР.

Таблица 13 - Баланс рабочего времени

Показатели рабочего времени	Руководитель	Бакалавр
Календарное число дней	365	365
Количество нерабочих дней:	-	-
Выходные дни	91	91
Праздничные дни	14	14
Потери рабочего времени:	-	-
Отпуск	28	28
Невыходы по болезни	0	0
Действительный годовой фонд рабочего времени	232	232

Месячный должностной оклад работника:

$$Z_m = (Z_{тс} + (1 + k_{пр} + k_d)) \cdot k_p, \quad (14)$$

где $Z_{тс}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30 % от $Z_{тс}$);

k_d – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5;

k_p – районный коэффициент, для Томска равный 1,3.

В таблице 14 приведен расчет основной заработной платы.

Таблица 14 - Расчет основной заработной платы

Категория	$Z_{тс}$, руб.	k_d	$k_{пр}$	k_p	Z_m , руб	$Z_{дн}$, руб.	T_p , раб. дн.	$Z_{осн}$, руб.
	Руководитель							
ППС3	26300	0,2	0,3	1,3	51285	2 276,87	12,4	28 233,2
	Инженер							
ППС1	17000	0,2	0,3	1,3	33150	1 471,7	94	138 339,8

Итого:	166 573
--------	---------

В таблице 15 представлена общая заработная исполнителей работы.

Таблица 15 - Общая заработная плата

Исполнитель	$Z_{осн}$, руб.	$Z_{доп}$, руб.	$Z_{ит}$, руб.
Руководитель	28 233,2	4 234,98	32 468,18
Инженер	138 339,8	20 750,97	159 090,77
Итого	166 573	24 985,95	190 558,95

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{доп} = k_{доп} \cdot Z_{осн} \quad (15)$$

где $k_{доп}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).

3.2.8 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{внеб} = k_{внеб} \cdot (Z_{осн} + Z_{доп}), \quad (16)$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

На 2023 г. в соответствии с Федеральным законом от 05.08.2000 N 117-ФЗ (ред. от 28.04.2023) установлен размер страховых взносов равный 30,2 %.

Отчисления во внебюджетные фонды представлены в таблице 16.

Таблица 16 - Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.
Руководитель	28 233,2	4 234,98
Инженер	138 339,8	20 750,97
Коэффициент отчислений во внебюджетные фонды	0,32	
Итого:	60 978,864	

3.2.9 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 17.

Таблица 17 - Расчет бюджета затрат НИР

Наименование статьи	Сумма, руб.			Примечание
	Текущий проект	Исп. 2	Исп. 3	
1. Материальные затраты НИР	8290	8290	8290	Таблица 11
2. Затраты на специальное оборудование	65000	113000	105000	Таблица 12
3. Затраты по основной заработной плате исполнителей темы	166 573			Таблица 14
4. Затраты по дополнительной заработной плате исполнителей темы	24 985,95			Таблица 15

5. Отчисления во внебюджетные фонды	60 978,864			Таблица 16
6. Накладные расходы	50 806			
7. Бюджет затрат НИР	376 633,81	424 633,81	416 633,81	Сумма ст. 1-6

Как видно из таблицы, основные затраты НИР приходится на приобретение оборудования.

3.2.10 Накладные расходы

Накладные расходы учитывают затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = (\text{сумма статей } 1 - 5) \cdot k_{\text{нр}}$$

Величину коэффициента накладных расходов можно взять в размере 16%. $Z_{\text{накл}} = (65000 + 166\,573 + 24\,985,95 + 60\,978,86) \cdot 0,16 = 50\,806$ руб.

3.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{\text{pi}}}{\Phi_{\text{max}}}, \quad (12)$$

где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-

исследовательского проекта (в т.ч. аналоги).

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в разгах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в разгах (значение меньше единицы, но больше нуля).

В таблице 18 представлена сравнительная оценка характеристик вариантов исполнения проекта.

Таблица 18 - Сравнительная оценка характеристик вариантов исполнения

Объект исследования Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Скорость загрузки приложения	0,25	5	2	3
2. Удобство в эксплуатации	0,15	5	5	5
3. Помехоустойчивость	0,20	5	4	5
4. Надежность	0,25	2	4	2
5. Материалоемкость	0,15	5	4	5
ИТОГО	1	4,25	3,65	3,7

Сравнивая значения интегральных показателей ресурсоэффективности можно сделать вывод, что реализация приложения в первом исполнении является более эффективным вариантом для проектирования с позиции ресурсосбережения.

Интегральный показатель эффективности вариантов исполнения разработки ($I_{исп.i.}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.1} = \frac{I_{p-исп1}}{I_{финр}}, \quad I_{исп.2} = \frac{I_{p-исп2}}{I_{финр}} \quad \text{и т.д.} \quad (13)$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта (табл. 19) и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта (\mathcal{E}_{cp}):

$$\mathcal{E}_{cp} = \frac{I_{исп.1}}{I_{исп.2}} \quad (14)$$

Таблица 19 - Сравнительная эффективность разработки

№ п/п	Показатели	Текущий проект	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,76	1	0,96
2	Интегральный показатель ресурсоэффективности разработки	4,25	3,65	3,7
3	Интегральный показатель эффективности	5,59	3,65	3,85
4	Сравнительная эффективность вариантов исполнения	1	0,86	0,87

На основании всех расчетов видно, что исполнение в первом варианте наиболее эффективно с точки зрения финансовой и ресурсной эффективности. Поэтому для создания прототипа выбирается именно оно. В ходе выполнения данного раздела ВКР был составлен план работ в графическом отображении (диаграмма Ганта), произведена оценка конкурентоспособности проекта, а также выполнен расчет затрат на выполнение НИР.

ВЫВОДЫ ПО РАЗДЕЛУ

В результате выполнения целей раздела можно сделать следующие выводы:

1. Результатом анализа конкурентных технических решений является выбор одного из вариантов реализации НИР как наиболее подходящего и оптимального по сравнению с другими.

2. В ходе планирования для руководителя и инженера был разработан график реализации этапа работ, который позволяет оценивать и планировать рабочее время исполнителей. Определено следующее: общее количество календарных дней для выполнения работ составляет 123 дня; общее количество дней, в течение которых работал инженер, составляет 94 дня; общее количество дней, в течение которых работал руководитель, составляет 12 дней;

3. Для оценки затрат на реализацию проекта разработан проектный бюджет, который составляет 376 633,81 руб;

4. Результат оценки эффективности ИР показывает следующие выводы:

1) значение интегрального финансового показателя ИР составляет 0,76, что является показателем того, что ИР является финансово выгодной по сравнению с аналогами;

2) значение интегрального показателя ресурсоэффективности ИР составляет 4,25, по сравнению с 3,65 и 3,7;

3) значение интегрального показателя эффективности ИР составляет 5,59, по сравнению с 3,65 и 3,85, и является наиболее высоким, что означает, что техническое решение, рассматриваемое в ИР, является наиболее эффективным вариантом исполнения.

ГЛАВА 4. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

ВВЕДЕНИЕ

Объектом разработки данной ВКР является мобильное приложение на основе OS IOS, обеспечивающее пользователям возможность мониторинга изменения невусов, контроль внешних признаков невусов для самодиагностики. Предоставление пользователю информации о погоде. Возможность пользователю выбрать нужный фототип кожи и получить информацию о необходимых мерах защиты в выбранном городе в зависимости от значения УФ-индекса.

Проект выполняется на персональном компьютере (ПК), поэтому в данном разделе проводится анализ опасных и вредных факторов при работе с ПК, влияния этих факторов на окружающую среду и мероприятий по её защите.

Предметом исследования является рабочая зона разработчика, включая компьютерный стол, ПК, клавиатуру, компьютерную мышь и стул. Работы выполнялись в компьютерном классе 10 корпуса ТПУ.

4.1 Правовые и организационные вопросы обеспечения безопасности

Разработка приложения происходит за компьютерным столом. Рабочее место должно удовлетворять требованиям ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя» [13]:

- Рабочее место для выполнения работ сидя организуют при легкой работе, не требующей свободного передвижения работающего, а также при работе средней тяжести в случаях, обусловленных особенностями технологического процесса. Категории работ - по ГОСТ 12.1.005-88.

- Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и

т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы.

- Рабочее место должно быть организовано в соответствии с требованиями стандартов, технических условий и (или) методических указаний по безопасности труда.

Согласно РД 153-34.0-03.298-2001 «Типовая инструкция по охране труда для пользователей персональными электронно-вычислительными машинами (ПЭВМ) в электроэнергетике», работа оператора ПК относится к категории работ, связанных с опасными и вредными условиями труда. На оператора ПК может воздействовать ряд опасных и вредных производственных факторов, наиболее значимые из которых следующие:

- повышенный уровень напряжения в электрических цепях питания и управления ПК, который может привести к электротравме оператора при отсутствии заземления или зануления оборудования (источник - переменный ток промышленной частоты 50 Гц напряжением 220 В, служащий для питания ПК, а также токи высокой частоты напряжением до 12 000 В систем питания отдельных схем и узлов дисплея);

- повышенный уровень напряженности электрического и магнитного полей в широком диапазоне частот (в том числе от токов промышленной частоты 50 Гц от ПК, вспомогательных приборов, других электроустановок, силовых кабелей, осветительных установок и т.п. - особенно при отсутствии заземления или зануления оборудования).

Требования к нормам труда (продолжительность рабочего дня, перерывы в течение рабочего дня, перерывы на обед) регламентируются ТК РФ «Рабочее время» [14]. Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю. В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Правилами внутреннего трудового распорядка или трудовым договором может быть предусмотрено, что указанный перерыв

может не предоставляться работнику, если установленная для него продолжительность ежедневной работы (смены) не превышает четырех часов. Время предоставления перерыва и его конкретная продолжительность устанавливаются правилами внутреннего трудового распорядка или по соглашению между работником и работодателем.

4.2 Производственная безопасность

При разработке программного обеспечения разработчики подвергаются воздействию различных вредных и опасных факторов, которые представлены в таблице 20. В таблице также представлены соответствующие нормативные документы и этапы работ, во время которых разработчики могут столкнуться с их влиянием.

Таблица 20 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Отклонение показателей микроклимата	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [15]
Недостаточная освещённость рабочей зоны	СП 52.13330.2016 «Естественное и искусственное освещение» [16]
Повышенная световая и цветовая контрастность	СП 52.13330.2016 «Естественное и искусственное освещение» [16]
Повышенный уровень шума на рабочем месте	ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности [17]
Повышенный уровень статического электричества	ГОСТ Р 53734.1-2014 «Электростатические явления» [18]
Опасность поражения электрическим током	ГОСТ Р 58698-2019 «Защита от поражения электрическим током» [19]

По данной таблице можно сделать вывод, что на разработчиков программного обеспечения в ходе их деятельности воздействуют только физические и психологические факторы, а химические и биологические факторы отсутствуют.

4.2.1 Отклонение показателей микроклимата

Отклонение показателей микроклимата на рабочем месте от комфортных непосредственно влияет на здоровье работников. Повышение скорости движения воздуха и понижение температуры может привести к переохлаждению организма путем усиления теплообмена и процесса теплоотдачи при испарении пота. Недостаточная влажность в свою очередь ведет к интенсивному испарению влаги со слизистых оболочек. Это может привести к пересыханию, растрескиванию и затем к заражению болезнетворными бактериями. При разработке программного обеспечения используются персональные компьютеры, которые могут непосредственно влиять на микроклимат путем снижения относительной влажности и повышению температуры в рабочем помещении.

Общие требования к микроклимату производственных помещений регламентируются СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений». Санитарные нормы регулируют оптимальные и допустимые значения показателей в рабочей зоне, соответствующие физиологическим потребностям организма человека, для создания комфортных и безопасных условий труда.

Работа, выполняемая командой разработки программного обеспечения, по энергозатратам относится к категории Ia (производится сидя, сопровождается незначительными физическими усилиями). В таблицах 21 и 22 представлены оптимальные и допустимые значения показателей микроклимата на рабочих местах для данной категории.

Таблица 21 – Оптимальные величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22-24	21-25	60-40	0,1
Тёплый	23-25	22-26	60-40	0,1

Таблица 22 – Допустимые величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, °С		Температура поверхности, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
	диапазон ниже оптимальных величин	диапазон выше оптимальных величин			для диапазона температур воздуха ниже оптимальных величин, не более	для диапазона температур воздуха выше оптимальных величин, не более
Холодный	20,0 – 21,9	24,1 – 25,0	19,0 – 26,0	15 – 75	0,1	0,1
Тёплый	21,0 – 22,9	25,1 – 28,0	20,0 – 29,0	15 – 75	0,1	0,1

В производственных помещениях, где поддерживать допускаемые нормативные величины локального микроклимата не представляется возможным, необходимо проводить мероприятия по защите работников от возможного перегревания и охлаждения. Это достигается разными способами: использование систем местного кондиционирования воздуха; регламентацией периодов работы в неблагоприятном локальном микроклимате и отдыха в помещении с микроклиматом, нормализующим тепловое состояние; уменьшение длительности рабочей смены и др.

4.2.2 Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, приводящим к повышенной утомляемости и снижению работоспособности человека на предприятии. Продолжительная работа в условиях низкой освещенности приводит к ухудшению зрения.

Нормы естественного, искусственного и совместного освещения регламентируются СП 52.13330.2016 «Естественное и искусственное освещение». Разработка программного обеспечения относится к категории работ высокой точности – Б (наименьший или эквивалентный объект различения 0,30 – 0,50 мм), подразряд 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее

70%).

В таблице 23 представлены требования к освещению рабочего помещения для разряда Б1.

Таблица 23 - Требования к освещению рабочего помещения

Искусственное освещение				Естественное освещение	
Освещенность на рабочей поверхности от системы общего освещения, лк	Цилиндрическая освещенность	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, Кп, %, не более	Коэффициент естественной освещенности, %, при	
				верхнем или комбинированном	боковом
300	100	21	15	3	1

Яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение. Для снижения влияния вредного фактора недостаточной освещенности необходимо, чтобы уровень естественного освещения рабочего пространства приблизительно совпадал с яркостью дисплея. Проблему недостаточной освещенности помещения можно решить при помощи установки дополнительных осветительных приборов, расширения световых проемов.

4.2.3 Повышенная световая и цветовая контрастность

Отклонение светового и цветового контраста на рабочем месте приводит к быстрому утомлению и снижению уровня работоспособности человека на предприятии. Продолжительное воздействие этого вредного фактора приводит к возникновению проблем со зрением. Нормы светового и цветового контраста регламентируются СП 52.13330.2016 «Естественное и искусственное освещение». Для работы за компьютером (категория работ Б1) нормы контраста представлены в таблице 24.

Таблица 24 – Требования к освещению рабочего помещения

Характеристика зрительной работы	Контраст объекта с фоном	Характеристика фона
Высокой точности	Малый	Средний
	Средний	Темный

Для изменения светового и цветового контраста необходимо отрегулировать уровень естественной и искусственной освещенности

рабочего помещения или заменить текущее оборудование (мониторы) на более качественные, которые позволят сгладить контраст.

4.2.4 Повышенный уровень шума на рабочем месте

Превышение уровня шума на рабочем месте создает психологический и физический стресс, снижающий производительность, концентрацию, внимание, повышает утомляемость. Повышение уровня шума на рабочем месте возможно из-за фона, создаваемого работой персональных компьютеров, наличия центральной системы вентиляции и кондиционирования воздуха.

Предельно допустимые показатели уровня звука, звукового давления регламентируются СП 51.13330.2011 Защита от шума. Актуализированная редакция СНиП 23-03-2003 [20]. Для команды разработчиков программного обеспечения, эти показатели представлены в таблице 25.

Таблица 25 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ в октавных полосах со среднегеометрическими частотами, Гц						
	31,5	63	125	250	500	1000	2000
Творческая деятельность, руководящая работа с повышенными требованиями, научная деятельность, конструирование и проектирование, программирование, преподавание и обучение, врачебная деятельность. Рабочие места в помещениях дирекции, проектноконструкторских бюро, расчетчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных, приема больных в здравпунктах	86	71	61	54	49	45	42

Для снижения уровня шума в производственном помещении можно использовать защитные звукопоглощающие экраны. Для любого

оборудования необходимо регулярно проводить техническое обслуживание, так как загрязнение может увеличить производимый шум.

4.2.5 Повышенный уровень статического электричества

Статическое электричество является опасным производственным фактором, проявление которого может нанести вред здоровью человека (ожоги) или привести к пожару и другим чрезвычайным ситуациям.

При работе за компьютером статический заряд может накапливаться, если нет хорошего контакта с землей или влажность/ионизация воздуха превышает допустимые нормы. Статический разряд в производственных помещениях рассматриваемого типа при условии соответствии нормам микроклимата и организации работ при воздействии на человека вызывает дискомфорт.

Допустимые показатели уровня статического электричества на производстве регламентируются ГОСТ Р 53734.1-2014 «Электростатические явления». В таблице 26 представлены уровни восприятия электростатического заряда человеком.

Таблица 26 – Уровни восприятия людьми электростатического заряда и ответной реакции при емкости тела в 200 пФ

Энергия разряда, мДж	Реакция	Потенциал тела, В
0,1	Разряд ощутим	1000
0,9	Четко ощутим	3000
6,4	Неприятный шок	8000

Для уменьшения накапливаемого статического заряда при работе за компьютером необходимо организовать антистатические рабочие места, соблюдать установленную норму влажности воздуха и поддерживать чистоту помещения, поскольку пыль обладает свойствами диэлектрика.

4.2.6 Опасность поражения электрическим током

Под электробезопасностью подразумевается система технических и организационных мероприятий, направленных на защиту людей от опасного воздействия электрического тока, статического электричества и электромагнитного поля. Значения вышеперечисленных факторов

регулируются ГОСТ Р 58698-2019.

Таблица 27 – Пороги напряжения прикосновения для реагирования

Характер реагирования	Пороги напряжения, В
Реакция испуга	2 (переменный ток)
	8 (постоянный ток)
Мышечная реакция	20 (переменный ток)
	40 (постоянный ток)

Меры предосторожности для основной защиты от поражения электрическим током:

- использование защитных ограждений или оболочек;
- размещение опасных для жизни и здоровья человека участков электропроводов и приборов вне зоны досягаемости рукой;
- ограничение напряжения или питание должно осуществляться от безопасного источника питания;
- автоматическое отключение питания (защитное устройство, которое будет отключать систему, питающую электрическое оборудование в случае замыкания)

Защита от поражения электрическим током может осуществляться посредством системы безопасного сверхнизкого напряжения (БСНН) и защитного сверхнизкого напряжения (ЗСНН).

4.3 Экологическая безопасность

Программное обеспечение не оказывает влияния на окружающую среду, так как его разработка и использование происходит при помощи персональных компьютеров, однако использование самого компьютера может оказывать влияние на окружающую среду. Так в случаях нагрева материнской платы и корпуса монитора происходит выброс в воздух вредных веществ, а в процессе работы компьютера, воздух вблизи него ионизируется, что приводит к повышенной сухости воздуха.

В производстве компьютеров и их комплектующих используются материалы, которые при неправильной утилизации компьютерной техники могут стать причиной загрязнения литосферы. Утилизировать компьютер необходимо после извлечения его компонент, их сортировки и отправки на

повторное использование, это необходимо делать на специально отведённых полигонах с присутствием квалифицированного персонала. Стоит также учитывать, что в технологических процессах производства компьютеров и их комплектующих образуются производственные сточные воды, которые могут являться фактором загрязнения гидросферы.

Соблюдение всех норм при использовании и утилизации компьютерной техники позволяет уменьшить вредное воздействие на окружающую среду.

Класс опасности и предельно-допустимая концентрация указаны в таблице 28 [21].

Таблица 28 – Предельно-допустимые концентрации (ПДК) вредных веществ в воздухе рабочей зоны

Наименование вещества	Формула	Величина ПДК, мг/м ³	Класс опасности
Ртуть	Hg	0,01/0,05	I
Литий и его неорганические соединения	Li	0,02	I
Свинец и его неорганические соединения	Pb	-/0,05	I
Олово фторид	FSn	1/0,2	II
Железо	Fe	-/10	IV
Полиэтиленхлорид	[C ₂ H ₃ Cl] _x	6	III
Алюминий магнит	AlMg	-/6	IV

Для оказания наименьшего влияния на окружающую среду, необходимо проводить специальную процедуру утилизации ПЭВМ и оргтехники, при которой разбирается, сортируется и более 90% отправится на вторичную переработку и менее 10% будут отправлены на свалки. При этом она должна соответствовать процедуре утилизации ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла ОТХОДОВ».

4.4 Безопасность в чрезвычайных ситуациях

4.4.1 Пожар

Причинами возникновения пожара при работе с ЭВМ может служить короткое замыкание проводки, в том числе вследствие неисправности прибора, сильный перегрев ЭВМ в результате его использования в режиме повышенной нагрузки.

Для предотвращения возникновения пожара, необходимо проводить периодическую своевременную диагностику оборудования и электрической проводки, соблюдать нормы при работе с ЭВМ.

На основании Федерального закона от 22.07.2008 N 123-ФЗ (ред. От 30.04.2021) "Технический регламент о требованиях пожарной безопасности" помещения учебной аудитории оборудованы следующими средствами пожаротушения: огнетушитель ручной углекислотный ОУ-5, пожарный кран с рукавом, также аудитория оборудована системой противопожарной сигнализации [21]. Помещение аудитории категории помещения группы – В4, возможный класс пожара – Е [21].

При появлении возгорания необходимо сообщить в службу пожарной охраны адрес и место возникновения пожара.

Вывод по разделу

В данном разделе были рассмотрены правовые и организационные вопросы обеспечения безопасности, аспекты производственной и экологической безопасности, а также рассмотрен вопрос безопасности рабочей группы при чрезвычайном происшествии. В результате можно сделать вывод о работнике:

- Имеет нормальную продолжительность рабочего времени;
- Обеспечен нормальными условиями труда для выполнения норм выработки;
- Имеет, предусмотренные ТК РФ, выходные и праздничные дни;
- Имеет категорию тяжести труда - Ia (производится сидя, сопровождается незначительными физическими усилиями), согласно [3];
- Относится к первой группе по электробезопасности;

А также о рабочей зоне:

- Рабочая зона имеет оптимальные значения показаний микроклимата согласно таблице 21;
- Фактические показатели освещения совпадают с оптимальными в таблице 23;
- Мониторы ЭВМ имеют оптимальную световую и цветовую контрастность;
- Соблюдаются меры предосторожности для основной защиты от поражения электрическим током;
- Вышедшие из строя ПЭВМ утилизируются согласно ГОСТ Р 53692-2009[23];
- Помещения учебной аудитории оборудованы положенными средствами пожаротушения;
- Рабочая зона имеет категорию помещения по электробезопасности согласно ПУЭ – без повышенной опасности [24];
- Помещение аудитории категории помещения группы – В4 (пожароопасное);

- Помещение аудитории находится в объекте, относящийся к IV категории по оказанию негативного воздействия на окружающую среду [25].

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы:

- была описана и проанализирована предметная область;
- определены основные действия пользователей мобильного приложения.

На этапе проектирования и разработки было:

- выбраны используемые паттерны;
- создан функционал для хранения данных пользователя;
- разработан функционал для работы с сетью;
- разработано мобильное приложение для мониторинга внешних признаков невусов и их самодиагностики, также приложение предоставляет возможность пользователю: посмотреть погоду, узнать свой фототип, получить рекомендацию по защите кожи основанную по данным УФ-индекса и фототипа пользователя;
- проведено тестирование.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Gartner Inc. [Электронный ресурс]. – URL: <http://www.gartner.com/newsroom/id/3215217/> (дата обращения: 26.05.23);
2. Использование мобильных телефонов в разных странах [Электронный ресурс]. – URL: <https://www.cossa.ru/152/37433/> (дата обращения: :26.05.23);
3. Рак кожи: профилактика, диагностика, лечение [Электронный ресурс]. – URL: <https://www.niioncologii.ru/highlights/index?id=9643> (дата обращения: :26.05.23);
4. Директор НМИЦ онкологии им. Н.Н. Петрова Минздрав России Алексей Беляев о лечении онкологических заболеваний в России [Электронный ресурс]. – URL: <https://minzdrav.gov.ru/news/2022/07/26/19074-direktor-nmits-onkologii-im-n-n-petrova-minzdrav-rossii-aleksey-belyaev-o-lechenii-onkologicheskikh-zabolevaniy-v-rossii/> (дата обращения: :26.05.23);
5. Скрытая угроза: зачем и как правильно наблюдать за родинками [Электронный ресурс]. – URL: <https://www.gmsclinic.ru/blog/art-skrytaya-ugroza-zachem-i-kak-pravilno-nablyudat-za-rodinkami> (дата обращения: :26.05.23);
6. ЗАЩИТА ОТ СОЛНЦА И ФОТОТИПЫ КОЖИ [Электронный ресурс]. – URL: <http://eraesthetic.lv/ru/blog/adas-fototipi> (дата обращения: :26.05.23);
7. Данные о погоде в быстром и удобном виде [Электронный ресурс]. – URL: <https://openweathermap.org/guide> (дата обращения: :26.05.23);
8. В чем отличия между MVC и MVP шаблонами? [Электронный ресурс]. – URL: <https://for-each.dev/lessons/b/-mvc-vs-mvp-pattern> (дата обращения: :26.05.23);
9. Паттерны для новичков: MVC vs MVP vs MVVM [Электронный ресурс]. – URL: <https://habr.com/ru/articles/215605/> (дата обращения: :26.05.23);
10. Архитектурные паттерны в iOS: привет от дядюшки Боба, или Clean Architecture [Электронный ресурс]. – URL: https://pcnews.ru/blogs/arhitekturnye_patterny_v_iOS_privet_ot_daduski_boba_ili

_clean_architecture-1087296.html#gsc.tab=0 (дата обращения: :26.05.23);

11. Современное программирование [Электронный ресурс]. – URL: https://konference.nvsu.ru/konffiles/357/Sovremennoe_programmirovanie_2020_itog.pdf (дата обращения: :26.05.23);

12. Построение приложений шаг за шагом, часть вторая [Электронный ресурс]. – URL: https://pcnews.ru/blogs/postroenie_android_prilozenij_sag_za_sagom_cast_vtoraa-681835.html#gsc.tab=0 (дата обращения: :26.05.23);

13. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200003913> (дата обращения: 20.05.2023);

14. Трудовой кодекс (ТК РФ) «Рабочее время»// Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <http://base.garant.ru/12125268/> (дата обращения: 23.05.2023);

15. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. 2021. – Режим доступа: <https://docs.cntd.ru/document/573500115> (дата обращения: 20.05.2023);

16. 52.13330.2016 «Естественное и искусственное освещение» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/456054197> (дата обращения: 21.05.2023);

17. ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа:

<https://docs.cntd.ru/document/901703278> (дата обращения: 22.05.2023);

18. ГОСТ Р 53734.1-2014 «Электростатические явления» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200111323> (дата обращения: 22.05.2023);

19. ГОСТ Р 58698-2019 «Защита от поражения электрическим током» // Электронный фонд правовой и нормативнотехнической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200170001> (дата обращения: 22.05.2023);

20. СП 51.13330.2011 «Защита от шума. Актуализированная редакция СНиП 23-03-2003» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200084097> (дата обращения: 25.05.2023);

21. Федеральный закон от 22 июля 2008 г. №123-ФЗ «Технический регламент о требованиях пожарной безопасности» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/902111644> (дата обращения: 23.05.2023);

22. ГН 2.2.5.3532-18 «Предельно допустимые концентрации (ПДК) вредных веществ в воздухе рабочей зоны» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/557235236> (дата обращения: 23.05.2023);

23. ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200081740> (дата обращения: 25.05.2023);

24. «Правила устройства электроустановок» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200030216> (дата обращения: 25.05.2023);

25. Постановление «Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III и IV категорий» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/573292854> (дата обращения: 25.05.2023).

Приложение 1. Внедрение зависимостей

```
import Foundation
import UIKit

class MainWeatherComponent {

    private let citySelectionComponent = CitySelectionComponent()
    private let birthmarkComponent = BirthMarkComponent()
    private let phototypeScreenComponent = PhototypeScreenComponent()
    private let sunProtectionComponent = SunProtectionComponent()
    private unowned let dependencies: RootComponent
    init(parent: RootComponent) {
        dependencies = parent
    }
    func makeMainWeatherScreenViewController() -> UIViewController {
        let precenter = makeMainWeatherScreenPrecenter()
        let citySelectionDialogFabric = makeCitySelectionDialogFabric()
        let makeBirthmarkScreenFabric = makeBirthmarkScreenFabric()
        let phototypeScreenFabric = makePhototypeScreenFabric()
        let sunProtectionFabric = makeSunProtectionFabric()
        let viewController = MainWeatherScreenViewController(
            presenter: precenter,
            citySelectionDialogFabric: citySelectionDialogFabric,
            birthmarkScreenFabric: makeBirthmarkScreenFabric,
            phototypeScreenFabric: phototypeScreenFabric,
            sunProtectionScreenFabric: sunProtectionFabric
        )
        precenter.controller = viewController
        return viewController
    }

    private func makeMainWeatherScreenPrecenter() ->
MainWeatherScreenPrecenter {
        return MainWeatherScreenPrecenterImp(mainWeatherUseCases:
makeUseCases())
    }

    private func makeUseCases() -> WeatherServerQueryUseCase {
        return WeatherServerQueryUseCaseImp(
            locationService: makeLocationService(),
            urlRequestOneDay: makeURLRequestOneDay(),
            urlRequestFiveDays: makeURLRequestFiveDays()
        )
    }
}
```

```

private func makeLocationService() -> LocationService {
    return LocationServiceImp()
}

private func makeURLRequestOneDay() -> URLRequestOneDay {
    return URLRequestOneDayImp(facadeURL:
dependencies.makeURLSessionFacade())
}

private func makeURLRequestFiveDays() -> URLRequestFiveDays {
    return URLRequestFiveDaysImp(facadeURL:
dependencies.makeURLSessionFacade())
}

private func makeCitySelectionDialogFabric() ->
CitySelectionDialogFabric {
    return citySelectionComponent
}

private func makeBirthmarkScreenFabric() -> BirthmarkScreenFabric {
    return birthmarkComponent
}

private func makePhototypeScreenFabric() -> PhototypeScreenFabric {
    return phototypeScreenComponent
}

private func makeSunProtectionFabric() -> SunProtectionScreenFabric {
    return sunProtectionComponent
}

```

Приложение 2. Паттерн проектирования «Фасад»

```
import Foundation

class URLSessionFacadeImp: URLSessionFacade {
    func get<T: Codable>(
        url: String,
        params: [String: String],
        expecting: T.Type
    ) async throws -> T {
        var urlComps = URLComponents(string: url)!

        let queryItems = params.map { key,value in URLQueryItem(name: key,
value: value) }

        if urlComps.queryItems != nil {
            urlComps.queryItems!.append(contentsOf: queryItems)
        }

        let url = urlComps.url!

        let (data, _) = try await URLSession.shared.data(from: url)
        let response = try JSONDecoder().decode(expecting, from: data)
        return response
    }
}
```

Приложение 3. Модель данных на текущий день и на следующие пять дней

```
import Foundation
struct WeatherNowResponse: Codable {
    var id: Int
    var main: String
    var description: String
    var icon: String
}
struct MainNowResponse: Codable {
    var temp: Double = 0.0
    var feels_like: Double = 0.0
}
struct WeatherDataNowResponse: Codable {
    var weather: [WeatherNowResponse] = []
    var main: MainNowResponse = MainNowResponse()
    var name: String = ""
    var dt: Date
}

import Foundation
struct WeatherDataFiveDaysResponse: Codable {
    var list: [ListResponse] = []
    var city: CityResponse = CityResponse()
}
struct ListResponse: Codable {
    var main: MainClassResponse = MainClassResponse()
    var weather: [WeatherResponse] = []
    var dt_txt: String
    var dt: Date
}
struct MainClassResponse: Codable {
    var temp: Double = 0.0
    var feels_like = 0.0
    var temp_min: Double = 0.0
    var temp_max: Double = 0.0
}
struct WeatherResponse: Codable {
    var id: Int = 0
    var main: String = ""
    var icon: String = ""
}
struct CityResponse: Codable {
    var name: String = ""}
```

Приложение 4. Получение данных о погоде на текущий день в формате JSON, конвертация полученных данных.

```
import Foundation
class URLRequestOneDayImp: URLRequestOneDay {

    private let facadeURL: URLSessionFacade

    init(facadeURL: URLSessionFacade) {
        self.facadeURL = facadeURL
    }

    private let latitude = "lat"
    private let longitude = "lon"
    private let url =
"https://api.openweathermap.org/data/2.5/weather?units=metric&lang=ru&appid=
0cb5cfb3c159fb38bbb437174bf5789b"
    func getInfoURLOneDay (
        coordinates: Coordinates) async throws -> WeatherDataNow {
        let params = [
            self.latitude: String(coordinates.latitude),
            self.longitude: String(coordinates.longitude)
        ]

        let weatherDataNowResponse = try await facadeURL.get(url: url, params:
params, expecting: WeatherDataNowResponse.self)
        let weatherDataNowConvert = try
convertResponse(weatherDataNowResponse: weatherDataNowResponse)
        return weatherDataNowConvert
    }

    func convertResponse(weatherDataNowResponse: WeatherDataNowResponse)
throws -> WeatherDataNow {
        let weatherImage = try convertWeatherImage(weatherDataNowResponse:
weatherDataNowResponse)

        let weatherDataNowConvert = WeatherDataNow(
            cityName: weatherDataNowResponse.name,
            date: weatherDataNowResponse.dt,
            weatherImage: weatherImage,
            temp: weatherDataNowResponse.main.temp,
            weatherAnnotation: try weatherAnnotationIntToEnum(id:
weatherDataNowResponse.weather[0].id).enumToString(),
            weatherFeelsLike: weatherDataNowResponse.main.feels_like
        )
    }
}
```



```

    return weatherDataNowConvert
}

func convertWeatherImage(weatherDataNowResponse:
WeatherDataNowResponse) throws -> WeatherImage {
    switch (try weatherImageStringToEnum(id:
weatherDataNowResponse.weather[0].icon)) {
        case .sun: return .sun
        case .rain: return .rain
        case .thunder: return .thunder
    }
}
}
import Foundation
class URLRequestFiveDaysImp: URLRequestFiveDays {

    private let facadeURL: URLSessionFacade

    init(facadeURL: URLSessionFacade) {
        self.facadeURL = facadeURL
    }

    private let latitude = "lat"
    private let longitude = "lon"
    private let url =
"https://api.openweathermap.org/data/2.5/forecast?units=metric&lang=ru&appid=
0cb5cfb3c159fb38bbb437174bf5789b"

    func getInfoURLFiveDays(
        coordinates: Coordinates
    ) async throws -> [WeatherDataForFiveDays] {
        let params = [
            self.latitude: String(coordinates.latitude),
            self.longitude: String(coordinates.longitude)
        ]

        let weatherDataFiveDaysResponse = try await facadeURL.get(url: url,
params: params, expecting: WeatherDataFiveDaysResponse.self)
        let weatherDataFiveDaysConvert = try
convertResponse(weatherDataFiveDaysResponse:
weatherDataFiveDaysResponse)
        return weatherDataFiveDaysConvert
    }

    private func convertResponse(
        weatherDataFiveDaysResponse: WeatherDataFiveDaysResponse

```

```

) throws -> [WeatherDataForFiveDays] {
  var weatherList = [WeatherDataForFiveDays]()
  var collectionWeather = [HourlyWeatherData]()

  var maxTemp: [Double] = []
  var minTemp: [Double] = []

  try weatherDataFiveDaysResponse.list.enumerated().map { (index, element)
in
  if index % 8 == 0 && index != 0 {
    weatherList.append(contentsOf: fillWeatherDataForFiveDays(
      index: index,
      element: element,
      collectionWeather: collectionWeather,
      minTemp: maxTemp,
      maxTemp: maxTemp
    ))
    maxTemp = []
    minTemp = []
    collectionWeather = []
  }

  maxTemp.append(element.main.temp_max)
  minTemp.append(element.main.temp_min)

  let weatherImage = try convertWeatherImage(element: element, i: index)

  collectionWeather = fillCollectionOfHourlyWeather(
    collectionWeather: collectionWeather,
    element: element,
    i: index,
    weatherImage: weatherImage, dateString: element.dt
  )
}
return weatherList
}

func convertWeatherImage(element: ListResponse, i: Int) throws ->
WeatherImage {
  switch (try weatherImageStringToEnum(id: element.weather[0].icon)) {
  case .sun: return .sun
  case .rain: return .rain
  case .thunder: return .thunder
  }
}
}

```

```

func fillWeatherDataForFiveDays(
    index: Int,
    element: ListResponse,
    collectionWeather: [HourlyWeatherData],
    minTemp: [Double],
    maxTemp: [Double]
) -> [WeatherDataForFiveDays] {
    var weatherList = [WeatherDataForFiveDays]()
    weatherList.append(WeatherDataForFiveDays(
        date: element.dt,
        minTemp: String(minTemp.min()!),
        maxTemp: String(maxTemp.max()!),
        weatherImage: WeatherImage.sun,
        hourlyWeatherData: collectionWeather
    )
    )
    return weatherList
}

func fillCollectionOfHourlyWeather(
    collectionWeather: [HourlyWeatherData],
    element: ListResponse,
    i: Int,
    weatherImage: WeatherImage,
    dateString: Date
) -> [HourlyWeatherData] {
    var collectionWeather = collectionWeather
    collectionWeather.append(HourlyWeatherData(
        temp: String(Int(element.main.temp)) + "°",
        weatherImage: weatherImage,
        time: dateString
    )
    )
    return collectionWeather
}
}

```

Приложение 5. CoreData

```
import UIKit
import CoreData
public final class CoreDataManager: NSObject {
    public static let shared = CoreDataManager()
    private override init() {}
    private var appDelegate: AppDelegate {
        UIApplication.shared.delegate as! AppDelegate
    }

    private var context: NSManagedObjectContext {
        appDelegate.persistentContainer.viewContext
    }

    public func createBirthMark(location: String, size: String, color: String,
    contour: String, image: Data, id: Int16) {
        guard let userEntityDescription =
    NSEntityDescription.entity(forEntityName: "UserCoreData", in: context) else {
            return
        }
        let birthmark = UserCoreData(entity: userEntityDescription, insertInto:
    context)
        birthmark.color = color
        birthmark.contour = contour
        birthmark.location = location
        birthmark.size = size
        birthmark.image = image
        birthmark.id = id

        appDelegate.saveContext()
    }

    public func fethPhotoBirthMarks() -> [UserCoreData] {
        let fetchRequest =
    NSFetchRequest<NSFetchRequestResult>(entityName: "UserCoreData")
        do {
            return (try? context.fetch(fetchRequest) as? [UserCoreData]) ?? []
        }
    }
}
```

Приложение 6. Предсказательная модель для определения злокачественных образований

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten
from tensorflow.keras.metrics import BinaryAccuracy, Precision, Recall
import warnings
warnings.filterwarnings("ignore")
tf.keras.backend.clear_session()

test_data =
keras.utils.image_dataset_from_directory('/content/drive/MyDrive/archive/test')
train_data =
keras.utils.image_dataset_from_directory('/content/drive/MyDrive/archive/train')

batch = train_data.as_numpy_iterator().next()

fig, ax = plt.subplots(3, 5, figsize=(15,10))
ax = ax.flatten()
for idx, img in enumerate(batch[0][:15]):
    ax[idx].imshow(img.astype(int))
    ax[idx].title.set_text(batch[1][idx])

train_data = train_data.map(lambda x,y: (x/255, y))
test_data = test_data.map(lambda x,y: (x/255, y))
batch = train_data.as_numpy_iterator().next()
print("Minimum value of the scaled data:", batch[0].min())
print("Maximum value of the scaled data:", batch[0].max())
batch = train_data.as_numpy_iterator().next()
data_augmentation = Sequential([
    layers.RandomFlip("horizontal_and_vertical", input_shape=(256,256,3)),
    layers.RandomZoom(0.1),
    layers.RandomContrast(0.1),
    layers.RandomRotation(0.2)
])

image = batch[0]
```

```

plt.figure(figsize=(10, 10))
for i in range(9):
    augmented_image = data_augmentation(image)
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(augmented_image[0])
    plt.axis("off")
model = Sequential([
    data_augmentation,
    Conv2D(16, (3,3), 1, activation="relu", padding="same"),
    Conv2D(16, (3,3), 1, activation="relu", padding="same"),
    MaxPooling2D(),
    Conv2D(32, (5,5), 1, activation="relu", padding="same"),
    Conv2D(32, (5,5), 1, activation="relu", padding="same"),
    MaxPooling2D(),
    Conv2D(16, (3,3), 1, activation="relu", padding="same"),
    Conv2D(16, (3,3), 1, activation="relu", padding="same"),
    MaxPooling2D(),

    Flatten(),
    Dense(128, activation="relu"),
    Dense(1, activation="sigmoid")
])
model.compile(loss='binary_crossentropy', # binary kullanilir 2 sinifta
              optimizer='adam',
              metrics=[tf.keras.metrics.Precision(), tf.keras.metrics.Recall(), "acc"])
history = model.fit(train_data, epochs=10, validation_data=test_data)
bin_acc = BinaryAccuracy()
recall = Recall()
precision = Precision()

for batch in test_data.as_numpy_iterator():
    X, y = batch
    yhat = model.predict(X)
    bin_acc.update_state(y, yhat)
    recall.update_state(y, yhat)
    precision.update_state(y, yhat)

print("Accuracy:", bin_acc.result().numpy(), "\nRecall:", recall.result().numpy(),
      "\nPrecision:", precision.result().numpy())

batch = test_data.as_numpy_iterator().next()

img, label = batch[0][15], batch[1][15]
plt.imshow(img)
if label==1:
    plt.title("Malignant")

```

```
else:  
    plt.title("Benign")  
plt.show()  
y_hat = model.predict(np.expand_dims(img, 0))
```

```
if y_hat < 0.5:  
    print("Benign")  
else:  
    print("Malignant")
```

Приложение 7. Верстка

```
import UIKit
import SnapKit
fileprivate enum SectionTableWeather {
    case main
}
class MainWeatherScreenView: UIView {
    var phototypeTapped: (() -> Void)?
    var goToSearchCity: (() -> Void)?
    var goToBirthmark: (() -> Void)?
    var goToSunInfo: (() -> Void)?

    private let mainWeatherView = UIView()
    private let findPhototypeView = UIButton()
    private let labelFindPhototype = UILabel()
    private let viewBlue = UIView()
    private let labelToDayWeather = UILabel()
    private let imageWeather = UIImageView()
    private let labelTempWheather = UILabel()
    private let labelTextWheather = UILabel()
    private let tableView = UITableView()

    private lazy var dataSource:
UITableViewDiffableDataSource<SectionTableWeather, WeatherDataForFiveDay
s> = createDataSource()

    private func createDataSource() ->
UITableViewDiffableDataSource<SectionTableWeather,
WeatherDataForFiveDays> {

UITableViewDiffableDataSource<SectionTableWeather, WeatherDataForFiveDay
s> (tableView: tableView) {
        (tableView, indexPath, weatherList) -> UITableViewCell? in
            let cell = tableView.dequeueReusableCell(withIdentifier:
CollectionTableViewCell.identifier,
                for: indexPath) as?
CollectionTableViewCell
            cell?.setupCell(weatherDataCell: weatherList)

        return cell
    }
}

override init(frame: CGRect) {
```



```

super.init(frame: frame)
self.backgroundColor = .white

setupTableView()
setupFindPhototypeView()
setupMainWeatherView()
setupViewBlue()
setupLabelToDayWeather()
setupImageWeather()
setupLabelTempWheather()
setupLabelTextWheather()
}

private func setupTableView() {
    self.addSubview(tableView)

    self.tableView.register(
        CollectionTableViewCell.self,
        forCellReuseIdentifier: CollectionTableViewCell.identifier
    )

    tableView.isHidden = true
    tableView.allowsSelection = false
    tableView.showsVerticalScrollIndicator = false
    tableView.separatorStyle = .none
    tableView.snp.makeConstraints { make in
        make.left.right.equalToSuperview().inset(0)
        make.bottom.equalTo(self.snp_bottom).inset(0)
    }

    tableView.delegate = self
}

private func setupFindPhototypeView() {
    self.addSubview(findPhototypeView)
    findPhototypeView.layer.cornerRadius = 16
    findPhototypeView.layer.maskedCorners = [.layerMaxXMinYCorner,
.layerMinXMinYCorner, .layerMinXMaxYCorner, .layerMaxXMaxYCorner]
    findPhototypeView.setTitleColor(R.color.white(), for: .normal)
    findPhototypeView.titleLabel?.font = R.font.manropeBold(size: 16)
    findPhototypeView.backgroundColor = R.color.dodger_Blue()
    findPhototypeView.titleLabel?.numberOfLines = 0

    findPhototypeView.snp.makeConstraints { make in
        make.height.equalToSuperview().multipliedBy(0.08)
        make.top.equalTo(self.safeAreaLayoutGuide).inset(16)

```

```

        make.left.right.equalToSuperview().inset(16)
    }

    findPhototypeView.addTarget(
        self,
        action: #selector(self.phototypeTappedView),
        for: .touchUpInside
    )
}

func myPhototypeView(text: String) {
    findPhototypeView.setTitle(text, for: .normal)
}

@objc func phototypeTappedView() {
    phototypeTapped?()
}

private func setupMainWeatherView() {
    self.addSubview(mainWeatherView)

    mainWeatherView.isHidden = true
    mainWeatherView.layer.cornerRadius = 24
    mainWeatherView.backgroundColor = R.color.malibu()
    mainWeatherView.clipsToBounds = true
    mainWeatherView.snp.makeConstraints { make in
        make.left.right.equalToSuperview().inset(16)
        make.top.equalTo(findPhototypeView.snp_bottom).inset(-16)
        make.bottom.equalTo(tableView.snp_top).inset(-16)
    }
}

private func setupViewBlue() {
    mainWeatherView.addSubview(viewBlue)
    viewBlue.layer.cornerRadius = 100
    viewBlue.layer.maskedCorners = [.layerMaxXMinYCorner,
    .layerMinXMinYCorner]
    viewBlue.backgroundColor = R.color.brilliant_Azure()

    viewBlue.snp.makeConstraints { make in
        make.height.equalToSuperview().multipliedBy(0.65)
        make.left.right.bottom.equalToSuperview().inset(0)
    }
}

private func setupLabelToDayWeather() {

```

```

mainWeatherView.addSubview(labelToDayWeather)

labelToDayWeather.textColor = .white
labelToDayWeather.font = R.font.manropeMedium(size: 14)
labelToDayWeather.snp.makeConstraints { make in
    make.top.equalToSuperview().inset(16)
    make.centerX.equalToSuperview()
}
}

private func setupImageWeather() {
    mainWeatherView.addSubview(imageWeather)

    imageWeather.snp.makeConstraints { make in
        make.height.equalToSuperview().multipliedBy(0.45)
        make.top.equalTo(labelToDayWeather).inset(28)
        make.centerX.equalToSuperview()
    }
}

private func setupLabelTempWheather() {
    mainWeatherView.addSubview(labelTempWheather)

    labelTempWheather.textColor = .white
    labelTempWheather.font = R.font.manropeBold(size: 48)
    labelTempWheather.snp.makeConstraints { make in
        make.top.equalTo(imageWeather.snp_bottom).inset(0)
        make.centerX.equalToSuperview()
    }
}

private func setupLabelTextWheather() {
    mainWeatherView.addSubview(labelTextWheather)

    labelTextWheather.textColor = .white
    labelTextWheather.font = R.font.manropeMedium(size: 14)
    labelTextWheather.snp.makeConstraints { make in
        make.bottom.equalToSuperview().inset(16)
        make.centerX.equalToSuperview()
    }
}

required init?(coder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
}

```

```

func setupNavigationBar(navigationItem: UINavigationController) {
    navigationItem.title = R.string.localizable.chooseCity()

    navigationItem.leftBarButtonItem = UIBarButtonItem(
        title: R.string.localizable.moles(),
        style: .plain,
        target: self,
        action: #selector(self.goToBirthmarkView)
    )
    navigationItem.leftBarButtonItem?.tintColor = .black

    let sunHeader = UIBarButtonItem (
        image: R.image.sunHeader()!.withRenderingMode(.alwaysOriginal),
        style: .plain,
        target: self,
        action: #selector(self.goToSunInfoView)
    )

    sunHeader.customView?.heightAnchor.constraint(equalToConstant:
4).isActive = true
    sunHeader.customView?.widthAnchor.constraint(equalToConstant:
4).isActive = true

    let searchHeader = UIBarButtonItem(
        image:
R.image.searchHeader()!.withRenderingMode(.alwaysOriginal),
        style: .plain,
        target: self,
        action: #selector(self.goToSearchCityView)
    )
    navigationItem.rightBarButtonItemItems = [searchHeader, sunHeader]
}

@objc func goToSunInfoView() {
    goToSunInfo?()
}

@objc func goToBirthmarkView() {
    goToBirthmark?()
}

@objc func goToSearchCityView() {
    goToSearchCity?()
}

```

```

func          updateViewNow(weatherDataNow:          WeatherDataNow,

```

```

navigationItem: UINavigationController) {
    //Выводим актуальную погоду
    navigationItem.title = weatherDataNow.cityName

    let dateFormatter = DateFormatter()
    dateFormatter.dateFormat = "d MMMM"
    let dateString = dateFormatter.string(from: weatherDataNow.date)
    updateView(dateString: dateString, weatherDataNow:
weatherDataNow)
}

private func updateView(dateString: String, weatherDataNow:
WeatherDataNow) {

    labelToDayWeather.text = R.string.localizable.today() + " " + dateString

    let weatherImage = weatherDataNow.weatherImage.enumToImage()
    setupImage(image: weatherImage)

    labelTempWheather.text = String(Int(weatherDataNow.temp)) + "°"

    labelTextWheather.text = weatherDataNow.weatherAnnotation +
    ", " + R.string.localizable.feelsLike() + " " +
    String(Int(weatherDataNow.weatherFeelsLike)) + "°"

}

func setupImage(image: UIImage) {
    imageWeather.image = image
}

func showTable() {
    mainWeatherView.isHidden = false
    tableView.isHidden = false
}

func reloadData(from weatherList: [WeatherDataForFiveDays]) {
    var snapshot = NSDiffableDataSourceSnapshot<SectionTableWeather,
WeatherDataForFiveDays>()
    snapshot.appendSections([.main])
    snapshot.appendItems(weatherList)

    dataSource.apply(snapshot, animatingDifferences: true)
}
}
extension MainWeatherScreenView: UITableViewDelegate {

```

```
func tableView(_ tableView: UITableView, heightForRowAt indexPath:
IndexPath) -> CGFloat {
    return 230.0
}
}
```

Приложение 8. Тестирование

```
import XCTest
@testable import Wheather_App
enum ErrorTest: Error {
    case errorTest(String)
}
class TestMainWeatherScreenPrecenter: XCTestCase {

    let component = MainWeatherComponent()

    override func setUp() {
        super.setUp()
        component.precenter.controller = component.controller
    }

    override func tearDown() {
        super.tearDown()
    }

    func
testLocationServiceRunningSuccessfullyAndServerSuccessfullyResponded() {
        component.locationService.responseLocationService           =
responseLocationService
        component.urlSessionFacadeOneDayMock.response = responseOneDay
        component.urlSessionFacadeFiveDayMock.response           =
responseFiveDay

        component.precenter.confirmButtonTapped(city: "Москва")

        wait(for: [component.controller.expectationReloadData,
                    component.controller.expectationUpdateViewNow,
                    component.controller.expectationShowTable,
                    component.controller.expectationNavSetup], timeout: 5.0)

        XCTAssertEqual(component.controller.showTableBool, true)
        XCTAssertEqual(component.controller.reloadDataBool, true)
        XCTAssertEqual(component.controller.navSetupBool, true)
        XCTAssertEqual(component.controller.updateViewNowBool, true)

        XCTAssertEqual(component.controller.showToastBool, false)

        XCTAssertEqual(component.urlSessionFacadeOneDayMock.url,
expectedURLOneDay)
        XCTAssertEqual(component.urlSessionFacadeOneDayMock.params,
```

expectedParameters)

```
    XCTAssertEqual(component.urlSessionFacadeFiveDayMock.url,
expectedURLFiveDays)
    XCTAssertEqual(component.urlSessionFacadeFiveDayMock.params,
expectedParameters)
```

```
    XCTAssertEqual(component.locationService.city, "Москва")
}
```

```
func testLocationServiceGaveError() {
    component.locationService.responseLocationError =
responseLocationServiceError
```

```
    component.precenter.confirmButtonTapped(city: "Несуществующий
город")
```

```
    wait(for: [component.controller.expectationShowToast], timeout: 5.0)
```

```
    XCTAssertEqual(component.controller.updateViewNowBool, false)
    XCTAssertEqual(component.controller.showTableBool, false)
    XCTAssertEqual(component.controller.reloadDataBool, false)
    XCTAssertEqual(component.controller.navSetupBool, false)
```

```
    XCTAssertEqual(component.controller.showToastBool, true)
}
```

```
func
testLocationServiceRunningSuccessfullyAndServerOneDayReturnedError() {
    component.locationService.responseLocationService =
responseLocationService
    component.urlSessionFacadeOneDayMock.responseError =
responseError
```

```
    component.precenter.confirmButtonTapped(city: "Москва")
```

```
    wait(for: [component.controller.expectationShowToast], timeout: 5.0)
```

```
    XCTAssertEqual(component.controller.updateViewNowBool, false)
    XCTAssertEqual(component.controller.showTableBool, false)
    XCTAssertEqual(component.controller.reloadDataBool, false)
    XCTAssertEqual(component.controller.navSetupBool, false)
```

```
    XCTAssertEqual(component.controller.showToastBool, true)
}
```



```

func
testLocationServiceRunningSuccessfullyAndServerFiveDayReturnedError() {
    component.locationService.responseLocationService =
responseLocationService
    component.urlSessionFacadeOneDayMock.response = responseOneDay
    component.urlSessionFacadeFiveDayMock.responseError =
responseError

    component.precenter.confirmButtonTapped(city: "MockBa")

    wait(for: [component.controller.expectationShowToast], timeout: 5.0)

    XCTAssertEqual(component.controller.updateViewNowBool, false)
    XCTAssertEqual(component.controller.showTableBool, false)
    XCTAssertEqual(component.controller.reloadDataBool, false)
    XCTAssertEqual(component.controller.navSetupBool, false)

    XCTAssertEqual(component.controller.showToastBool, true)
}
}

```