**Министерство сельского хозяйства Российской Федерации** Федеральное государственное бюджетное образовательное учреждение высшего образования

**КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ имени И.Т. ТРУБИЛИНА**

**Факультет прикладной информатики Кафедра информационных систем**

КУРСОВАЯ РАБОТА

По дисциплине: Разработка приложений под мобильные устройства

На тему: «Разработка мобильного приложения “Цветы и уход за ними”»

Выполнила: Хоружая Алина Максимовна

Группа: ПИ1901

Руководитель: Л. Е. Попок

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ ИМЕНИ И.Т.ТРУБИЛИНА»

**Факультет прикладной информатики**

**КУРСОВАЯ РАБОТА**

по дисциплине: Разработка приложений под мобильные устройства на тему: «Разработка мобильного приложения “Цветы и уход за ними”»

выполнил студент группы ПИ1901 Хоружая Алина Максимовна

Руководитель работы Попок Л. Е.

(подпись, расшифровка подписи)

Защищен

(дата)

Оценка

Члены комиссии

(подпись, дата, расшифровка подписи)

Краснодар 2022

**Министерство сельского хозяйства Российской Федерации** Федеральное государственное образовательное учреждение высшего образования

**КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ**

**Имени И.Т. Трубилина**

**Кафедра информационных систем**

УТВЕРЖДАЮ

Заведующий кафедрой Информационных систем

Попова Е.В.

«\_» 20 г.

**ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Студенту группы ПИ1901 Хоружая Алина Максимовна

По дисциплине Разработка приложений под мобильные устройства Тема «Разработка мобильного приложения “Цветы и уход за ними”»

Утверждена на заседании кафедры от « » 20 г. Протокол №

Срок выполнения работы с « » 20 г. по « » 20 г. Даты выдачи задания « » 20 г.

Срок сдачи первого этапа « » 20 г.

Срок сдачи второго этапа « » 20 г.

Дата сдачи проекта на кафедру « » 20 г.

Срок защиты « » 20 г.

Задание к исполнению принял студент

(ФИО., личная подпись, дата)

Руководитель работы Попок Л. Е.

(подпись, ФИО)

Краснодар 2022

**РЕФЕРАТ**

Работа содержит: 49 страниц, 17 рисунков, 7 использованных источников, 1 приложение.

Ключевые слова: FLUTTER, VS CODE, ЦВЕТЫ, МОБИЛЬНАЯ РАЗРАБОТКА, КОД.

Целью данной курсовой работы, является разработка мобильного приложения – «Цветы и уход за ними».

Результат работы: Разработанное мобильное приложение «Цветы и уход за ними».

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc104243949)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc104243950)

[1.1. Современное состояние рынка мобильных приложений 8](#_Toc104243951)

[1.2. Основные подходы к мобильной разработке мобильных приложений 11](#_Toc104243952)

[1.2.1. Нативные приложения 11](#_Toc104243953)

[1.2.2. Кроссплатформенное приложение 12](#_Toc104243954)

[1.2.3. Гибридное приложение 14](#_Toc104243955)

[1.2.4. RMAD - быстрая мобильная разработка 15](#_Toc104243956)

[1.2.5. Прогрессивная разработка 16](#_Toc104243957)

[1.3. Выбор инструментальной среды разработки 18](#_Toc104243958)

[1.4. Обоснование выбора целевой группы 19](#_Toc104243959)

[2. РАЗРАБОТКА ПРИЛОЖЕНИЯ 21](#_Toc104243960)

[2.1. Постановка задачи 21](#_Toc104243961)

[2.2. Разработка пользовательского интерфейса 21](#_Toc104243962)

[2.3. Разработка программной части приложения 26](#_Toc104243963)

[3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 29](#_Toc104243964)

[ЗАКЛЮЧЕНИЕ 34](#_Toc104243965)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 35](#_Toc104243966)

[ПРИЛОЖЕНИЕ 36](#_Toc104243967)

# ВВЕДЕНИЕ

Мобильное приложение, чаще всего называемое приложением, представляет собой тип прикладного программного обеспечения, предназначенного для запуска на мобильном устройстве, таком как смартфон или планшетный компьютер. Мобильные приложения часто служат для предоставления пользователям услуг, аналогичных тем, которые доступны на ПК (персональном компьютере). Приложения, как правило, представляют собой небольшие отдельные программные модули с ограниченными функциями. Такое использование программного обеспечения для приложений изначально было популяризировано Apple Inc. и ее AppStore, который предлагает тысячи приложений для iPhone, iPad и iPod Touch.

Приложения делятся на две большие категории: *собственные приложения и веб-приложения (нативные).* Собственные приложения создаются для конкретной мобильной операционной системы, обычно iOS или Android. Нативные приложения обладают более высокой производительностью и более тонко настроенным пользовательским интерфейсом (UI), и обычно перед их выпуском им необходимо пройти гораздо более строгий процесс разработки и контроля качества.

Веб-приложения используются в HTML5 или CSS и требуют минимального объема памяти устройства, поскольку они запускаются через браузер. Пользователь перенаправляется на определенную веб-страницу, и вся информация сохраняется в серверной базе данных. Для использования веб-приложений требуется стабильное соединение.

В настоящее время доступно несколько типов приложений:

1. **Игровые приложения:** эквивалент компьютерных видеоигр, они являются одними из самых популярных типов приложений. На их долю приходится одна треть всех загрузок приложений и три четверти всех потребительских расходов.
2. **Приложения для повышения производительности:** они направлены на повышение эффективности бизнеса за счет упрощения различных задач, таких как отправка электронной почты, отслеживание хода работы, бронирование отелей и многое другое.
3. **Приложения для образа жизни и развлечений:** они становятся все более популярными и охватывают многие аспекты личного образа жизни и социализации, такие как знакомства, общение в социальных сетях, а также обмен (и просмотр) видео. Некоторые из наиболее широко известных приложений, таких как VK, YouTube, Кинопоиск, Telegram и т.д.

Другие типы приложений включают приложения для мобильной коммерции (M-commerce), используемые для покупки товаров онлайн, такие как Ozon или Wildberries, приложения для путешествий, которые помогают путешественнику во многих отношениях (бронирование туров и билетов, поиск по картам и геолокации, дневники путешествий и т.д.), и полезные приложения, такие как приложения для здоровья и сканеры штрих-кодов.

Целю данной курсовой работы, является разработка мобильного приложения - «Цветы и уход за ними». Приложение позволит пользователю быстро получить информацию о том, как ухаживать за тем или иным комнатным растением. Это приложение, скорее справочник, который содержит в себе краткую и чёткую информацию. Аналоги у данного приложения есть, но они по большей части содержат в себе всю информацию с Википедии и различных садоводческих форумов.

Данная цель реализуется в работе на основе решения следующих задач:

* Произведение анализа объекта исследования и предметной области;
* Выбор IDE и продумывание основного функционала, который будет использован для разработки мобильного приложения;
* Описание основных этапов разработки приложения;
* Разработка мобильного приложения.

Результат работы: Разработанное мобильное приложение «Цветы и уход за ними».

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## **Современное состояние рынка мобильных приложений**

Объем мирового рынка мобильных приложений в 2018 году оценивался в $106,27 млрд. и, по прогнозам, достигнет $407,31 млрд. к 2026 году, увеличившись в среднем на 18,4% с 2019 по 2026 год.

Тенденции рынка мобильных приложений включают в себя увеличение использования различных устройств и увеличение клиентской базы для индустрии электронной коммерции, которые являются основными факторами, стимулирующими рост рынка. Кроме того, непрерывный рост корпоративных приложений, повышенное внимание к приложениям, специально используемым для здоровья и фитнеса, а также высокие загрузки и покупки игровых приложений в приложениях также оказывают свое влияние.

Ожидается, что отсутствие высокоскоростной связи в развивающихся и неразвитых регионах, и неуверенность предприятий в разработке собственных приложений, будут препятствовать росту этого рынка. Кроме того, ожидается, что рост инвестиций в цифровизацию и расширение внедрения технологий интернета вещей и смарт-объектов, подключенных к мобильным устройствам, откроют большие возможности для роста рынка в течение прогнозируемого периода.

Игровой сегмент доминировал на рынке мобильных приложений в 2018 году (см. рис. 1) и, по прогнозам, сохранит свое доминирование в течение прогнозируемого периода из-за роста числа мобильных игровых приложений в развивающихся странах, таких как Китай и Индия. Кроме того, ожидается, что сегмент развлечений и музыки будет расти значительными средними темпами в течение прогнозируемого периода из-за быстрого увеличения использования различных развлекательных приложений.

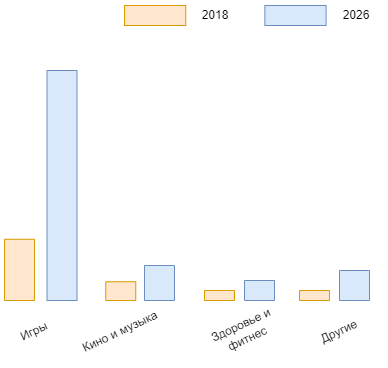


Рисунок 1 - Рынок мобильных приложений

Показатели Российского рынка мобильных приложений достаточно очевидны и понятны (данные на конец 2021г.). *В ТОП3 вошли: мобильные игры, приложения для финансов и ритейл.*

1. **Мобильные игры.** Объём рынка мобильных игр вырос до $116 млрд — это на 15% больше, чем в прошлом году. В России самой быстрорастущей игрой по количеству загрузок и по MAU стал один из лидеров прошлого года PUBG: New state, по тратам пользователей — Genshin Impact.
2. **Финансы.** Число загрузок финансовых приложений выросло 28% по всему миру, достигнув 5,9 млрд. На Россию приходится более 170 млн загрузок. Самый быстрый рост количества скачиваний показали «Тинькофф», криптовалютная биржа Binance и «Тинькофф Инвестиции». По росту MAU лидируют «Сбербанк Online», «Тинькофф» и «ВТБ Online».
3. **Ритейл.** Время, проведенное в приложениях для шоппинга, достигло 100 млрд часов по всему миру, увеличившись на 18% год к году. В России самыми быстрорастущими приложениями по количеству скачиваний стали «AliExpress Россия», Shein и «СберМегаМаркет», по MAU — «AliExpress Россия», Ozon и Wildberries.
4. **Видеостриминг**. Общее время, проведённое в видеосервисах пользователями по всему миру, выросло на 16% по сравнению с 2019 годом. Россия вышла на второе место по этому показателю — рост составил 61%. Netflix, мировой лидер по количеству скачиваний, в России оказался только на девятом месте. Лидерами стали «Кинопоиск», YouTube и Okko. Больше всего денег пользователи потратили в ivi, YouTube и Premier.
5. **Еда и напитки.** Приложения для еды и напитков достигли нового рубежа — 194 млрд сеансов в 2021 году. Рост составил 49% по сравнению с 2019 годом. В четвёртом квартале 2021 года россияне провели на 154% больше сессий в таких приложениях — более 2,1 млрд. По количеству загрузок в России на верхние строчки поднялись «Пятёрочка», «Макдональдс Россия» и Delivery Club, по MAU — «Пятёрочка», Delivery Club и «Едадил».
6. **Здоровье и фитнес.** Больше всего россияне скачивали приложения Mi Fit, Urent и Step Counter, а больше всего денег потратили в приложениях «Где мои дети», Meditopia и Flo.
7. **Социальные сети.** Благодаря *(\*с 6 марта 2022 года его работа ограничена на территории РФ)* TikTok (+90% по всему миру за пределами Китая), 70% времени пользователи потратили на социальные, фото- и/или видеоприложения. В России TikTok значительно опередил другие приложения категории — пользователи провели в нём в среднем 26,3 часа в месяц. В топ по загрузкам в России вошли TikTok *(\*с 6 марта 2022 года его работа ограничена на территории РФ)*, Telegram и WhatsApp, по тратам — VK, TikTok *(\*с 6 марта 2022 года его работа ограничена на территории РФ*) и GetContact.
8. **Путешествия.** Самый быстрый рост скачиваний в России показали DiDi, Uber Russia и «Туту.ру», по общему количеству загрузок лидерами стали «Яндекс.Карты», «Яндекс.Go» и Uber Russia.
9. **Знакомства.** Мировые расходы на приложения для знакомств превысили $4,2 млрд — это на 55% больше по сравнению с 2019 годом. В России больше всего загрузок пришлось на «Одноклассники», Badoo и Chatruletka, самые большие траты были в Tinder, «Одноклассниках» и Badoo.
10. Бизнес. Самый большой рост по абсолютному числу скачиваний в России заняло приложение «Мой Налог» — 1,27 млн. За ним следуют «СберАвто» и «Госуслуги Авто» с 1,02 млн загрузок каждое.

## **Основные подходы к мобильной разработке мобильных приложений**

Как владелец проекта, вы можете спросить себя: какой подход к мобильной разработке принесет наибольшую пользу моему видению? Ответ таков: это зависит от проекта, хотя большинство компаний, занимающихся разработкой мобильных приложений, скажут, что нативная разработка – лучшая.

Однако правда в том, что наиболее эффективный подход к разработке мобильных приложений – это тот, который может удовлетворить все ваши требования, оптимизировать время разработки и стоимость производства желаемого приложения.

Процесс разработки мобильных приложений иногда может быть сложным. Ваш список требований должен включать удобство и способность быстро вносить изменения, соответствующие потребностям вашего бизнеса и требованиям рынка. Затем вам нужно подумать о том, как охватить и привлечь каждого пользователя. Наконец, вам нужно определиться с многочисленными техническими факторами, такими как операционные системы, средства разработки, SDK и так далее.

### **Нативные приложения**

Разработка нативных приложений – это использование языков программирования, специфичных для конкретной платформы, наборов для разработки программного обеспечения и сред разработки, предлагаемых поставщиками ОС. Другими словами, если вы планируете приложение для iOS и Android, эти приложения будут разрабатываться отдельно для каждой платформы с использованием совершенно разных технологических стеков.

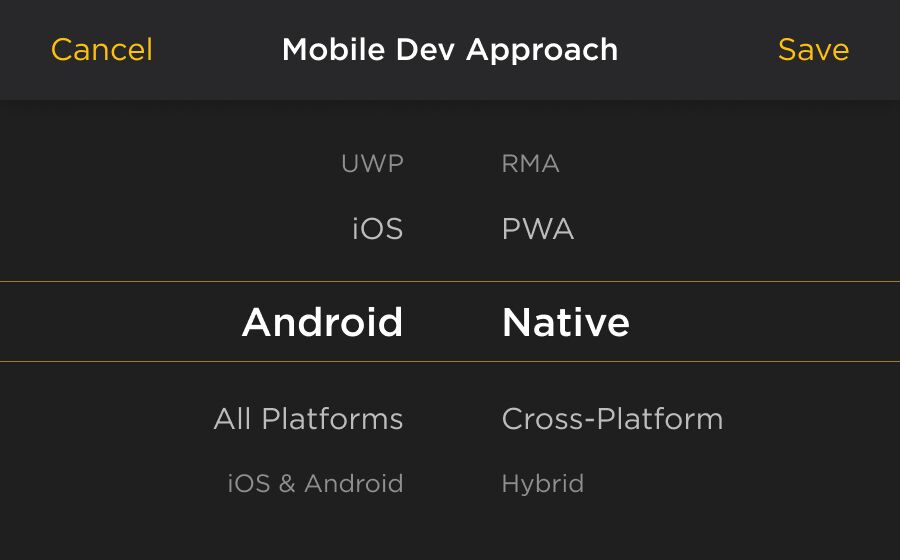


Рисунок 2 - Native App Development

Этот подход имеет значительные преимущества и остается наиболее предпочтительным среди разработчиков мобильных приложений. Прежде всего, собственные приложения поддерживают все доступные функции платформы и совместимых устройств. В то же время вы можете создать уникальный пользовательский интерфейс с помощью встроенных инструментов и быть уверенным, что каждый пользователь ОС чувствует себя как дома. Кроме того, нативные приложения обладают более высокой производительностью и быстродействием.

Преимущества Нативного подхода к разработке приложений:

* Наилучшая производительность
* Реализация пользовательского интерфейса для конкретной платформы
* 100% поддержка функций операционной системы
* Полный доступ к функциям, связанным с оборудованием
* Собственные приложения отличаются высокой надежностью, безопасностью и отзывчивостью

### **Кроссплатформенное приложение**

Кроссплатформенное мобильное приложение, как следует из названия, означает приложение, которое может работать на разных платформах. Кроссплатформенный подход является хорошей альтернативой нативной разработке, поскольку он решает проблему создания отдельного приложения для каждой мобильной платформы. Другими словами, эта концепция позволяет доставлять приложения на несколько платформ одновременно с использованием языков и инструментов, отличных от собственных наборов инструментов, предлагаемых Google и Apple.

Следуя этому подходу, разработчики используют кроссплатформенные фреймворки и инструменты, основанные на JavaScript или .NET/C#. В результате, если вы решите использовать кроссплатформенный подход - будь то для iOS, Android или веб–приложений, – вам понадобится всего одна команда с одинаковыми навыками для всех платформ. Более того, когда вам нужно выпустить все ваши приложения одновременно, вы можете сделать это без ущерба для размера команды и сроков.

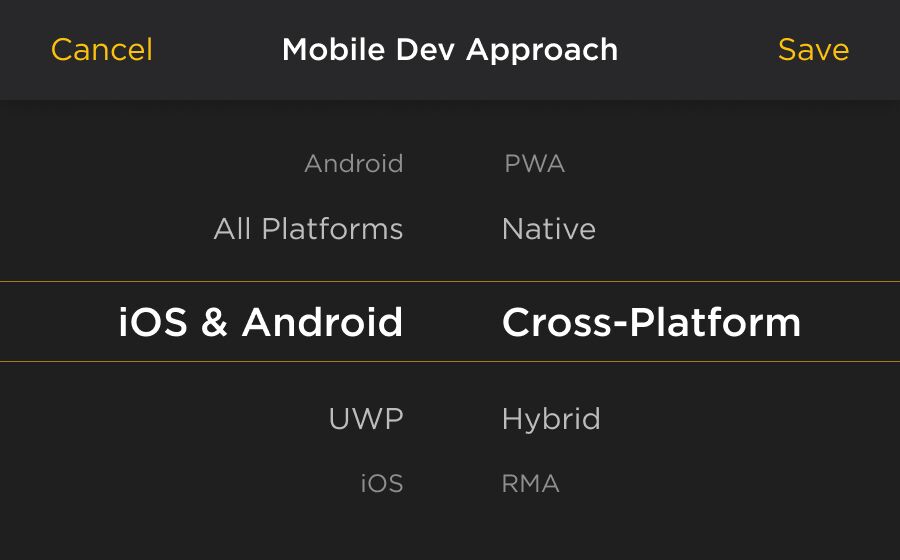


Рисунок 3 - Cross-Platform App Development

Сегодня самые популярные кроссплатформенные технологии имеют широкую поддержку сообществ и постоянно развиваются. В любом случае, они неизбежно отстают от обновлений мобильной ОС, и существует более высокий риск неясного пути обновления или серьезного переписывания кода.

Преимущества кросс-платформенного подхода к разработке:

* Единообразие на всех платформах
* Эффективен для бюджетного контроля или низкого финансирования
* Реализация проста
* Публикация может быть сразу для всех платформ
* Исходный код может быть повторно использован
* Высокий демографический охват
* Быстрое время выхода на рынок

### **Гибридное приложение**

Разработка гибридных приложений классифицируется как форма кроссплатформенности. При таком подходе к разработке ядро приложения разрабатывается с использованием стандартных веб-технологий и инструментов, таких как JavaScript, CSS и HTML5, а затем выполняется в собственной оболочке.

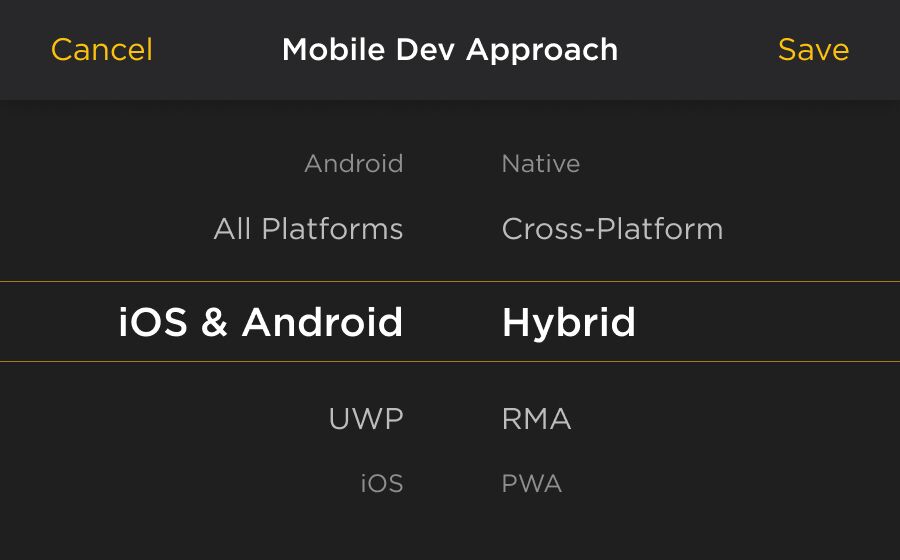


Рисунок 4 - Hybrid App Development

Приложения, созданные в результате гибридного подхода к разработке, имеют скорость обычного веб-приложения и пользовательский интерфейс, аналогичный любому типу встроенного мобильного приложения. Примечательно, что использование единой кодовой базы позволяет развертывать на всех платформах, снижая затраты по сравнению с собственными приложениями. Наконец, гибридные приложения с аппаратно-зависимыми функциями имеют доступ к аппаратным компонентам устройства и собственным библиотекам платформы.

Преимущества гибридного подхода к разработке:

* Единая команда разработчиков
* Короткое время для выхода на рынок
* Простая переносимость кода
* Возможность использования аппаратных компонентов
* Тот же пользовательский интерфейс, что и в родном мобильном приложении
* Более низкая стоимость разработки
* Возможность работы в сети и в автономном режиме

### **RMAD - быстрая мобильная разработка**

Подход быстрой разработки мобильных приложений используется для разработки кроссплатформенных приложений за короткое время. Это включает в себя использование специальных инструментов разработки без кода или с низким уровнем кода для программирования простых приложений для различных бизнес-решений.

Начнем с того, что этот подход очень похож на методологию быстрой разработки приложений, основанную на небольшом панорамировании, первоначальном прототипировании, перерабатываемых программных компонентах и использовании адаптивного процесса.



Рисунок 5 - Rapid Mobile Development

Разработчики Rapid mobile объявляют функции и возможности приложений на интерфейсе, а серверная часть переводит спецификации в код. В этом случае интерфейс использует метаданные для работы. Эффективно это делается путем обобщения основной информации о функциях приложения, таких как управляющие активами или элементы пользовательского интерфейса, в базе данных. Таким образом, это устраняет необходимость в кодировании базы данных.

Преимущества подхода RMAD:

* Низкое соотношение инвестиций к рентабельности инвестиций
* Низкая сложность процесса разработки
* Подходит для нескольких типов проектов
* Разработка с низким кодом/без кода
* Разработчикам требуется мало опыта или он вообще не требуется
* Многоразовая кодовая база

### **Прогрессивная разработка**

Последний тип подхода к разработке, но не менее важный, который мы не можем исключить, - это прогрессивная разработка веб-приложений. В этом случае использование HTML здесь является основным.

PWA на самом деле не являются мобильными приложениями, но по внешнему виду и функциям они похожи на любое собственное приложение. Это веб-приложения, специально разработанные с адаптируемым и удобным интерфейсом для мобильных пользователей.

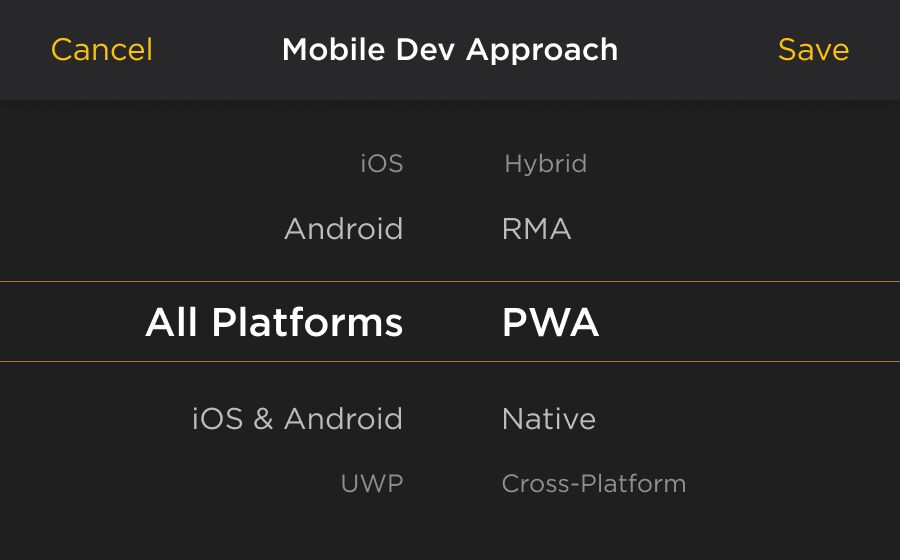


Рисунок 6 - Progressive Web Apps

Пользователю вообще не нужно загружать полученное приложение из app Store. Прогрессивные веб-приложения работают без обычного браузера и могут быть установлены на главном экране любого мобильного или настольного устройства. Прелесть здесь в том, что они настроены так, чтобы быть отзывчивыми и подключаться независимо.

С их скрытым браузером у них есть недостаток в том, что они ограничены в реализации, потому что вам нужны API-интерфейсы поставщика браузера для доступа к функциям мобильного оборудования.

Преимущества прогрессивных веб-приложений:

* Простое обслуживание приложения
* Использование единой кодовой базы
* Удобный для мобильных устройств пользовательский интерфейс
* Вам не нужен магазин приложений для размещения или загрузки

Приложения отличаются друг от друга по-своему. Если ваша цель и желание - разработать долгосрочный проект, приложение, которое будет отзывчивым, будет иметь превосходное качество пользовательского интерфейса, плавную производительность и надежную безопасность, тогда очень рекомендуется использовать собственный подход к разработке. С другой стороны, если вы хотите пойти на компромисс в бюджетных планах или хотите ускорить разработку, вы можете рассмотреть кросс-платформенный подход.

## **Выбор инструментальной среды разработки**

Visual Studio Code с открытым исходным кодом, с обширной библиотекой расширений, которая эффективно превращает легкий кроссплатформенный редактор в полноценную интегрированную среду разработки (IDE), – вариант № 1 для многих разработчиков.

Код Флаттера/Дротика

Портативный инструментарий пользовательского интерфейса Google с открытым исходным кодом для создания изначально скомпилированных мобильных приложений (а теперь и веб-приложений и даже настольных приложений) вызвал настоящий ажиотаж в отрасли из-за своего нового подхода.

Flutter также использует свои собственные виджеты, вместо того, чтобы пытаться предоставлять собственные виджеты, как это делают многие другие альтернативные фреймворки. Наряду с "stateful hot reload" Flutter использует функции аппаратного ускорения для обеспечения графики без сбоев и подергиваний на собственных скоростях.

Учитывая все это, неудивительно, что Flutter является мобильным расширением № 1 на рынке Visual Studio Code Marketplace. Он может похвастаться более чем 1,9 миллионами загрузок и отличной оценкой 5.0 от 13 разработчиков, которые его использовали. Полная документация по проекту содержит все, что вам нужно знать, включая образцы и учебные пособия.

Android для VS Code. Это предварительный просмотр, помогающий разработчикам мобильных приложений Android "устанавливать, запускать и отлаживать приложения для Android из среды VS Code".

Особенности Android для VS Code включают в себя:

* Пошаговое выполнение кода построчно
* Точки останова
* Переменная проверка и модификация
* Просмотр Logcat [ Палитра команд -> Android: Просмотр Logcat ]
* Перерыв в исключениях
* Пошаговое руководство по исходным кодам Android

Хотя для этого не требуется установка Android Studio, для этого требуется установка инструментов платформы Android SDK.

В записи Marketplace подчеркивается, что этот инструмент является предварительным просмотром, поэтому он имеет некоторые ограничения. Он был установлен более 57 000 раз. Разработчик этого проекта – Дэйв Холлоуэй.

## **Обоснование выбора целевой группы**

API Level это просто целое число, которое однозначно идентифицирует ревизию рабочего окружения программных библиотек (framework API revision), которая предоставляется версией платформы (операционной системой) Android.

Платформа Android предоставляет framework API, который могут использовать приложения для взаимодействия с нижележащей системой Android. Framework API состоит из:

* Базового набора пакетов и классов.
* Набора элементов XML и атрибутов для декларирования в файле манифеста (manifest file).
* Набора элементов XML и атрибутов для декларирования и доступа к ресурсам.
* Набора намерений (Intents).
* Набора разрешений доступа к ресурсам устройства и системы, которые приложение может запросить, а также защита ограничения доступа, встроенная в систему.

Каждая последующая версия платформы Android может включать в себя обновления предоставляемого framework API приложения Android.

Минимальная версия используемого SDK (minSdkVersion) или версия API в приложении – 23. Данная версия соответствует Android версии 6.0 (KitKat). На данный момент, используя эту версию, можно охватить порядка 95% пользователей.

Таблица 1 - Версии Android и уровни API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Версия** | **Уровень API** | **Выпущено** | **Код версии сборки** |
| Q | 10.0 | 29 | Август 2020 г. | BuildVersionCodes.Q |
| Pie | 9.0 | 28 | Август 2018 г. | BuildVersionCodes.P |
| Oreo | 8.1 | 27 | Декабрь 2017 г. | BuildVersionCodes.OMr1 |
| Oreo | 8.0 | 26 | Авг 2017 | BuildVersionCodes.O |
| Nougat | 7.1 | 25 | Декабрь 2016 г. | BuildVersionCodes.NMr1 |
| Nougat | 7,0 | 24 | Август 2016 г. | BuildVersionCodes.N |
| Marshmallow | 6.0 | 23 | Авг 2015 | BuildVersionCodes.M |
| Lollipop | 5.1 | 22 | Мар 2015 | BuildVersionCodes.LollipopMr1 |
| Lollipop | 5.0 | 21 | Ноя 2014 | BuildVersionCodes.Lollipop |
| Kitkat Watch | 4.4W | 20 | Июнь 2014 г. | BuildVersionCodes.KitKatWatch |
| Kitkat | 4.4. | 19 | Окт 2013 | BuildVersionCodes.KitKat |
| Желе Боб | 4.3 | 18 | Июль 2013 г. | BuildVersionCodes.JellyBeanMr2 |
| Желе Боб | 4.2-4.2.2 | 17 | Ноябрь 2012 г. | BuildVersionCodes.JellyBeanMr1 |

Сопутствующие минимальные характеристики устройств: экраны >= 2,5 дюймов в размере физической диагонали; плотность >=100 dpi; соотношение сторон быть между 1,333 (4: 3) и 1,779 (16: 9).

Целевая аудитория данного приложения охватывает возраст 12-14+, имеющих какое-либо отношение к разведению растений, либо желающих это начать.

# РАЗРАБОТКА ПРИЛОЖЕНИЯ

## **Постановка задачи**

Главной задачей данного курсового проекта является разработка приложения, которое поможет быстро и без каких-либо затрат изучить уход за тем или иным домашним растением.

Приложение «Цветы и уход за ними» будет представлять собой многоэкранное приложение с функцией поиска. После клика на карточку с тем или иным растением, открывается еще один экран с более подробной информацией по уходу.

## **Разработка пользовательского интерфейса**

Пользовательский интерфейс в VS Code на языке программирования Flutter конструируется в основных файлах проекта с расширением \*.dart. Создаем новый проект (см. рис. 7):

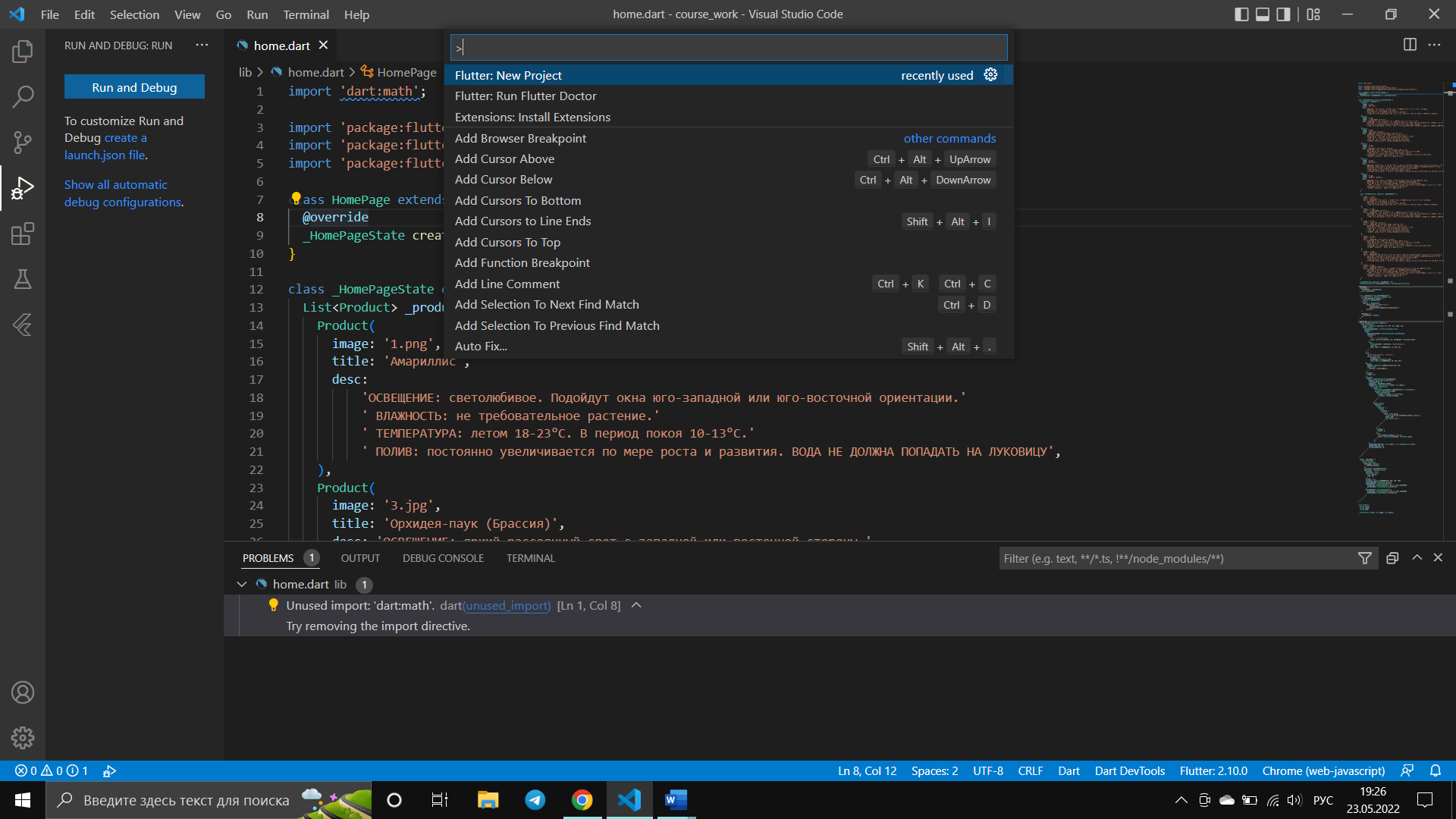


Рисунок 7 - создание нового Flutter-проекта

Теперь приступим к созданию первой и самой главной страницы нашего приложения. Она будет содержать: название приложения в верхней части, строку поиска и список из 10 карточек с цветами и растениями, который пролистывается (см. рис. 8):

Код формирования шапки экрана и расположения карточек с растениями и цветами:

Widget build(BuildContext context) {

return Container(

margin: EdgeInsets.only(top: 18, left: 14, right: 14),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

Text(

'Это я - твой помощник',

style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),

),

Icon(

IconData(0xe0ef, fontFamily: 'MaterialIcons'),

size: 40,

color: Color.fromARGB(255, 31, 106, 85),

)

],

),

Text(

'по уходу за цветами и растениями',

style: TextStyle(

fontSize: 25,

fontWeight: FontWeight.bold,

color: Color.fromARGB(255, 20, 109, 91)),

),

Padding(

padding: EdgeInsets.symmetric(vertical: 35),

child: Row(

children: [\_searchBox()],

),

),

SizedBox(

height: 24,

),

//формирование карточек

Expanded(

child: StaggeredGridView.countBuilder(

physics: BouncingScrollPhysics(),

crossAxisCount: 4,

itemCount: \_foundPlant.length,

itemBuilder: (BuildContext context, int index) {

return GestureDetector(

onTap: () {

FocusScope.of(context).requestFocus(new FocusNode());

Navigator.of(context).push(

MaterialPageRoute(

builder: (context) => DetailPage(

product: \_products[index],

),

),

);

},

child: Column(

children: [

Stack(

children: [

Stack(

children: [

Center(

child: Image.asset(

'assets/images/${\_foundPlant[index]['image']}',

width: 200,

fit: BoxFit.cover,

),

),

],

),

],

),

SizedBox(

height: 8,

),

Text(

'${\_foundPlant[index]['title']}',

style: TextStyle(fontWeight: FontWeight.bold),

),

],

));

},

staggeredTileBuilder: (int index) => new StaggeredTile.fit(2),

mainAxisSpacing: 24,

crossAxisSpacing: 24,

),

)

],

),

);

}

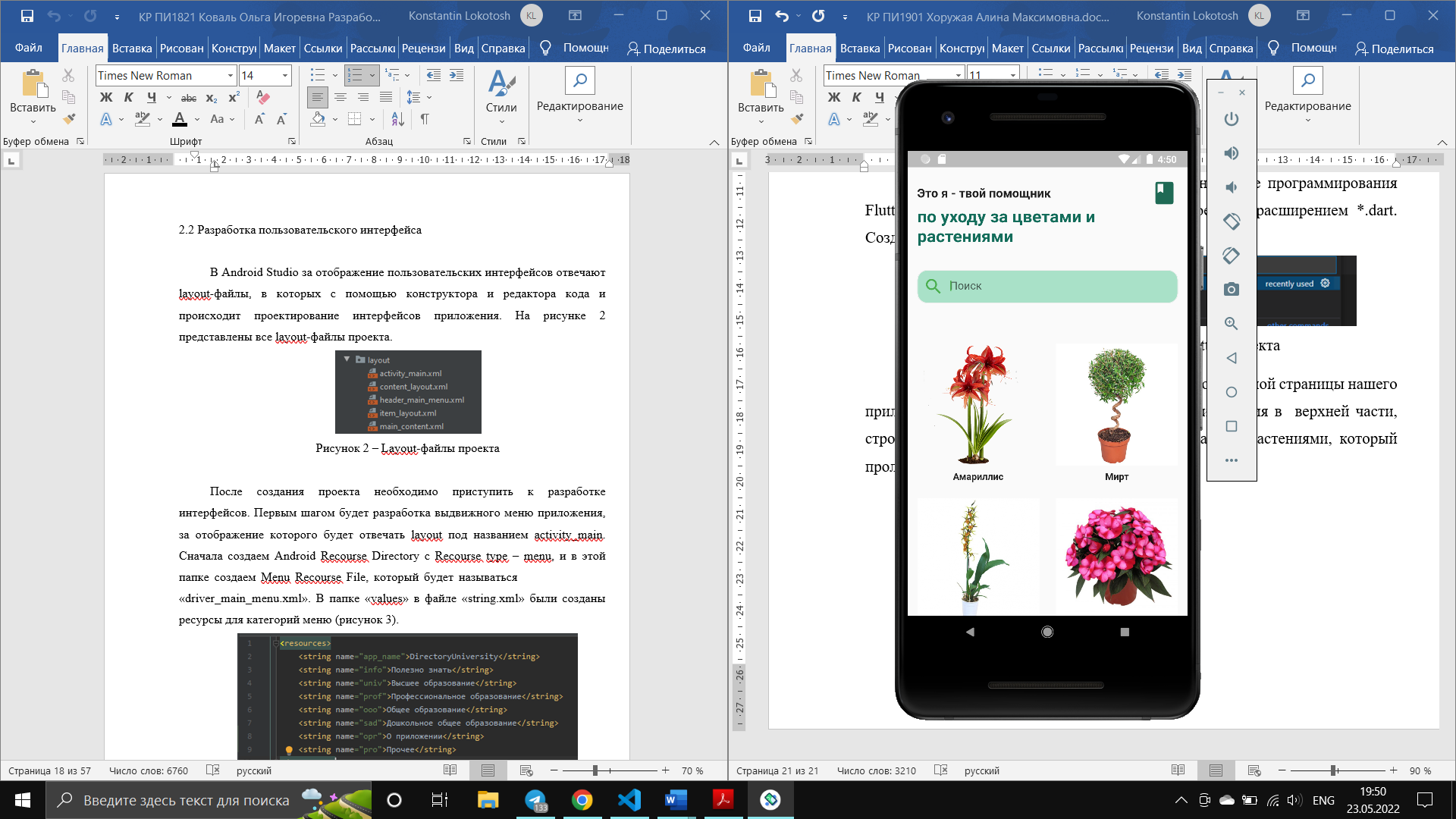


Рисунок 8 - Начальный экран

Код создания строки поиска:

  Widget \_searchBox() {

    return Expanded(

      child: TextField(

        onChanged: (value) {

          \_runFilter(value);

        },

        controller: editingController,

        decoration: InputDecoration(

          hintText: 'Поиск',

          prefixIcon: Icon(

            Icons.search,

            size: 30,

          ),

          filled: true,

          fillColor: Color.fromARGB(255, 169, 225, 200),

          contentPadding: EdgeInsets.all(15),

          enabledBorder: OutlineInputBorder(

            borderSide: BorderSide(color: Colors.grey.shade200),

            borderRadius: BorderRadius.circular(15),

          ),

          focusedBorder: OutlineInputBorder(

            borderSide: BorderSide(color: Colors.grey.shade200),

            borderRadius: BorderRadius.circular(15),

          ),

        ),

      ),

    );

  }

}

При клике на карточку с цветком или растением, открывается следующий экран с более подробным описанием этого организма (см. рис. 9). По причине того, что эти карточки представляют из себя список (List) рассмотрим в качестве примера 1 карточку любого растения.

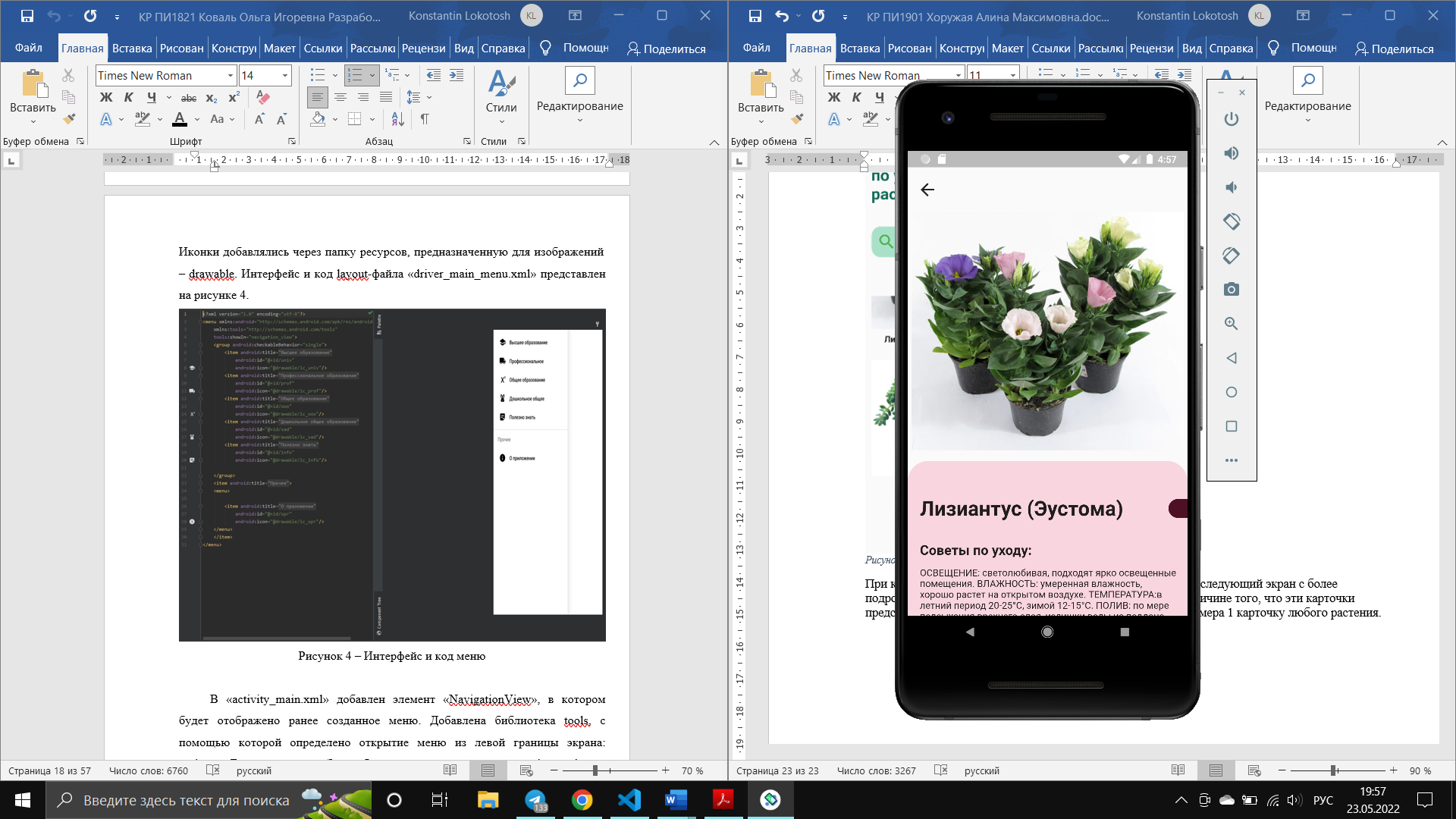


Рисунок 9 - карточка цветка Лизиантуса

Карточка растения состоит из следующих виджетов: иконка – кнопка возврата к главной странице, изображение растения, его название и блок с текстом.

Код содержимого карточки:

Product(

      image: '8.jpg',

      title: 'Лизиантус (Эустома)',

      desc:

          'ОСВЕЩЕНИЕ: светолюбивая, подходят ярко освещенные помещения.'

          ' ВЛАЖНОСТЬ: умеренная влажность, хорошо растет на открытом воздухе.'

          ' ТЕМПЕРАТУРА: в летний период 20-25°C, зимой 12-15°C.'

          ' ПОЛИВ: по мере подсыхания верхнего слоя, излишки воды из поддона слить, не допускать переувлажнение почвы.'

          ' ПОДКОРМКА: в период активного роста и цветения подкармливают комплексными минеральными удобрениями для цветущих растений 2 раза в месяц.'

          ' ОБРЕЗКА: отцветшие стебли срезают, оставляя не менее 2х пар листьев.',

## **Разработка программной части приложения**

Теперь необходимо дополнить созданные страницы интерфейсов кодом, который и будет обрабатывать действия пользователей приложения. Для начала проектируется файл «main.dart», который является более навигационным. Подключается библиотека flutter/material.dart, так же подключается файл «home.dart»

Файл «home.dart» со всеми основными элементами: список карточек, содержимое карточки, строка поиска и функция осуществления поиска. Подключаются следующие библиотеки:

import 'package:flutter/material.dart';

import 'package:flutter\_staggered\_grid\_view/flutter\_staggered\_grid\_view.dart';

import 'package:flutter/material.dart';

import 'home.dart';

Для поиска был создан: final List<Map<String, dynamic>> \_plantInform = []

А функция динамического поиска, позволяющая искать не полными названиями а по буквам, имеет следующий вид:

List<Map<String, dynamic>> \_foundPlant = [];

TextEditingController editingController = TextEditingController();

@override

initState() {

\_foundPlant = \_plantInform;

super.initState();

}

void \_runFilter(String enteredKeyword) {

List<Map<String, dynamic>> results = [];

if (enteredKeyword.isEmpty) {

results = \_plantInform;

} else {

results = \_plantInform

.where((user) => user['title']

.toLowerCase()

.contains(enteredKeyword.toLowerCase()))

.toList();

}

setState(() {

\_foundPlant = results;

});

}

Этот блок позволяет формировать карточки удобным списком друг за другом:

Expanded(

child: StaggeredGridView.countBuilder(

physics: BouncingScrollPhysics(),

crossAxisCount: 4,

itemCount: \_foundPlant.length,

itemBuilder: (BuildContext context, int index) {

return GestureDetector(

onTap: () {

FocusScope.of(context).requestFocus(new FocusNode());

Navigator.of(context).push(

MaterialPageRoute(

builder: (context) => DetailPage(

product: \_products[index],

),

),

);

},

child: Column(

children: [

Stack(

children: [

Stack(

children: [

Center(

child: Image.asset(

'assets/images/${\_foundPlant[index]['image']}',

width: 200,

fit: BoxFit.cover,

),

),

],

),

],

),

SizedBox(

height: 8,

),

Text(

'${\_foundPlant[index]['title']}',

style: TextStyle(fontWeight: FontWeight.bold),

),

],

));

},

staggeredTileBuilder: (int index) => new StaggeredTile.fit(2),

mainAxisSpacing: 24,

crossAxisSpacing: 24,

),

)

Файл «detail.dart» отвечает за то, как будет выглядеть карточка цветка или растения. Подключаются следующие библиотеки и файлы:

import 'package:flutter/material.dart';

import 'home.dart';

Блок подключения изображения к карточке выглядит так:

Widget \_productImage() {

return Stack(

children: [

Positioned.fill(

child: Align(

alignment: Alignment.bottomCenter,

child: Container(

height: 50,

width: 300,

),

),

),

Center(

child: Image.asset(

'assets/images/${product.image}',

height: 350,

fit: BoxFit.cover,

),

),

],

);

}

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

После загрузки приложения, перед пользователем открывается главная страница (см. рис. 10). Интерфейс довольно прост: список карточек с картинками и названиями цветов (растений) и строка поиска, расположенная в верхней части страницы.

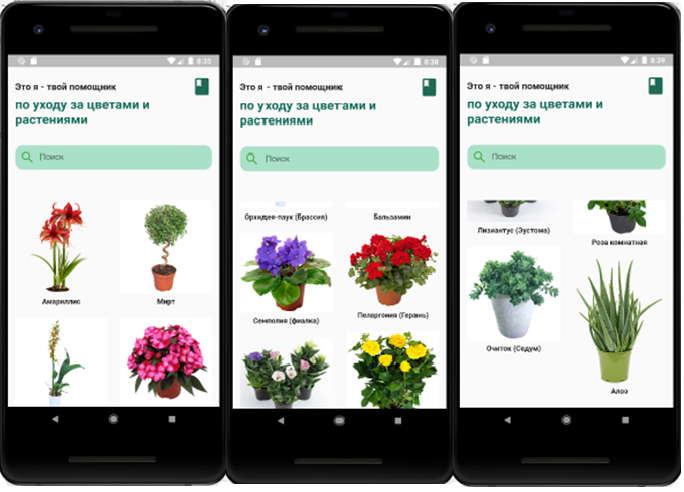


Рисунок 10 - Главный экран приложения

Теперь вы можете либо открыть карточку с растением сразу, либо осуществив поиск. Динамический поиск позволяет видеть результаты не только по введенным словам, но и буквам, не только в начале названия, но и в любой его части. Например, введет сочетание букв «МА». Сейчас в приложении есть 10 карточек со следующими растениями\цветами: «Амариллис», «Мирт», «Бальзамин», «Орхидея-паук (Брассия)», «Сенполия (Фиалка)», « Пеларгония (Герань)», «Лизиантус (Эустома)», «Роза комнатная», «Очиток (Седум)», «Алоэ».

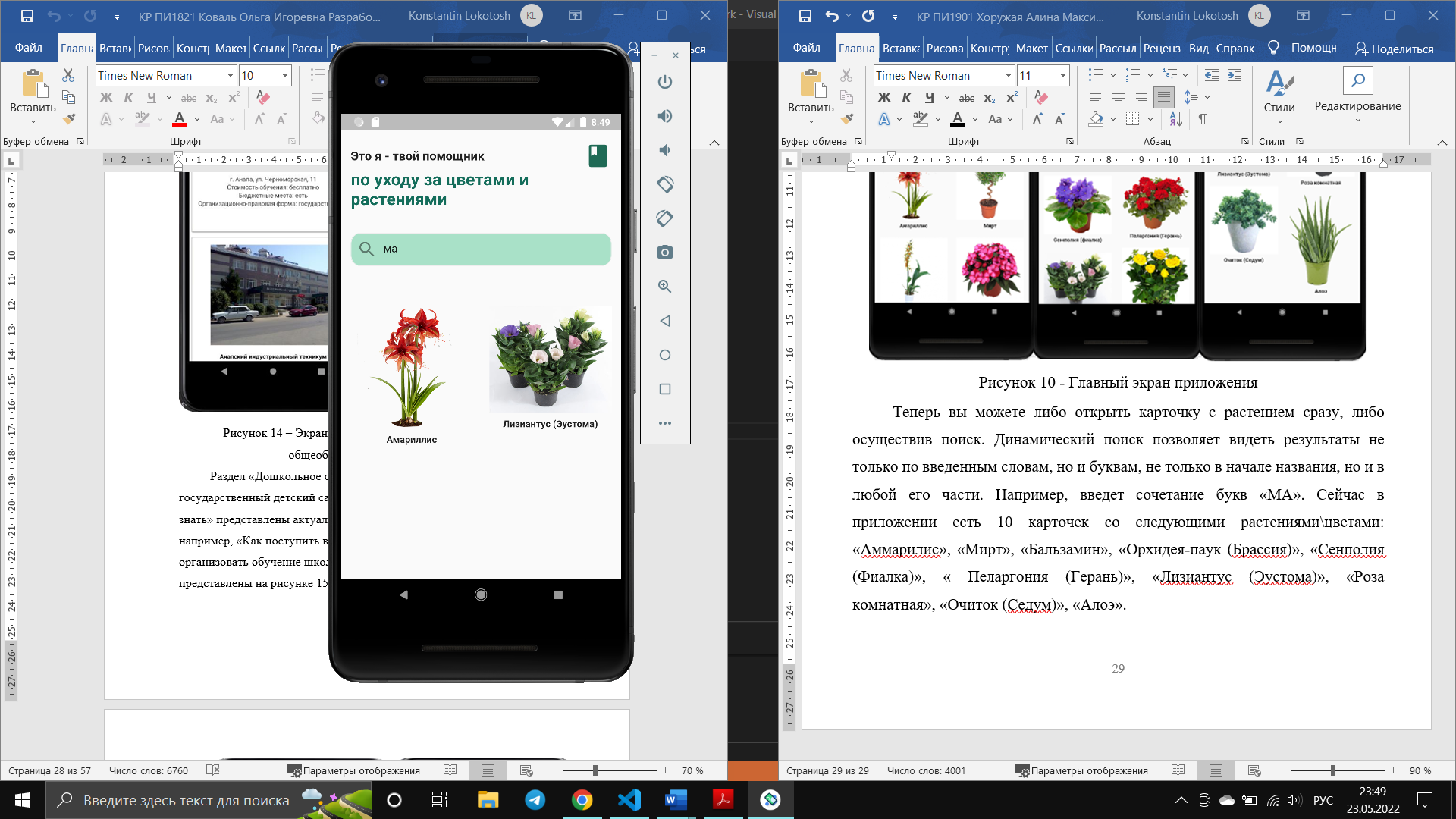


Рисунок 11 - Результат поиска

Как мы видим, он определил две карточки «Амариллис» и «Лизиантус (Эустома)». Теперь пользователь может открыть карточку найденных растений (см. рис. 12):

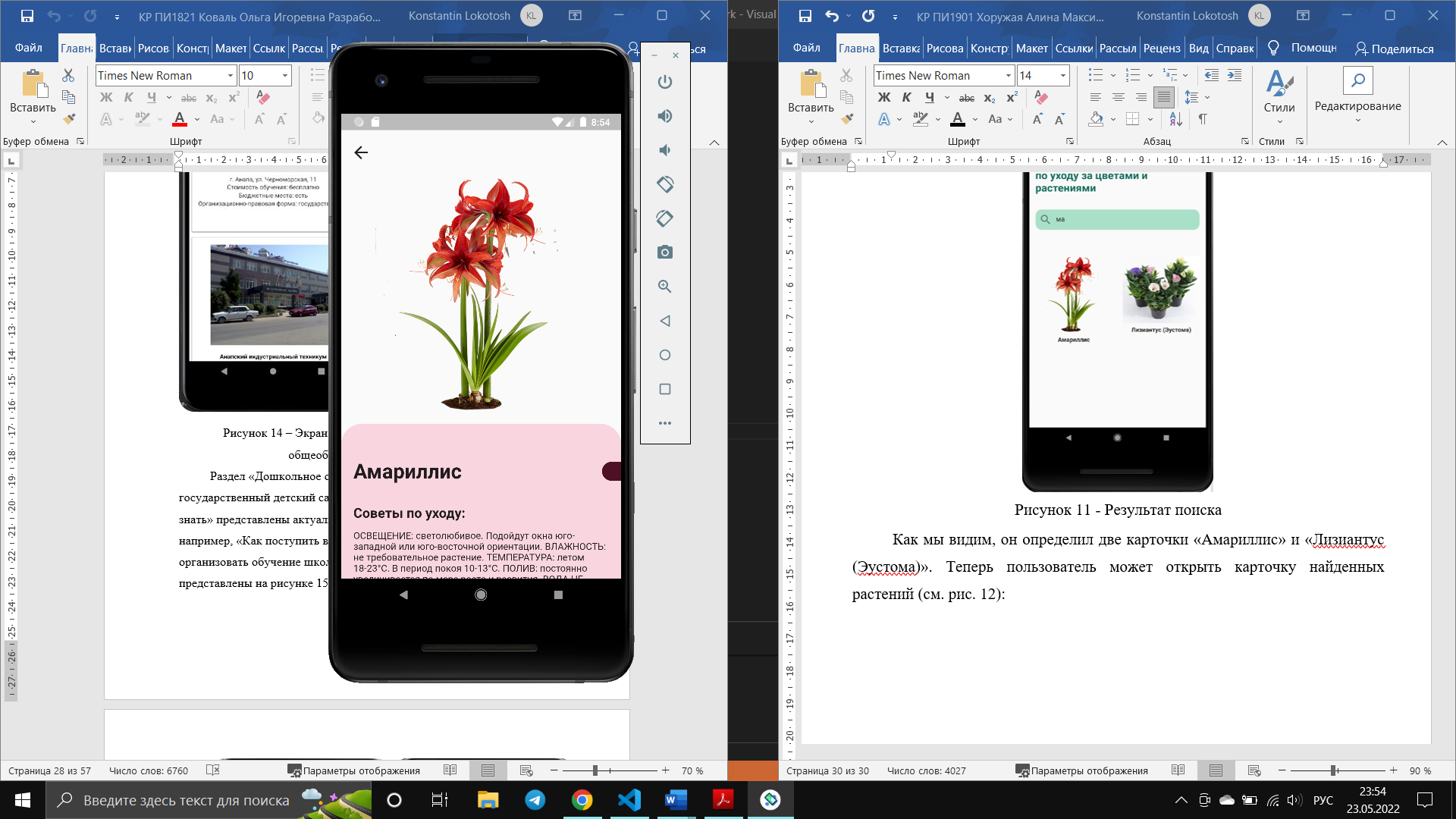


Рисунок 12 - содержимое карточки "Амариллис"

В случае, если содержимого карточки много (как в нашем случае), будет активирован скроллинг области описания (см. рис. 13)

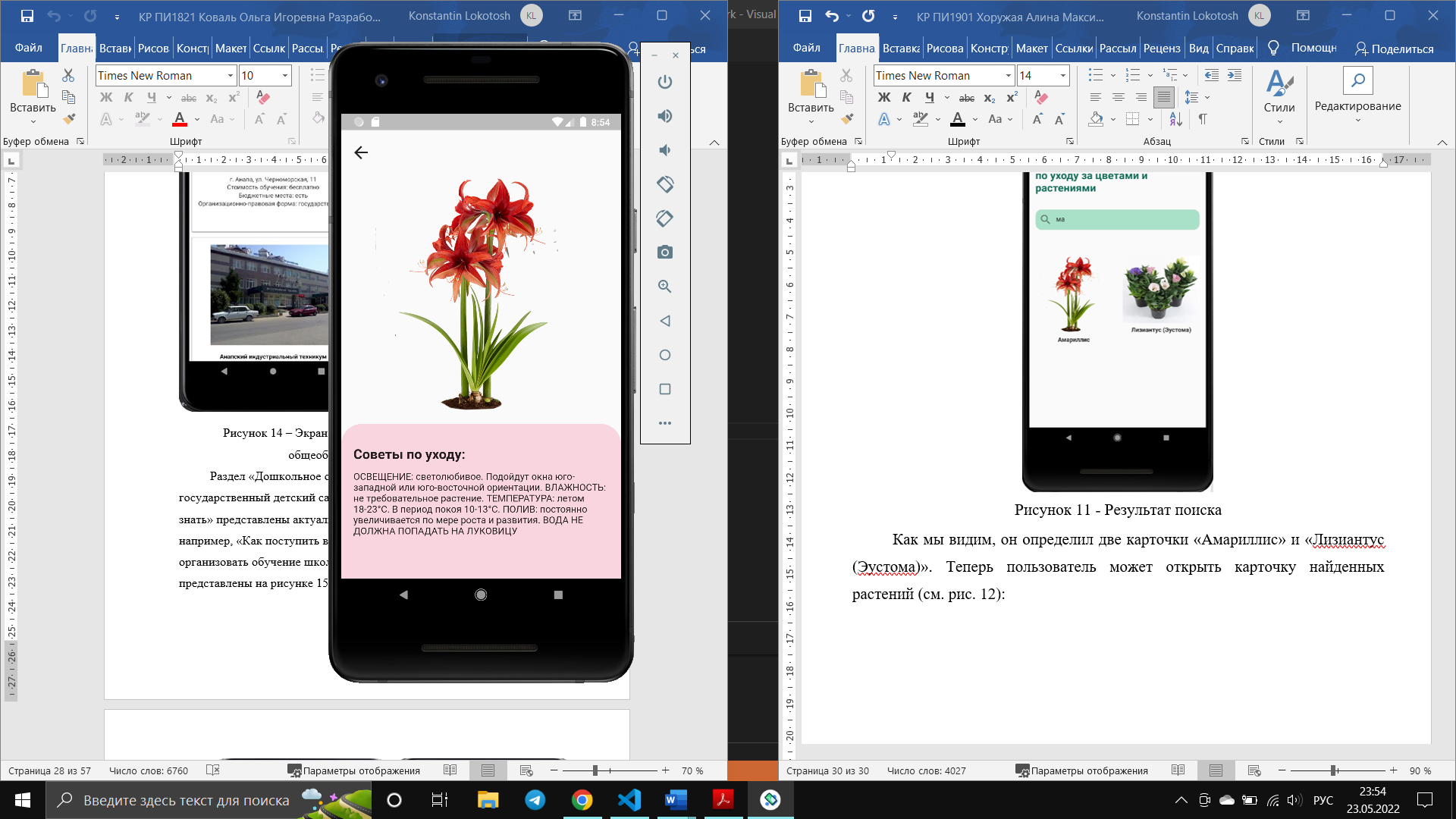


Рисунок 13 – продолжение содержимого карточки "Амариллис"

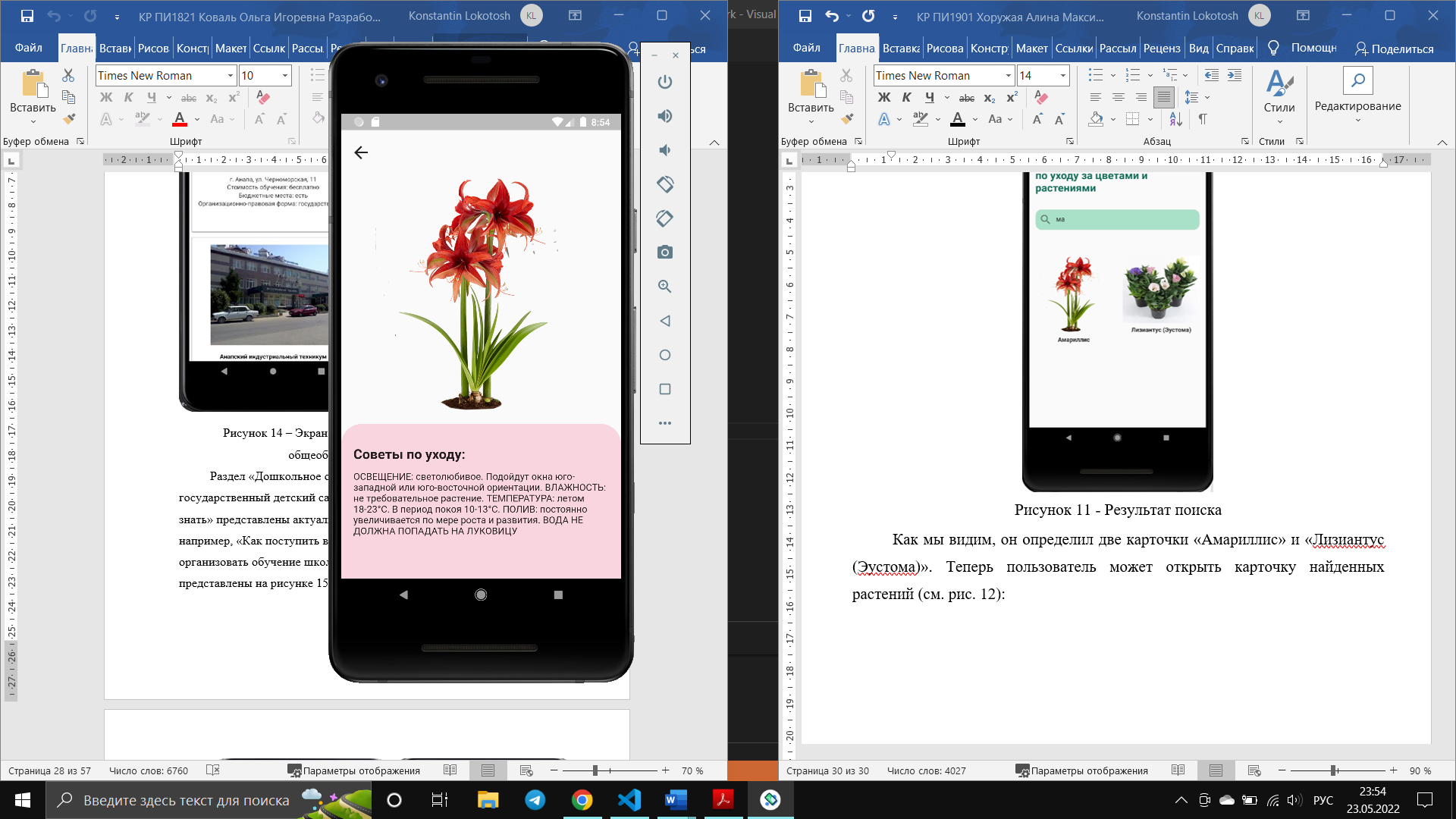


Рисунок 14 - Кнопка возврата к главной странице

Теперь будет продемонстрировано содержимое оставшихся 9 карточек (они построены аналогичным образом):

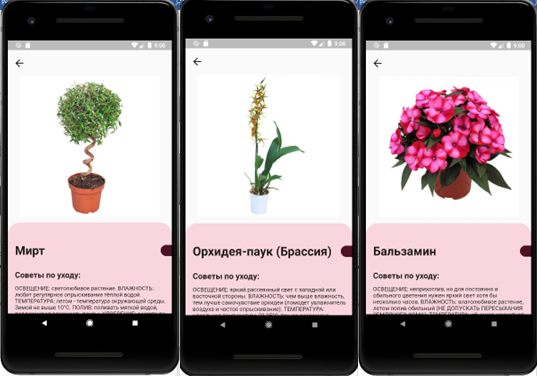


Рисунок 15 - содержимое карточек (1)

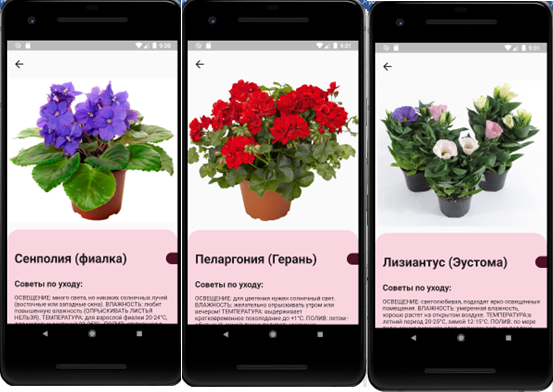


Рисунок 16 - содержимое карточек (2)

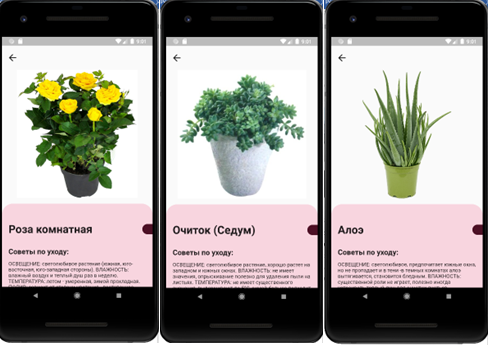


Рисунок 17 - содержимое карточек (3)

# ЗАКЛЮЧЕНИЕ

Подводя итоги выполнения курсовой работы, стоит отметить, что был проведен анализ современного состояния рынка, исследованы современные подходы в разработке мобильных приложений (выделены их преимущества и недостатки), изучены различные среды для разработки.

Разработанное приложение поможет легко ухаживать за цветами и растениями. Минималистичный интерфейс и дизайн просты в освоении и не вызывают трудностей в использовании, это значит, что абсолютно любой пользователь сможет разобраться и использовать приложение по назначению.

Поставленные задачи решены, а цель – создать мобильное приложение на базе ОС Andriod «Цветы и уход за ними» - достигнута.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Голощапов, Алексей Google Android. Программирование для мобильных устройств / Алексей Голощапов. - М.: БХВ-Петербург, 2012. - 448 c.
2. Дэрси, Лорен Android за 24 часа. Программирование приложений под операционную систему Google / Лорен Дэрси , Шейн Кондер. - М.: Рид Групп, 2011. - 464 c.
3. Официальный сайт с документацией Flutter | <https://flutter.dev> / [Электронный ресурс].
4. Beginning Flutter with Dart [Электронный ресурс] / Sanjib Sinha, 2022г.
5. Flutter на практике [Электронный ресурс] / Заметти Франк - ДМК-Пресс, 2020г.
6. Гриффитс Дон, Гриффитс Дэвид Г85 Head First. Программирование для Android. — СПб.: Питер, 2016. — 704 с.: ил. — (Серия «Head First O’Reilly»).
7. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.

# ПРИЛОЖЕНИЕ

**Содержимое файла: main.dart**

import 'package:flutter/material.dart';

import 'home.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Справочник растений и цветов',

theme: ThemeData(

primarySwatch: Colors.green,

),

debugShowCheckedModeBanner: false,

home: MainPage(),

);

}

}

class MainPage extends StatefulWidget {

@override

MainPageState createState() => MainPageState();

}

enum TabItem { home, favorite, notification, account }

class MainPageState extends State<MainPage> {

TabItem \_currentItem = TabItem.home;

@override

Widget build(BuildContext context) {

return Scaffold(

body: SafeArea(

child: \_buildPage(),

),

);

}

Widget \_buildPage() {

switch (\_currentItem) {

case TabItem.home:

return HomePage();

case TabItem.favorite:

// return; Favorite

case TabItem.notification:

// return; Notification

case TabItem.account:

// return; Account

default:

return HomePage();

}

}

}

**Содержимое файла: home.dart**

import 'package:flutter/material.dart';

import 'package:flutter\_staggered\_grid\_view/flutter\_staggered\_grid\_view.dart';

class HomePage extends StatefulWidget {

@override

\_HomePageState createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

List<Product> \_products = [

Product(

image: '1.png',

title: 'Амариллис',

desc:

'ОСВЕЩЕНИЕ: светолюбивое. Подойдут окна юго-западной или юго-восточной ориентации.'

' ВЛАЖНОСТЬ: не требовательное растение.'

' ТЕМПЕРАТУРА: летом 18-23°C. В период покоя 10-13°C.'

' ПОЛИВ: постоянно увеличивается по мере роста и развития. ВОДА НЕ ДОЛЖНА ПОПАДАТЬ НА ЛУКОВИЦУ',

),

Product(

image: '2.jpg',

title: 'Мирт',

desc:

'ОСВЕЩЕНИЕ: светолюбивое растение.'

' ВЛАЖНОСТЬ: любит регулярное опрыскивание тёплой водой.'

' ТЕМПЕРАТУРА: летом - температура окружающей среды. Зимой не выше 10°C.'

' ПОЛИВ: поливать мягкой водой, поддерживая влажность почвы.'

' УДОБРЕНИЕ: с марта по август удобряют раз в месяц комплексным минеральным удобрением.'

' ОБРЕЗКА: должна быть регулярной и планомерной.',

),

Product(

image: '3.jpg',

title: 'Орхидея-паук (Брассия)',

desc:

'ОСВЕЩЕНИЕ: яркий рассеянный свет с западной или восточной стороны.'

' ВЛАЖНОСТЬ: чем выше влажность, тем лучше самочувствие орхидеи (помодет увлажнитель воздуха и частое опрыскивание).'

' ТЕМПЕРАТУРА: идеальная температура 23-25°C, так же разрешена комнатная.'

' ПОЛИВ: метод погружения горшка в сосуд с водой на 20-30 минут (НЕЛЬЗЯ ДОПУСКАТЬ СМОРЩИВАНИЯ ПСЕВДОБУЛЬБ)',

),

Product(

image: '4.jpg',

title: 'Бальзамин',

desc:

'ОСВЕЩЕНИЕ: неприхотлив, но для постоянно и обильного цветения нужен яркий свет хотя бы несколько часов.'

' ВЛАЖНОСТЬ: влаголюбивое растение, летом полив обильный (НЕ ДОПУСКАТЬ ПЕРЕСЫХАНИЯ ЗЕМЛЯНОГО КОМА).'

' ТЕМПЕРАТУРА: обычная комнатная температура (зимой не ниже 12-14°C).'

' ПОЛИВ: поливать мягкой водой, поддерживая влажность почвы.'

' УДОБРЕНИЕ: подкармливать с весны до осени каждые 2 недели минеральным удобрением для цветущих растений.',

),

Product(

image: '5.jpg',

title: 'Сенполия (фиалка)',

desc:

'ОСВЕЩЕНИЕ: много света, но никаких солнечных лучей (восточные или западные окна).'

' ВЛАЖНОСТЬ: любит повышенную влажность (ОПРЫСКИВАТЬ ЛИСТЬЯ НЕЛЬЗЯ).'

' ТЕМПЕРАТУРА: для взрослой фиалки 20-24°C, для молодых растений 23-25°C .'

' ПОЛИВ: отстоянная в открытой посуде водопроводная вода (вода не должна попадать на листья и в центр розетки).'

' УДОБРЕНИЕ: примерно через 1.5 месяца после пересадки.'

' ОБРЕЗКА: должна быть регулярной и планомерной.',

),

Product(

image: '6.png',

title: 'Пеларгония (Герань)',

desc:

'ОСВЕЩЕНИЕ: для цветения нужен солнечный свет.'

' ВЛАЖНОСТЬ: желательно опрыскивать утром или вечером!'

' ТЕМПЕРАТУРА: выдерживает кратковременное похолодание до +1°C.'

' ПОЛИВ: летом - обильный, зимой лучше поливать умеренно.'

' ПОДКОРМКА: раз в 2 недели жидким минеральным удобрением.'

' ОБРЕЗКА: ранней весной до начала вегетационного периода.',

),

Product(

image: '7.png',

title: 'Роза комнатная',

desc:

'ОСВЕЩЕНИЕ: светолюбивое растение (южная, юго-восточная, юго-западная стороны).'

' ВЛАЖНОСТЬ: влажный воздух и теплый душ раз в неделю.'

' ТЕМПЕРАТУРА: летом - умеренная, зимой прохладная.'

' ПОЛИВ: зависит от степени цветения - постепенное увеличение, осенью полив прекращается.'

' ПОДКОРМКА: раз в неделю во время роста и цветения удобрением для растущих растений.'

' ОБРЕЗКА: обрезка побегов осенью, примерно на 1/3 оставляя на каждой ветке по 5 живых почек.',

),

Product(

image: '8.jpg',

title: 'Лизиантус (Эустома)',

desc:

'ОСВЕЩЕНИЕ: светолюбивая, подходят ярко освещенные помещения.'

' ВЛАЖНОСТЬ: умеренная влажность, хорошо растет на открытом воздухе.'

' ТЕМПЕРАТУРА:в летний период 20-25°C, зимой 12-15°C.'

' ПОЛИВ: по мере подсыхания верхнего слоя, излишки воды из поддона слить, не допускать переувлажнение почвы.'

' ПОДКОРМКА: в период активного роста и цветения подкармливают комплексными минеральными удобрениями для цветущих растений 2 раза в месяц.'

' ОБРЕЗКА: отцветшие стебли срезают, оставляя не менее 2х пар листьев.',

),

Product(

image: '9.jpg',

title: 'Очиток (Седум)',

desc:

'ОСВЕЩЕНИЕ: светолюбивое растение, хорошо растет на западном и южных окнах.'

' ВЛАЖНОСТЬ: не имеет значения, опрыскивание полезно для удаления пыли на листьях.'

' ТЕМПЕРАТУРА: не имеет существенного значения, выдерживают до 5°C, зимой больше подходит прохладное место с температурой 8-12°C.'

' ПОЛИВ: поливать умеренно, увлажнять по мере подсыхания почвы, пересушивание не вредит.'

' ПОДКОРМКА: весной и летом в период активного роста, достаточно 1 раза в месяц, используют удобрение для кактусов и суккулентов.',

),

Product(

image: '10.png',

title: 'Алоэ',

desc:

'ОСВЕЩЕНИЕ: светолюбивое, предпочитает южные окна, но не пропадает и в тени -в темных комнатах алоэ вытягивается, становится бледным.'

' ВЛАЖНОСТЬ: существенной роли не играет, полезно иногда устраивать теплый душ для очистки листьев.'

' ТЕМПЕРАТУРА: летом температура должна быть умеренной, зимой прохладной, выдерживает температуру до 6°C.'

' ПОЛИВ: летом - умеренный, зимой очень редко водой комнатной температуры или на несколько градусов теплее.'

' ПОДКОРМКА: достаточно 1 раз в месяц весной и летом слабым раствором минерального удобрения для кактусов и суккулентов.',

),

];

final List<Map<String, dynamic>> \_plantInform = [

{

'image': '1.png',

'title': 'Амариллис',

'desc': 'ОСВЕЩЕНИЕ: светолюбивое. Подойдут окна юго-западной или юго-восточной ориентации.'

' ВЛАЖНОСТЬ: не требовательное растение.'

' ТЕМПЕРАТУРА: летом 18-23°C. В период покоя 10-13°C.'

' ПОЛИВ: постоянно увеличивается по мере роста и развития. ВОДА НЕ ДОЛЖНА ПОПАДАТЬ НА ЛУКОВИЦУ',

},

{

'image': '2.jpg',

'title': 'Мирт',

'desc': 'ОСВЕЩЕНИЕ: светолюбивое растение.'

' ВЛАЖНОСТЬ: любит регулярное опрыскивание тёплой водой.'

' ТЕМПЕРАТУРА: летом - температура окружающей среды. Зимой не выше 10°C.'

' ПОЛИВ: поливать мягкой водой, поддерживая влажность почвы.'

' УДОБРЕНИЕ: с марта по август удобряют раз в месяц комплексным минеральным удобрением.'

' ОБРЕЗКА: должна быть регулярной и планомерной.',

},

{

'image': '3.jpg',

'title': 'Орхидея-паук (Брассия)',

'desc': 'ОСВЕЩЕНИЕ: яркий рассеянный свет с западной или восточной стороны.'

' ВЛАЖНОСТЬ: чем выше влажность, тем лучше самочувствие орхидеи (помодет увлажнитель воздуха и частое опрыскивание).'

' ТЕМПЕРАТУРА: идеальная температура 23-25°C, так же разрешена комнатная.'

' ПОЛИВ: метод погружения горшка в сосуд с водой на 20-30 минут (НЕЛЬЗЯ ДОПУСКАТЬ СМОРЩИВАНИЯ ПСЕВДОБУЛЬБ)',

},

{

'image': '4.jpg',

'title': 'Бальзамин',

'desc': 'ОСВЕЩЕНИЕ: неприхотлив, но для постоянно и обильного цветения нужен яркий свет хотя бы несколько часов.'

' ВЛАЖНОСТЬ: влаголюбивое растение, летом полив обильный (НЕ ДОПУСКАТЬ ПЕРЕСЫХАНИЯ ЗЕМЛЯНОГО КОМА).'

' ТЕМПЕРАТУРА: обычная комнатная температура (зимой не ниже 12-14°C).'

' ПОЛИВ: поливать мягкой водой, поддерживая влажность почвы.'

' УДОБРЕНИЕ: подкармливать с весны до осени каждые 2 недели минеральным удобрением для цветущих растений.',

},

{

'image': '5.jpg',

'title': 'Сенполия (фиалка)',

'desc': 'ОСВЕЩЕНИЕ: много света, но никаких солнечных лучей (восточные или западные окна).'

' ВЛАЖНОСТЬ: любит повышенную влажность (ОПРЫСКИВАТЬ ЛИСТЬЯ НЕЛЬЗЯ).'

' ТЕМПЕРАТУРА: для взрослой фиалки 20-24°C, для молодых растений 23-25°C .'

' ПОЛИВ: отстоянная в открытой посуде водопроводная вода (вода не должна попадать на листья и в центр розетки).'

' УДОБРЕНИЕ: примерно через 1.5 месяца после пересадки.'

' ОБРЕЗКА: должна быть регулярной и планомерной.',

},

{

'image': '6.png',

'title': 'Пеларгония (Герань)',

'desc': 'ОСВЕЩЕНИЕ: для цветения нужен солнечный свет.'

' ВЛАЖНОСТЬ: желательно опрыскивать утром или вечером!'

' ТЕМПЕРАТУРА: выдерживает кратковременное похолодание до +1°C.'

' ПОЛИВ: летом - обильный, зимой лучше поливать умеренно.'

' ПОДКОРМКА: раз в 2 недели жидким минеральным удобрением.'

' ОБРЕЗКА: ранней весной до начала вегетационного периода.',

},

{

'image': '7.png',

'title': 'Роза комнатная',

'desc': 'ОСВЕЩЕНИЕ: светолюбивое растение (южная, юго-восточная, юго-западная стороны).'

' ВЛАЖНОСТЬ: влажный воздух и теплый душ раз в неделю.'

' ТЕМПЕРАТУРА: летом - умеренная, зимой прохладная.'

' ПОЛИВ: зависит от степени цветения - постепенное увеличение, осенью полив прекращается.'

' ПОДКОРМКА: раз в неделю во время роста и цветения удобрением для растущих растений.'

' ОБРЕЗКА: обрезка побегов осенью, примерно на 1/3 оставляя на каждой ветке по 5 живых почек.',

},

{

'image': '8.jpg',

'title': 'Лизиантус (Эустома)',

'desc': 'ОСВЕЩЕНИЕ: светолюбивая, подходят ярко освещенные помещения.'

' ВЛАЖНОСТЬ: умеренная влажность, хорошо растет на открытом воздухе.'

' ТЕМПЕРАТУРА:в летний период 20-25°C, зимой 12-15°C.'

' ПОЛИВ: по мере подсыхания врехнего слоя, излишки воды из поддона слить, не допускать переувлажнение почвы.'

' ПОДКОРМКА: в период активного роста и цветения подкармливают комплексными минеральными удобрениями для цветущих растений 2 раза в месяц.'

' ОБРЕЗКА: отцветшие стебли срезают, оставляя не менее 2х пар листьев..',

},

{

'image': '9.jpg',

'title': 'Очиток (Седум)',

'desc': 'ОСВЕЩЕНИЕ: светолюбивое растение, хорошо растет на западном и южных окнах.'

' ВЛАЖНОСТЬ: не имеет значения, опрыскивание полезно для удаления пыли на листьях.'

' ТЕМПЕРАТУРА: не имеет существенного значения, выдерживают до 5°C, зимой больше подходит прохладное место с температурой 8-12°C.'

' ПОЛИВ: поливать умеренно, увлажнять по мере подсыхания почвы, пересушивание не вредит.'

' ПОДКОРМКА: весной и летом в период активного роста, достаточно 1 раза в месяц, используют удобрение для кактусов и суккулентов.',

},

{

'image': '10.png',

'title': 'Алоэ',

'desc':

'ОСВЕЩЕНИЕ: светолюбивое, предпочитает южные окна, но не пропадает и в тени -в темных комнатах алоэ вытягивается, становится бледным.'

' ВЛАЖНОСТЬ: существенной роли не играет, полезно иногда устраивать теплый душ для очистки листьев.'

' ТЕМПЕРАТУРА: летом температура должна быть умеренной, зимой прохладной, выдерживает температуру до 6°C.'

' ПОЛИВ: летом - умеренный, зимой очень редко водой комнатной температуры или на несколько градусов теплее.'

' ПОДКОРМКА: достаточно 1 раз в месяц весной и летом слабым раствором минерального удобрения для кактусов и суккулентов.',

},

];

List<Map<String, dynamic>> \_foundPlant = [];

TextEditingController editingController = TextEditingController();

@override

initState() {

\_foundPlant = \_plantInform;

super.initState();

}

void \_runFilter(String enteredKeyword) {

List<Map<String, dynamic>> results = [];

if (enteredKeyword.isEmpty) {

results = \_plantInform;

} else {

results = \_plantInform

.where((user) => user['title']

.toLowerCase()

.contains(enteredKeyword.toLowerCase()))

.toList();

}

setState(() {

\_foundPlant = results;

});

}

@override

Widget build(BuildContext context) {

return Container(

margin: EdgeInsets.only(top: 18, left: 14, right: 14),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

Text(

'Это я - твой помощник',

style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),

),

Icon(

IconData(0xe0ef, fontFamily: 'MaterialIcons'),

size: 40,

color: Color.fromARGB(255, 31, 106, 85),

)

],

),

Text(

'по уходу за цветами и растениями',

style: TextStyle(

fontSize: 25,

fontWeight: FontWeight.bold,

color: Color.fromARGB(255, 20, 109, 91)),

),

Padding(

padding: EdgeInsets.symmetric(vertical: 35),

child: Row(

children: [\_searchBox()],

),

),

SizedBox(

height: 24,

),

Expanded(

child: StaggeredGridView.countBuilder(

physics: BouncingScrollPhysics(),

crossAxisCount: 4,

itemCount: \_foundPlant.length,

itemBuilder: (BuildContext context, int index) {

return GestureDetector(

onTap: () {

FocusScope.of(context).requestFocus(new FocusNode());

Navigator.of(context).push(

MaterialPageRoute(

builder: (context) => DetailPage(

product: \_products[index],

),

),

);

},

child: Column(

children: [

Stack(

children: [

Stack(

children: [

Center(

child: Image.asset(

'assets/images/${\_foundPlant[index]['image']}',

width: 200,

fit: BoxFit.cover,

),

),

],

),

],

),

SizedBox(

height: 8,

),

Text(

'${\_foundPlant[index]['title']}',

style: TextStyle(fontWeight: FontWeight.bold),

),

],

));

},

staggeredTileBuilder: (int index) => new StaggeredTile.fit(2),

mainAxisSpacing: 24,

crossAxisSpacing: 24,

),

)

],

),

);

}

Widget \_searchBox() {

return Expanded(

child: TextField(

onChanged: (value) {

\_runFilter(value);

},

controller: editingController,

decoration: InputDecoration(

hintText: 'Поиск',

prefixIcon: Icon(

Icons.search,

size: 30,

),

filled: true,

fillColor: Color.fromARGB(255, 169, 225, 200),

contentPadding: EdgeInsets.all(15),

enabledBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.grey.shade200),

borderRadius: BorderRadius.circular(15),

),

focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.grey.shade200),

borderRadius: BorderRadius.circular(15),

),

),

),

);

}

}

class Product {

String title;

String image;

String desc;

Product({this.title, this.image, this.desc});

}

**Содержимое файла: detail.dart**

import 'package:flutter/material.dart';

import 'home.dart';

class DetailPage extends StatelessWidget {

final Product product;

DetailPage({@required this.product});

@override

Widget build(BuildContext context) {

return Scaffold(

body: SafeArea(

child: Column(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 18, horizontal: 14),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

GestureDetector(

onTap: () => Navigator.pop(context),

child: Icon(

Icons.arrow\_back,

size: 30,

),

),

],

),

),

\_productImage(),

SizedBox(

height: 16,

),

Expanded(

child: Container(

padding: EdgeInsets.only(left: 18, top: 28),

decoration: BoxDecoration(

borderRadius: BorderRadius.only(

topLeft: Radius.circular(30),

topRight: Radius.circular(30),

),

color: Color.fromARGB(133, 246, 178, 198), //блок описания цветка

),

child: \_productDescription()),

),

],

),

),

);

}

Widget \_productImage() {

return Stack(

children: [

Positioned.fill(

child: Align(

alignment: Alignment.bottomCenter,

child: Container(

height: 50,

width: 300,

),

),

),

Center(

child: Image.asset(

'assets/images/${product.image}',

height: 350,

fit: BoxFit.cover,

),

),

],

);

}

Widget \_productDescription() {

return ListView(

physics: BouncingScrollPhysics(),

children: [

SizedBox(height: 24),

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

Text(

'${product.title}',

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 30),

),

Container(

padding: EdgeInsets.all(14),

decoration: BoxDecoration(

borderRadius: BorderRadius.only(

topLeft: Radius.circular(25),

bottomLeft: Radius.circular(25),

),

color: Color.fromARGB(255, 77, 16, 37)),

)

]

),

SizedBox(

height: 32,

),

Text(

'Советы по уходу:',

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 20),

),

SizedBox(

height: 14,

),

Padding(

padding: const EdgeInsets.only(right: 14),

child: Text(

'${product.desc}',

),

),

SizedBox(

height: 14,

),

SizedBox(

height: 48,

)

]

);

}

}