

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»
(Университет ИТМО)**

Факультет Прикладной информатики

Направление подготовки 45.03.04 Интеллектуальные системы в гуманитарной сфере

Образовательная программа Языковые модели и искусственный интеллект

КУРСОВОЙ ПРОЕКТ

Тема: «iOS приложение для поиска картинок (Pinterest)»

Обучающийся: Скоринцева Таисия Александровна К3160

Санкт-Петербург 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Структура проекта и этапы работы над ним.....	7
1.1 Описание проекта.....	7
1.2 Описание этапов разработки.....	10
2 Результат проделанной работы и её анализ.....	13
2.1 Описание поставленной передо мной задачи.....	13
2.2 Решение поставленной задачи.....	14
2.3 Анализ проделанной работы.....	22
3 Описание взаимодействия с командой и руководителем.....	24
3.1 Взаимодействие с командой.....	24
3.2 Взаимодействие с руководителем проекта.....	24
3.3 Оценка работы руководителя проекта.....	25
ЗАКЛЮЧЕНИЕ.....	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	28
ПРИЛОЖЕНИЕ.....	29

ВВЕДЕНИЕ

Проект «iOS приложение для поиска картинок (Pinterest)», которое предстояло реализовать нашей команде, может быть актуален как для студентов, решивших попробовать себя в разработке, так и для потенциальных пользователей приложения в будущем. Данное приложение предоставляет пользователям возможность находить источники вдохновения через визуальные изображения, что особенно актуально в современном мире. Наша команда уверена, что умение работать с визуальным контентом, а также навыки программирования и проектирования, полученные в ходе работы над приложением, станут неоценимыми в будущем. Мы планируем применять эти знания для создания инновационных продуктов, которые удовлетворят потребности пользователей и помогут им выражать свою индивидуальность. Целевая аудитория проекта – молодежь, потому что именно нынешнее поколение черпает вдохновение через фотографии и картинки.

Для самих же пользователей наш проект может стать аналогом достаточно известного приложения Pinterest. Мы тщательно изучили вопрос актуальности подобного аналога и пришли к выводу, что наш проект смог бы полностью заменить Pinterest в случае его ухода с рынка.

Наша команда начала изучать новый язык программирования Swift, а именно его основы, с помощью которых в последующем были созданы экраны приложения и реализовано их взаимодействие друг с другом.

Значительное внимание было уделено эстетике приложения: концепции экранов, цветовой палитре, удобным и понятным навигационным иконкам. Таким образом, разработанные экраны гармонируют друг с другом, обращая внимание пользователя на удобный, стильный и понятный интерфейс.

Популярность визуального контента для поиска вдохновения в современном мире также указывает на актуальность темы данного проекта. С помощью данного приложения люди по всему миру могут обмениваться своим образом жизни, полезными привычками, любимыми местами для

путешествий, а также делиться советами, сокровенными эмоциями или просто идеями в самых разных сферах жизни человека. В настоящее время молодежь нуждается именно в таком способе вдохновения, поскольку каждый стремится развить в себе индивидуальность, хочет попробовать себя в чём-то новом, неизведанном, незнакомом своему образу жизни.

Интерес к тому, как функционируют приложения, привел нашу команду к изучению и познанию мобильной разработки, поскольку это универсальный инструмент для создания приложений, доступный каждому желающему благодаря обилию литературы и других ресурсов в этой сфере. Полное погружение в процессы мобильной разработки, от описания основополагающих элементов до реализации их взаимодействия, позволили членам нашей команды на практике продемонстрировать изученный материал и поработать с первым значимым для портфолио проектом.

Изучение выбранной темы курсового проекта имеет важное значение, так как подразумевает практический подход. Погружаясь в специализированную литературу и просматривая актуальные видеоматериалы, а также применяя приобретенные знания на практике, мы начинаем осознавать значение каждого элемента взаимодействия, узнаем, какие библиотеки, методы, функции помогут реализовать задуманные идеи и в чем заключается их актуальность для нашего проекта. Таким образом, формируется понимание внутреннего устройства приложения.

Цель проекта — за отведенный срок создать приложение на языке программирования Swift с помощью среды разработки Xcode и реализовать работу всех функций, обозначенных руководителем проекта.

Чтобы достичь поставленной цели, мне и моей команде было необходимо выполнить следующие задачи:

- 1) ознакомиться с основами языка программирования Swift;
- 2) изучить набор готовых компонентов UIKit;

- 3) создать набор цветов приложения;
- 4) создать набор шрифтов приложения;
- 5) создать набор иконок приложения;
- 6) создать дизайн экрана рекомендуемых картинок;
- 7) создать дизайн экрана детального просмотра картинки;
- 8) создать дизайн экрана поиска;
- 9) добавить шрифты в приложение;
- 10) добавить цвета в приложение;
- 11) добавить иконки в приложение;
- 12) разработать экран рекомендуемых картинок;
- 13) разработать экран детального просмотра картинки;
- 14) разработать экран поиска;
- 15) добавить API на экран рекомендованных картинок;
- 16) добавить API на экран поиска;
- 17) реализовать бизнес-логику на экране рекомендуемых картинок;
- 18) реализовать бизнес-логику на экране детального просмотра картинки;
- 19) реализовать бизнес-логику на экране поиска;
- 20) исправить обнаруженные в процессе тестирования приложения ошибки и реализовать дополнительные функции;

Всего в нашей команде было два дизайнера и три разработчика. Я выбрала роль IOS разработчика, поскольку мне было интересно поработать именно с технической частью создания приложения. Руководитель проекта распределил обязанности между всеми членами нашей команды, определил ответственных за определенные блоки задач и мы приступили к работе.

1 Структура проекта и этапы работы над ним

1.1 Описание проекта

Проект «iOS приложение для поиска картинок (Pinterest)» представляет собой прототип iOS-приложения, созданного для удобного поиска и работы с визуальным контентом. Сегодня, когда визуальная культура занимает важное место в повседневной жизни, многие пользователи ищут вдохновение для своих идей — от оформления интерьеров до создания блюд и выбора подарков. Это приложение предоставляет возможность находить изображения по ключевым словам, изучать персонализированные рекомендации, а также сохранять понравившиеся изображения для просмотра в оффлайн-режиме.

Программа ориентирована на молодую аудиторию, включая творческих людей, которым важно получать эстетическое удовольствие от просмотра фотографий, а также использовать их для презентаций или вдохновения. Итак, в приложении был реализован следующий функционал:

- 1) лента изображений, рекомендованных на основе предпочтений пользователя;
- 2) возможность детального просмотра и увеличения изображений;
- 3) функция сохранения картинок в галерею устройства;
- 4) поиск по ключевым словам с переходом к тематическим рекомендациям;
- 5) удобная навигация между экранами с результатами поиска и предложенными изображениями;

Для создания приложения использовалась архитектура VIPER. Этот паттерн основан на разделении ответственности, где один компонент отвечает за ограниченное количество процессов внутри модуля, что облегчает процесс написания кода и позволяет изменять его части вне зависимости от других. В

рамках данной архитектуры каждый модуль разделен на пять основных компонентов:

View отвечает за отображение пользовательского интерфейса и передачу данных от Presenter.

Interactor реализует бизнес-логику приложения, обрабатывает запросы от Presenter и обеспечивает взаимодействие с серверной частью.

Presenter управляет потоком данных между View и Interactor, реагирует на действия пользователя (например, нажатия), форматирует информацию для отображения и обеспечивает ее корректное представление на экране.

Entity представляет собой модель данных, описывающую их структуру и свойства.

Router отвечает за навигацию внутри приложения и управление переходами между экранами.

Для разработки приложения под операционную систему iOS использовался язык программирования Swift, созданный специально для данной платформы и известный своим лаконичным синтаксисом и защитой от типичных ошибок. Выбранный нами язык также обеспечивает управление памятью, что особенно важно для мобильных устройств в условиях ограниченности ресурсов.

Для создания графического интерфейса использовался фреймворк UIKit, предоставляющий программисту широкий выбор элементов для построения пользовательского интерфейса: кнопки, таблицы, коллекции, изображения и другие. Основным элементом интерфейса в данном проекте стал UICollectionView, благодаря которому удалось воссоздать стиль отображения, аналогичный Pinterest. Кроме того, UIKit поддерживает обработку пользовательских действий (таких как свайпы и нажатия) и обеспечивает плавные анимации, благодаря которым взаимодействие с приложением становится приятнее.

Для упрощения работы с Auto Layout использовалась библиотека SnapKit, которая позволяет задавать ограничения для элементов интерфейса всего за несколько строк кода. Это не только ускоряет процесс разработки, но и делает код более читаемым и поддерживаемым. Благодаря SnapKit интерфейс приложения адаптируется под разные размеры экранов и ориентацию устройства, что улучшает опыт пользователя.

Сетевые запросы в приложении реализованы с помощью библиотеки Alamofire, которая распространяется под лицензией MIT. Она позволяет эффективно обрабатывать запросы к серверу, загружать изображения и метаданные, а также отображать их в пользовательском интерфейсе. Использование этой библиотеки обеспечило быструю интеграцию API и стабильность приложения. Выбранная технология играет ключевую роль в предоставлении пользователю визуального контента, который загружается и отображается на экране.

1.2 Описание этапов разработки

При работе над проектом наша команда следовала четкому плану, а совместно с руководителем были определены основные этапы работы над проектом.

На первом этапе мы организовали знакомство внутри команды, установили правила общения и выполнения поставленных задач, далее обсудили техническое задание, каждый рассказал о своем опыте работы в подобных проектах, а руководитель проекта рассказал о задачах и требуемых от нас промежуточных и итоговых результатах. На платформе Odoo руководителем были размещены блоки задач с дедлайнами. Нашей главной задачей на данном этапе было ознакомление с платформой и поставленными задачами. Также на данном этапе команда разработчиков синхронизировала локальные репозитории с удаленным на GitHub, чтобы руководитель в дальнейшем отслеживал процесс работы над проектом и мог вносить правки, касательно возникающих ошибок.

На втором этапе наша команда приступила к выполнению первого блока задач, установленных руководителем. Команда дизайнеров разработала основу интерфейса приложения в Figma: брендбук и иконки. Брендбук включал в себя набор шрифтов с определенным стилем и размерами, классическую цветовую палитру с названиями каждого оттенка, иконки же включали в себя классические изображения, подобранные специально для обеспечения удобной навигации внутри приложения. Созданные элементы дизайна, а именно готовые макеты экранов и брендбук, были сохранены на общей доске в приложении Figma.

На третьем этапе команда разработчиков приступила к непосредственному созданию и настройке экранов по готовым макетам от дизайнеров, которые к тому моменту уже были одобрены руководителем. Команда разработчиков совместно с руководителем организовала первую онлайн-встречу, где мы подробно разбирали функционал отдельных базовых

элементов Swift. Мы выслушали рекомендации и советы от магистранта по работе с данными элементами, сохранили полезные источники, нашли онлайн-пособие для начинающих и приступили к интеграции шрифтов, цветов и иконок, созданию своих экранов, ориентируясь на готовые макеты дизайнеров и параллельно изучая дополнительную литературу. Главным результатом нашей работы на данном этапе стали реализованные три модуля: рекомендованных картинок, детального просмотра картинки и поиска. Кроме того, на данном этапе команда разработчиков реализовала обработку событий для того, чтобы контролировать корректность работы реализованных экранов при запуске приложения.

На четвертом этапе нам предстояло уделить время всем значительным и незначительным ошибкам, которые не устраивали руководителя при взаимодействии с приложением. Так для каждого экрана был прописан ряд дополнительных задач по совершенствованию работы приложения, такие как возможность скрыть иконки при увеличении изображения, обработка сетевых запросов и повторная настройка бизнес логики с исправлением ошибок.

Подводя итоги работы на каждом этапе, можно сделать вывод о том, что внутри нашей команды было понимание и вовлеченность со стороны каждого участника, поддержка со стороны руководителя в любое время дня и ночи. Если возникали какие-либо трудности у команды разработчиков, то организовывались дополнительные онлайн-консультации от руководителя. Кроме того, мы в любой момент могли попросить помощи или спросить совета у сокомандников, поскольку по мере работы над проектом были найдены общие точки соприкосновения и все были заинтересованы в конечном результате.

2 Результат проделанной работы и её анализ

2.1 Описание поставленной передо мной задачи

В начале работы над проектом передо мной были поставлены следующие задачи:

- 1) осуществить первоначальную настройку компьютера;
- 2) добавить цвета из палитры, созданной дизайнерами;
- 3) создать модуль (экран) детального просмотра;
- 4) реализовать навигацию между модулем детального просмотра и другими экранами внутри приложения;
- 5) добавить функцию увеличения/приближения картинки при нажатии на нее в общем списке картинок на экране поиска или рекомендаций;
- 6) добавить функцию растягивания картинки поверх navBar и tabBar;
- 7) добавить возможность скрыть или показать кнопки “назад” и “сохранить” при нажатии на экран, и автоматически скрыть при изменении масштаба картинки;
- 8) добавить функцию сохранения в галерею устройства, нажав на кнопку “Сохранить” .

Основной моей целью стала разработка адаптивного и функционального экрана детального просмотра картинки. Для этого были разработаны следующие функции:

- 1) открытие экрана детального просмотра при нажатии на картинку;
- 2) скрытие стандартного навигационного меню и разработка отдельного для экрана детального просмотра;
- 3) сохранение картинки пользователем;
- 4) увеличение или уменьшение картинки;
- 5) изменение цвета кнопок при их пересечении с изображением для повышения контрастности;

В бизнес-логике я прописала алгоритмы взаимодействия с картинкой после нажатия на нее в общем списке.

Отдельной задачей было добавление палитры цветов брендбука, который разработали дизайнеры. Для этого я добавила названия оттенков и их HEX-код. В качестве дополнительной задачи руководитель определил следующее задание: кнопки должны менять цвет, когда изображение оказывается под ними и кнопки должны анимировано открываться после нажатия и исчезать при изменении масштаба картинки.

2.2 Решение поставленной задачи

Перед началом работы с выбранными библиотеками и языком программирования Swift, я решила изучить основы синтаксиса данного языка программирования и ознакомиться с его базовыми компонентами с помощью книги Василия Усова «Основы разработки приложений под iOS, iPadOS и macOS», представленной на рисунке 1 [1].

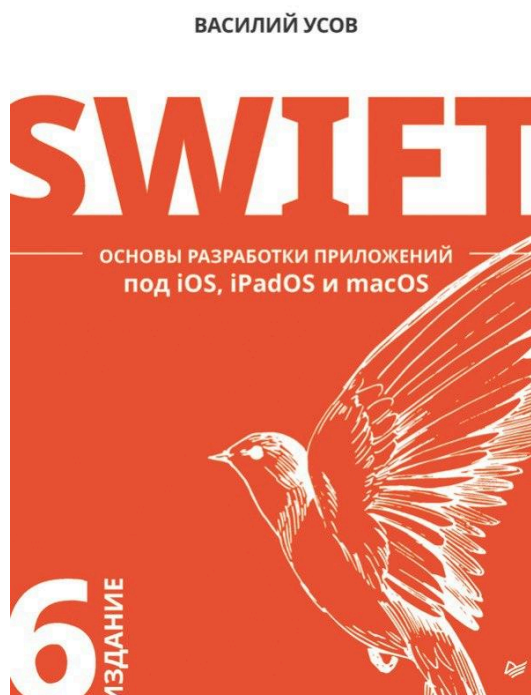


Рисунок 1 – книга Василия Усова «Основы разработки приложений под iOS, iPadOS и macOS».

Для реализации экрана детального просмотра изображений я просмотрела несколько лекций в видеоформате, а также изучила документацию библиотеки UIKit, поскольку это самый достоверный источник информации о данной технологии. Моя работа над проектом также состояла из нескольких этапов. Для разработки мне необходимо было использовать коллекцию (компонент UICollectionView библиотеки UIKit) по многим техническим причинам.

Первым этапом моей работы было интегрирование цветовой палитры в приложение с помощью поиска оттенков через HEX-код. Если говорить более детально, то для добавления цветов из брендбука мне нужно было создать отдельную палитру в папке “Colors” в Xcode, где каждый цвет имеет название из брендбука и индивидуальный код. Результат выполнения – цвета интегрированы в приложение и могут использоваться в любом месте.

На втором этапе работы мне нужно было реализовать навигацию между модулем детального просмотра картинки и другими экранами в приложении. Я выполнила данную задачу с помощью описания функции нажатия на картинку из общего списка. Так считывалось нажатие на картинку `didTap()`, после чего открывался экран детального просмотра картинки [2].

На третьем этапе работы мне нужно было реализовать функцию уменьшения/увеличения картинки с помощью стандартных жестов. Для этого был создан экземпляр `UIImageView`, который отображает изображение, затем была добавлена инициализация и распознавание жестов увеличения/уменьшения картинки. А затем уже была реализована обработка жеста и непосредственно изменение масштаба картинки. На этом же этапе мне было необходимо реализовать функцию сохранения картинки в галерею устройства после нажатия на кнопку “Сохранить”. Для этого я реализовала метод `didTapSave` и добавила необходимые права на запись изображения в

галерею. На данном этапе сложнее всего было реализовать сохранение изображения, поскольку возникла проблема с форматом изображения, в котором оно может быть сохранено, но благодаря советам руководителя и дополнительным правкам, мне удалось устранить проблему и реализовать сохранение картинки.

Так по окончании трех этапов мною был реализован экран детального просмотра, что показано на рисунке 2.



Рисунок 2 – Модуль детального просмотра картинок

А также была реализована возможность менять масштаб картинки, что показано на рисунке 3.

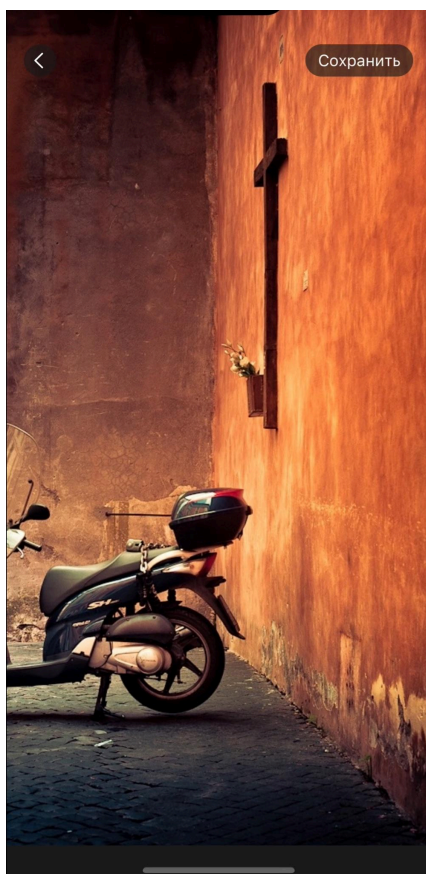


Рисунок 3 – Модуль детального просмотра картинок с возможностью изменения масштаба картинки

На четвертом этапе были выполнены дополнительные задачи от руководителя. Поскольку картинка на экране детального просмотра имеет разрешение лучше, чем в общем списке, то для того, чтобы пользователь мог растянуть картинку на весь экран, было скрыто стандартное навигационное меню и было реализовано собственное. Также кнопки “назад” и “сохранить” можно скрыть или показать, нажав на экран, и они автоматически скрываются при изменении масштаба картинки. Для повышения читабельности кнопок было реализовано изменение их фона на черный, если картинка при увеличении залезла под кнопки. Если пользователь уменьшил картинку, и она вылезла из под кнопок, то у них фон перекрашиваются назад в белый. Это нужно для увеличения читабельности текста кнопок, потому что белая кнопка на белом фоне и на светлой картинке теряется. По окончании

данного этапа мною были реализованы дополнительные функции. Результат работы продемонстрирован на рисунке 4.

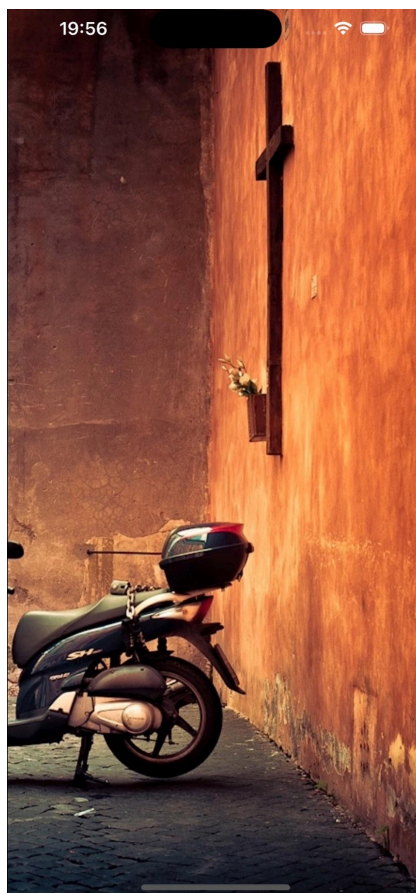


Рисунок 4 – Модуль детального просмотра, дополнительные функции исчезновения стандартного навигационного меню

В процессе и по окончании каждого этапа разработки я синхронизировала свой локальный репозиторий с удаленным репозиторием на сервисе GitHub, где руководитель проверял и одобрял работу.

2.3 Анализ проделанной работы

Подводя итоги проделанной работы, хочется отметить, что поставленные задачи оказались достаточно сложными, поскольку без помощи руководителя некоторые возникающие ошибки было невозможно решить. Я считаю, что со своей работой я справилась, поскольку у меня получилось не только за короткий срок ознакомиться с совершенно новым языком программирования, но и на практике успешно применить полученные знания. Так мной был реализован целый модуль, включающий в себя множество функций для осуществления взаимодействия пользователя с картинкой, что полностью соответствует заявленному техническому заданию. Эти задачи требовали полного погружения в тему.

Итак, мной были выполнены следующие задачи:

- добавлены цвета в приложение;
- реализован модуль детального просмотра с продуманной бизнес-логикой;
- добавлены дополнительные функции: растягивание картинки поверх основного навигационного меню, изменение цвета кнопок в зависимости от цвета фона, на котором они появляются.

Что вызвало трудности:

- нехватка опыта в программировании;
- разработка приложения конкретно под IOS и привязанность к отдельной, новой для меня программы;
- изучение абсолютно нового языка программирования;
- отсутствие опыта в мобильной разработке;
- выполнение сложных для первокурсника задач в сжатые сроки;
- отсутствие должного времени для глубоко изучения необходимых тем.

Чему я научилась:

- изучила новый язык программирования Swift;

- научилась работать под контролем наставника, взаимодействуя с командой, где у каждого свои отдельные задачи, которые в будущем становятся одним целым;
- научилась самостоятельно искать необходимую информацию для определения ошибок и их исправления;
- повысила свой уровень владения GitHub;
- получила навыки работы на GitHub в команде разработчиков;
- освоила новые фреймворки и библиотеки;
- научилась программно реализовывать требования команды дизайнеров;
- научилась взаимодействовать с формальными документациями языков программирования и различных библиотек на иностранном языке.

Я считаю, что мне удалось справиться с поставленной задачей именно благодаря правильному и рациональному распределению поставленных задач. Кроме того, я всегда была на связи с руководителем и командой разработчиков, благодаря чему все возникающие проблемы решались в кратчайшие сроки и под наблюдением наставника.

Работа над проектом с технической точки зрения привнесла в мою жизнь много полезных навыков и методов эффективного планирования.

3 Описание взаимодействия с командой и руководителем

3.1 Взаимодействие с командой

Взаимодействие с участниками команды осуществлялось через телеграм-чат, созданный нашим руководителем. В нем мы согласовывали все изменения, нововведения и прочие правки в наш проект. В чате каждый делился материалами для изучения новых для нас технологий, библиотек и фреймворков.

Также для живого общения регулярно устраивались созвоны, на которых совместно с руководителем обсуждали все рабочие моменты. Для созвонов мы использовали сервис компании Google под названием Google Meet.

Для проекта был создан совместный удаленный репозиторий на сервисе GitHub, куда каждый из команды разработчиков мог вносить свои изменения. Именование этих изменений были строго регламентированы руководителем и каждый из команды следовал этим правилам.

Благодаря всему вышеперечисленному мы поддерживали внутри команды рабочую и дружественную обстановку, за счет чего всем участникам было комфортно работать в проекте.

3.2 Взаимодействие с руководителем проекта

Коммуникация с руководителем также происходила с использованием мессенджера телеграм. Александр всегда был на связи как в общем чате, так и в личном. Наставник помогал с освоением новых технологий, делился материалами для получения знаний и навыков.

В начале работы над проектом руководитель объявил правила, которым мы должны следовать при совместной разработке. Все требования были сформулированы исходя из большого коммерческого опыта наставника. За счет этого каждый из команды усвоил различные хорошие практики, применяющиеся в промышленной разработке.

Также Александр внес огромный вклад в защиту нашего проекта. Он помогал нам составить речь для презентации, редактировал доклад и указывал на все моменты требующие доработки.

3.3 Оценка работы руководителя проекта

Я ставлю отличную оценку работе руководителя. Наше сотрудничество было приятным и познавательным. Наставник передал нам очень много знаний и опыта в сфере промышленной разработки. Несмотря на то, что мы часто сталкивались с трудностями и сомнениями при освоении нового, он всегда был рядом, поддерживал нас и объяснял все сложные вопросы доступным языком.

ЗАКЛЮЧЕНИЕ

Проект «iOS приложение для поиска картинок (Pinterest)» помог реализовать прототип мобильного приложения, для просмотра ленты рекомендованных изображений, а также искать картинки и детально их рассматривать.

Цель проекта — за полтора месяца (с 01.11.2024 по 17.12.2024) разработать прототип приложения для семейства операционных систем iOS с возможностями поиска, детального просмотра и сохранения изображений, а также получения персональных подборок через API Flickr — была достигнута. Команда разработчиков справилась со всеми поставленными задачами, хотя во время разработки каждый столкнулся с определенными трудностями, которые смог преодолеть. Тем не менее, помимо основных целей, нам удалось выполнить дополнительные задачи, которые значительно улучшили функциональность и уникальность приложения. В итоге в приложении появились модули рекомендаций, поиска и детального просмотра изображений, а также функции обновления экранов рекомендаций, поиска и сохранения картинок в галерею устройства.

К сожалению, из-за ограниченного числа iOS-разработчиков (вместо запланированных пяти) и нехватки времени не удалось реализовать модули профиля и авторизации. Кроме того, во время работы было решено заменить API Pinterest на API Flickr и RxAlamofire на Alamofire из-за проблем с установкой библиотеки у одного из участников. Тем не менее, эти изменения не снизили качество проекта, поскольку основная цель была достигнута.

Мой личный вклад в проект заключался в разработке модуля рекомендаций. Я отвечала за интеграцию цветов, реализацию модуля детального просмотра, внедрение бизнес-логики. Помимо этого, я выполнила дополнительные задачи: растягивание картинки поверх основного

навигационного меню, изменение цвета кнопок в зависимости от цвета фона, на котором они появляются.

Этот проект стал для меня возможностью изучить новый язык программирования и освоить новую среду разработки, а также познакомиться с областью мобильной разработки, и поработать в команде, в окружении мотивированных на результат людей. Я считаю этот проект ценным опытом для своего портфолио.

Наш проект показал, что мобильная разработка доступна каждому, даже несмотря на все трудности и нехватку опыта в программировании. Прототип обладает значительным потенциалом, и с добавлением некоторых дополнительных функций, его можно превратить в полноценное мобильное приложение. В будущем я бы хотела продолжить работу над проектом, добавив модули профиля и авторизации, а также добавить возможность сохранять картинки в отдельные подборки внутри самого приложения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Усов В. Swift. Основы разработки приложений под iOS, iPadOS и macOS. 6-е изд., доп. и перераб. — СПб.: Питер, 2021. — 544 с.: ил. — (Серия «Библиотека программиста»). — ISBN 978-5-4461-1796-3 (дата обращения: с 8.11.2024 по 13.12.24)
2. Официальная документация языка Swift
<https://www.swift.org/documentation/>
3. Swift Apprentice: Fundamentals (First Edition): Beginning Programming in Swift. — ISBN-13 : 978-1950325825
4. Универсальный ресурс для программистов с ответами на различные вопросы <https://stackoverflow.com/>
5. Российский ресурс с различными статьями от опытных разработчиков IT-компаний <https://habr.com/ru/feed/>

ПРИЛОЖЕНИЕ

Техническое задание для iOS приложения для поиска картинок (Pinterest)

Описание проекта

Модули МП

Описание модулей МП

Модуль авторизации/регистрации

Модуль рекомендуемых картинок

Модуль поиска

Модуль детального просмотра картинки с возможностью сохранить её на устройство

Модуль профиля

Способы навигации

Возможные улучшения приложения

Добавить работу с пользовательскими бордами и пинами:

Добавить в модуль рекомендуемых картинок при открытии с поиска уточнения к поиску.

Модуль входа по пинкоду и биометрии

Основное назначение проекта: iOS приложение, в котором есть возможность искать, детально просматривать и сохранять картинки, смотреть персональные подборки картинок, получая их с API Pinterest.

Технические характеристики: Swift 5, UIKit, SnapKit, RxAlomafire, PinterestSDK, Kingfisher, FloatingPanel.

API: <https://developers.pinterest.com/docs/api/v5/introduction>

Модули МП:

- 1) модуль авторизации/регистрации
- 2) модуль рекомендуемых картинок
- 3) модуль поиска

4) модуль детального просмотра картинки с возможностью сохранить её на устройство

5) модуль профиля

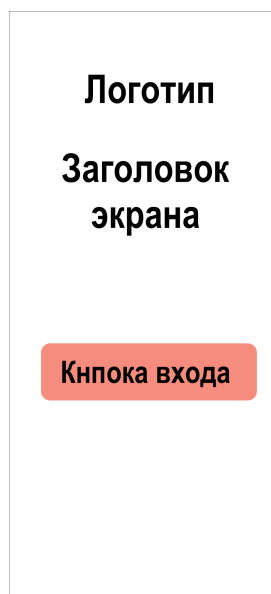
Навигация между экранами

Навигация между модулями рекомендуемых картинок, поиска и профиля осуществляется с помощью таб бара. Между остальными модулями с помощью нав бара.

Описание модулей МП

- 1) модуль авторизации/регистрации
- 2) экран авторизации/регистрации содержит логотип, заголовок экрана кнопку, которая перенаправляет на сайт/в мобильное приложение (если оно установлено) пинтереста для авторизации/регистрации и по диплинку возвращает пользователя назад.

Для реализации обязательна библиотека [PinterestSDK for iOS](#)



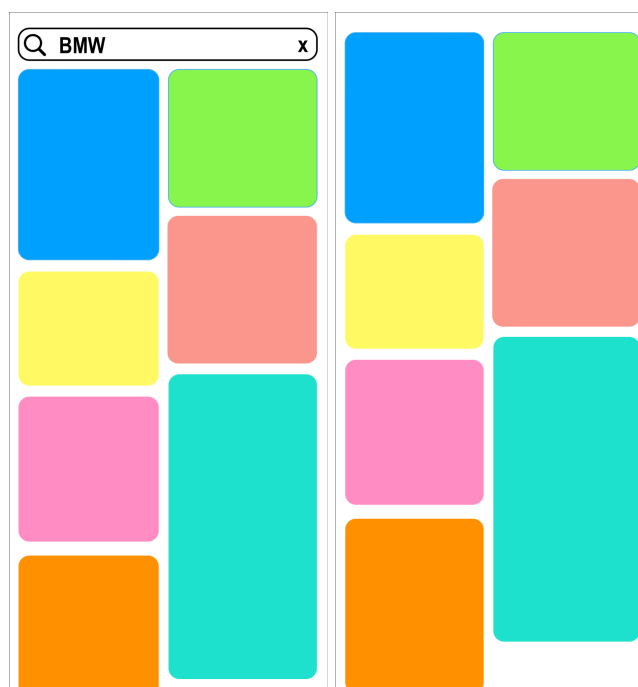
Модуль рекомендуемых картинок состоит из следующих секций:

Секция с названием бордов состоит только из текста с возможностью горизонтального скrolла. Для выбора борда можно использовать свайп влево или вправо и тап по самому названию борда. Если у пользователя нет бордов, то не отображается. (опционально)

Секция с SearchBar, доступна при открытии данного модуля из поиска.

Секция с картинками состоит из картинок, которые относятся к текущему борду, картинка показывается полностью с сохранением оригинальной пропорции высоты и ширины. В ряд помещается 2 картинки. При тапе на картинку открывается модуль детального просмотра картинки.

Для реализации модуля использовать UICollectionView, CompositionalLayout.



Модуль поиска состоит из пяти секций:

Секция поиска - состоит из SearchBar.

Секция рекомендаций - показывается, пока пользователь не нажал на SearchBar.

Секция истории поиска - состоит из иконки поиска, текста поиска и кнопки удалить. Показывается, когда пользователь нажал на SearchBar, но ничего не ввел. При тапе открывается модуль просмотра подходящих картинок под запрос.

Секция предложенных вариантов поиска - состоит из иконки поиска, текста возможного поиска. Показывается, когда пользователь начал что-то вводить в SearchBar. В случае если не удалось найти похожий запрос, показывается заглушка. При тапе открывается модуль просмотра подходящих картинок под запрос.

Для реализации модуля использовать UICollectionView, CompositionalLayout.

Модуль детального просмотра картинки с возможностью сохранить её на устройство

Модуль детального просмотра картинки с возможностью сохранить её на устройство состоит из картинки на весь экран, кнопки в нав баре, которая сохраняет картинку. При свайпе вниз модуль закрывается и возвращает пользователя на предыдущий экран, при свайпе влево или вправо, открывается предыдущая или следующая картинка в борде. Так же есть возможность увеличивать картинку.



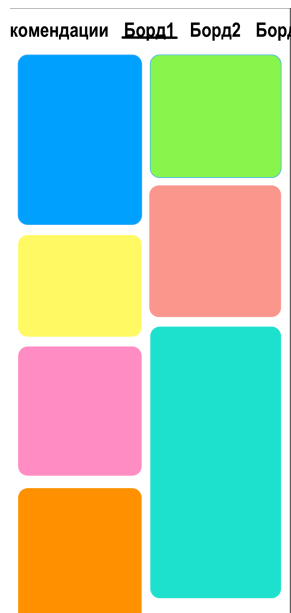
Модуль профиля

Модуль профиля состоит из аватарки пользователя, если её нет, то первая буква имени, имени, включения/выключения биометрии (опционально), кнопки “сменить пин код”, которая открывает экран смены пин кода (опционально), кнопки выхода.

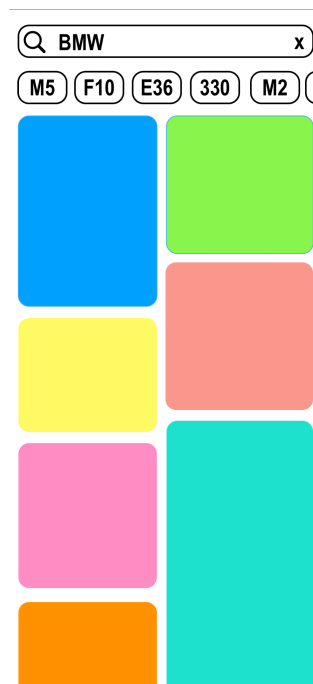
Возможные улучшения приложения

Добавить работу с пользовательскими бордами и пинами:

1. Возможность просматривать собственные пины и борды в модуле профиля
2. Добавлять картинки в пины или борды
3. Создавать собственные борды
4. Отображать на экране рекомендаций подборки картинок подходящие к пользовательским бордам



Добавить в модуль рекомендуемых картинок при открытии с поиска уточнения к поиску.



Модуль входа по пин коду и биометрии состоит из четырех экранов: экран входа, экран создания пароля, экран подтверждения пароля, экран смены пароля, и модального окна “Забыли пароль”. Все экраны данного модуля состоят из клавиатуры с цифрами от 0 до 9, с кнопкой, которая

удаляет последнюю цифру пин кода, точек, символизирующих введенные цифры.

Экран входа содержит текст “Введите пин код”, кнопку входа по биометрии и кнопку “Не можете войти?”, которая открывает модальное окно “Забыли пин код”, окно с кнопками “Войти по биометрии” и кнопкой “Войти заново”. При появлении экрана необходимо сразу проводить авторизацию по биометрии

Экран создания пароля содержит текст “Установите ПИН-код для входа”. После того, как пользователь ввел пин код, его автоматически перенаправляет на экран подтверждения пин кода.

Экран подтверждения пароля содержит текст “Повторите ПИН-код”. После успешного повторения пин кода открывается модальное окно “Использовать биометрию?”.

Экран смены пин кода аналогичен экрану создания пин кода, исключая надпись. Надпись на экране должна быть “Введите новый пин код”.

Поведение точек, символизирующих введенные цифры - на всех экранах они заполняются соответственно длине введенного пин кода. На экране входа и подтверждения в случае корректного пин кода они меняются на зеленый цвет, в случае некорректного окрашиваются в красный и трясутся.

Модальное окно “Забыли пин код” состоит из заголовка, подзаголовка и двух кнопок: войти по биометрии и заново авторизоваться. Если пользователь ранее не добавлял биометрию, то кнопка “войти по биометрии” отсутствует.

Модальное окно “Использовать биометрию?” состоит из заголовка, подзаголовка и кнопки “Использовать биометрию”.

Для реализации обязательна библиотека LocalAuthentication.

Логотип

Введите пин код



1	2	3
4	5	6
7	8	9
	0	

Не можете войти?