**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**](https://isu.ifmo.ru/pls/apex/f?p=2143:13:116963507483892::NO::EP_UCHEB_YEAR,EP_MEGAFACULTY,EP_FACULTY,EP_CATHEDRA,EP_GROUP,EP_FO,EP_KVAL,EP_UCHEB_PLAN:2024/2025,-1,725,-1,K3160,-1,-1,117277)

**К У Р С О В О Й   П Р О Е К Т**

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

Обучающийся: Рожнова Анастасия Игоревна, K3160

Санкт-Петербург 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_heading=h.gjdgxs)

[1 Суть проекта 6](#_heading=h.3znysh7)

[2 Процессы работы над проектом 9](#_heading=h.2et92p0)

[3 Проделанная работа и ее анализ 11](#_heading=h.tyjcwt)

[3.1 Проблема, поставленная передо мной 11](#_heading=h.3dy6vkm)

[3.2 Решение поставленной задачи 12](#_heading=h.1t3h5sf)

[3.3 Анализ проделанной работы 15](#_heading=h.2s8eyo1)

[4 Анализ взаимодействия с командой и руководителем проекта 17](#_heading=h.17dp8vu)

[4.1 Взаимодействие с командой 17](#_heading=h.3rdcrjn)

[4.2 Взаимодействие с руководителем 17](#_heading=h.26in1rg)

[4.3 Оценка работы руководителя 18](#_heading=h.35nkun2)

[ЗАКЛЮЧЕНИЕ 19](#_heading=h.1ksv4uv)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#_heading=h.2jxsxqh)

[ПРИЛОЖЕНИЕ 22](#_heading=h.3whwml4)

# ВВЕДЕНИЕ

Проект «iOS приложение для поиска картинок (Pinterest)» актуален как для студентов, занимающихся разработкой, так и для будущих потенциальных пользователей приложения. С точки зрения разработчиков, этот проект послужил нам идеальной платформой для освоения iOS-разработки. Наша команда, погрузившись в изучение языка Swift, получила бесценный опыт создания пользовательских интерфейсов и понимание принципов взаимодействия различных компонентов приложения.

В современных реалиях, когда доступность популярных сервисов может быть ограничена, создание альтернатив приобретает особую значимость. Наше приложение призвано стать надежной платформой для тех, кто ищет источник вдохновения и эстетического удовольствия в визуальном контенте.

Особое внимание в проекте уделено современным трендам: персонализации контента и уникальному пользовательскому опыту. Интерфейс приложения разработан с учетом современных требований к дизайну, сочетая в себе функциональность и эстетическую привлекательность.

Социальный аспект проекта отражает современные тенденции: растущую популярность визуального контента, стремление к самовыражению через изображения и потребность в обмене идеями. Приложение охватывает широкий спектр интересов: от моды и кулинарии до дизайна интерьеров и фотографии.

Проект также отвечает образовательным потребностям молодого поколения, предоставляя возможность погрузиться в мир мобильной разработки. Это особенно актуально в контексте современного тренда на изучение программирования и растущего интереса к технологическому сектору.

Практическая значимость проекта проявляется в развитии ключевых навыков разработчика: логического мышления, решения проблем, анализа информации и применения теоретических знаний на практике. Работа над приложением позволяет глубоко понять принципы мобильной разработки и получить реальный опыт создания продукта, востребованного современными пользователями.

Значимость проекта усиливается его двойной направленностью: он важен как для разработчиков, получающих практический опыт программирования, так и для конечных пользователей, ищущих удобную платформу для поиска и обмена визуальным контентом.

Цель проекта – за полтора отведенных месяца (с 01.11.2024 по 17.12.2024) разработать прототип iOS приложения, в котором есть возможность искать, детально просматривать и сохранять картинки, смотреть персональные подборки картинок, получая их с API Flickr.

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

1. изучить основы языка программирования Swift;
2. создать набор цветов приложения;
3. создать набор шрифтов приложения;
4. создать набор иконок приложения;
5. сделать дизайн экрана рекомендуемых картинок;
6. сделать дизайн экрана детального просмотра картинки;
7. сделать дизайн экрана профиля;
8. сделать дизайн экрана входа/авторизации;
9. сделать дизайн экрана поиска;
10. добавить шрифты в приложение;
11. добавить цвета в приложение;
12. добавить иконки в приложение;
13. изучить фреймворк UIKit;
14. разработать экран рекомендуемых картинок;
15. разработать экран детального просмотра картинки;
16. разработать экран поиска;
17. внедрить бизнес-логику на экран рекомендуемых картинок;
18. внедрить бизнес-логику на экран детального просмотра картинки;
19. внедрить бизнес-логику на экран поиска;
20. добавить API на экран рекомендованных картинок;
21. добавить API на экран детального просмотра картинки;
22. добавить API на экран поиска;
23. исправить ошибки, обнаруженные по ходу тестирования приложения.

Руководитель проекта определил данные задачи и распределил их среди членов команды. Для выполнения поставленных задач в команде были определены две роли: дизайнер и iOS-разработчик. За мной была закреплена роль iOS-разработчика.

# 1 Суть проекта

Проект «iOS приложение для поиска картинок (Pinterest)» представляет собой прототип мобильного приложения, ориентированного на пользователей платформы iOS, которые желают находить и сохранять изображения по своим интересам. В современном мире, где визуальный контент является неотъемлемой частью повседневной жизни, данное приложение предоставляет удобный инструмент для поиска изображений. Платформа создана для того, чтобы помочь пользователям в поиске вдохновляющих картинок, которые могут быть использованы для различных целей: от создания уникальных интерьерных решений до поиска идей для подарков своим близким.

Основная задача приложения – обеспечить пользователям доступ к обширной базе изображений, используя ключевые слова для поиска и алгоритмы рекомендаций, которые предлагали бы схожий контент. Благодаря данному механизму пользователи смогут просматривать ленту, составленную на основе их прошлых запросов и предпочтений, находя именно то, что будет их вдохновлять.

Целевой аудиторией приложения являются молодые люди, которые любят открывать для себя что-то новое. Эти пользователи интересуются качественными изображениями для личных и профессиональных целей, будь то подготовка впечатляющей презентации или поиск новых художественных идей.

Для удовлетворения запросов целевой аудитории в приложении реализованы следующие функции:

1. главный экран с индивидуальной подборкой картинок по интересам и истории запросов;
2. удобный интерфейс просмотра изображений с возможностью увеличения для детального изучения;
3. сохранение понравившихся изображений в память устройства для их последующего использования без подключения к сети;
4. поиск изображений по ключевым словам с предложениями рекомендаций, основанных на предыдущих запросах;
5. переключение между разными разделами, позволяющее пользователю легко перемещаться от личных рекомендаций к общему поиску.

Проект реализован на архитектуре VIPER (View-Interactor-Presenter-Entity-Router), которая способствует упрощению разработческой и тестировочной деятельности [1]. Эта архитектурная модель обеспечивает четкое разделение функциональных обязанностей:

* View отвечает за отображение данных пользователю;
* Interactor отвечает за бизнес-логику приложения, здесь выполняются операции выборки, обновления, доступ к API;
* Presenter – это своеобразный мост между View и Interactor;
* Entity – модельная часть приложения. Модели данных, связанные с приложением, находятся здесь;
* Router обеспечивает навигацию между экранами приложения и управляет переходами между ними.

Приложение было разработано для iOS с использованием языка программирования Swift. Этот язык применяется для создания как веб-приложений, так и мобильных приложений на устройствах от Apple. Среди его особенностей – простой и ясный синтаксис, а также защита от распространенных ошибок, связанных с невнимательностью, таких как неправильное использование типов данных или обращение к неинициализированным переменным. Дополнительно Swift поддерживает управление памятью, что особенно важно в условиях ограниченных ресурсов мобильных устройств [2].

Для разработки графического интерфейса нашего приложения был использован UIKit – фреймворк, предоставляющий обширный набор UI-элементов, включая заголовки, таблицы, коллекции, кнопки и изображения [3]. Это позволяет создать интерфейс, который будет одновременно удобным, функциональным и привлекательным. Основополагающим элементом проекта стал UICollectionView, с помощью которого нам и удалось воспроизвести интерфейс в стиле Pinterest.

Также для создания графического интерфейса была применена библиотека SnapKit, которая значительно упрощает использование Auto Layout – встроенной технологии компании Apple, ответственной за позиционирование и изменение размеров интерфейсных компонентов [4]. SnapKit позволяет задавать экранные ограничения всего в несколько строк, повышая темп и удобство разработки. Это также способствует улучшению читаемости и уменьшению сложности кода, облегчая его поддержку и информацию об ошибках. С помощью этой библиотеки можно создать интерфейс, который адаптируется к различным размерам и ориентациям экранов, обеспечивая комфорт пользователю.

Для обеспечения быстрого и надежного взаимодействия с внешними данными была интегрирована библиотека Alamofire, упрощающая работу с HTTP-запросами [5]. Она отвечает за получение изображений и связанных с ними данных, управляя сетевым взаимодействием и отображением информации пользователям.

# 2 Процессы работы над проектом

Наш проект проходил несколько ключевых этапов, чтобы быть успешно реализованным и соответствовать всем требованиям. Вначале мы провели созвон с магистрантом, где детально обсудили техническое задание. Это позволило нам четко определить все цели и задачи, которые предстояло решить в процессе работы над проектом. В диалоге мы также обсудили возникающие вопросы, установили правила взаимодействия, и руководитель распределил задачи между участниками, закрепив их на платформе Odoo.

После этого команда дизайнеров приступила к созданию макета приложения в Figma. Дизайнеры работали над разработкой интуитивно понятного и эстетически привлекательного интерфейса, уделяя внимание каждой детали, чтобы обеспечить пользователю максимальное удобство использования. В Figma они разработали брендбук с описанием шрифтов, их стилей и размеров, подобрали цветовую палитру с названиями и HEX-кодами, а также выбрали соответствующие иконки. Далее были сделаны макеты экранов для рекомендаций, детального просмотра, поиска, профиля и авторизации, которые впоследствии служили ориентиром для разработчиков.

Следующим ключевым этапом стала непосредственно разработка самого мобильного приложения. Магистрант организовал серию созвонов, где пояснил, какими средствами нужно пользоваться при кодировании, предложил полезные источники о библиотеках и поделился видеоматериалами с пояснениями. Первоначальной задачей стала интеграция шрифтов, цветов и иконок. Далее наша команда перешла к верстке экранов, таких как экраны поиска, рекомендации и детального просмотра изображения. Каждый из этих экранов требует тщательной проработки, чтобы все элементы интерфейса функционировали корректно и были логично объединены между собой. Также была разработана бизнес-логика приложения: создание основного функционала и алгоритмов, которые будут обеспечивать работоспособность всех задуманных функций. Этот этап оказался самым продолжительным из-за значительного объема работы и необходимости уделять внимание деталям.

На заключительном этапе была осуществлена интеграция сетевых запросов (API) в приложение. Это необходимо для связи приложения с внешними сервисами и базами данных, что позволяет пользователям получать актуальную информацию в реальном времени. На этом этапе мы активно работали с Interactor и Presenter, так как они отвечают за выполнение серверных запросов, обработку и отображение данных. Для успешной интеграции API требуется обеспечить надежность и безопасность передачи данных, а также протестировать все запросы на корректность и быстроту действий.

Таким образом, слаженная работа команды позволила нам шаг за шагом реализовать проект, следуя намеченному плану и учитывая все необходимые детали и требования, прописанные в техническом задании.

# 3 Проделанная работа и ее анализ

## 3.1 Проблема, поставленная передо мной

В начале работы над проектом передо мной были поставлены

следующие задачи:

1. перенести изображения (иконки) в приложение;
2. изучить библиотеку UIKit для работы с пользовательским интерфейсом;
3. разработать экран поиска картинок;
4. интегрировать бизнес-логику на экран поиска картинок;
5. добавить API для получения данных с сервера в разработанный модуль;
6. проверить код на наличие ошибок и корректность отображения элементов.

На начальном этапе работы над проектом передо мной была поставлена задача по разработке пользовательского интерфейса и реализации функциональности приложения с использованием библиотеки UIKit. Первая задача заключалась в переносе изображений (иконок) в приложение, которые были разработаны командой дизайнеров. Следующей задачей стало изучение библиотеки UIKit, которая является основным инструментом для работы с пользовательским интерфейсом в iOS. Понимание этой библиотеки было необходимо для грамотной разработки и настройки всех интерфейсных компонентов, чтобы они эффективно взаимодействовали между собой.

Третья задача сосредоточилась на разработке экрана поиска картинок, позволяющего пользователям легко находить и просматривать изображения. Интеграция бизнес-логики на данный экран также была важным этапом: это потребовало написания кода, который обеспечивал бы правильную фильтрацию и представление данных пользователю, учитывая различные запросы и предпочтения.

Добавление API для получения данных с сервера представляло следующую сложность. Необходимо было разработать отдельный модуль для взаимодействия с сервером, чтобы обеспечить быстрый поток данных, который может обновляться в реальном времени.

Наконец, проверка кода на наличие ошибок и корректность отображения элементов была крайне важной для обеспечения стабильности и надежности приложения. Это включало тестирование различных сценариев использования, устранение выявленных багов и оптимизацию работы интерфейса.

## 3.2 Решение поставленной задачи

Прежде всего, я начала с изучения языка программирования Swift, который является основным инструментом для разработки приложений на платформе iOS. Мое обучение включало в себя множество источников, таких как книги, обучающие видео и статьи в Интернете. Книги стали для меня основным источником теоретических знаний. Они охватывали основы языка, включая работу с переменными, функциями, объектами и контролируемыми структурами. Видеоуроки, в свою очередь, помогли мне увидеть, как использовать эти теоретические знания на практике, показав рабочие примеры кода.

После знакомства с языком программирования я перешла к следующему шагу – добавлению иконок в проект, работая в среде Xcode. Эти иконки, представленные на рисунке 1, были предварительно созданы дизайнерами и представляли собой важные визуальные элементы приложения. Основной задачей на этом этапе было правильное интегрирование визуальных материалов в среду разработки. Я изучила, как правильно настраивать размеры иконок для разных экранов устройств, и как строить ассеты, чтобы они корректно отображались в приложении.

Изображение выглядит как Шрифт, Графика, дизайн, белый

Автоматически созданное описание

Рисунок 1 – иконки, использовавшиеся в приложении

Следующим этапом стала реализация SearchBar (строка поиска) – ключевого элемента интерфейса, который пользователи будут использовать для ввода запросов. Решение интегрировать в поиск кнопку для удаления текста, текстовое поле и иконку лупы потребовало использования нескольких компонентов UIKit. SearchBar представлен на рисунке 2.

Используя UIButton, я создала интерактивную кнопку для очистки текста, что дало пользователям возможность быстро удалять введенные данные. UITextField использовался для создания текстового поля, куда пользователь мог вводить свои запросы, а UIImageView – для отображения стандартной иконки лупы, что интуитивно подсказывало бы пользователям функционал элемента.

Для объединения всех этих компонентов был использован UIStackView. Это позволило гибко настроить позиционирование и внешний вид элементов, обеспечив их правильное отображение вне зависимости от устройства.

Изображение выглядит как текст, Шрифт, белый, логотип

Автоматически созданное описание

Рисунок 2 – SearchBar, разработанный специально под наше приложение

Для реализации экрана поиска, который должен был отображать рекомендации, популярные запросы и историю поиска, я использовала UICollectionView. Эта структура позволила мне представить данные в виде ячеек, которые автоматически обновляются в зависимости от действий пользователя.

На начальной стадии, если пользователь не вводил текст, экран демонстрирует популярные запросы и рекомендации, что представлено на рисунке 3. Как только пользователь начинает ввод своего запроса, на экране начинает отображаться история его поиска, где можно нажать на один из предыдущих запросов и мгновенно повторить его, что продемонстрированно на рисунке 4.

Изображение выглядит как текст, снимок экрана, коллаж

Автоматически созданное описание

Рисунок 3 – экран поиска до начала ввода текста

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 4 – экран поиска после начала ввода текста

Один из наиболее важных этапов разработки заключался в интеграции API, обеспечивающей доступ к внешним данным и их обновление. Это позволило пользователям получать актуальные результаты поиска и отображать соответствующие изображения в зависимости от его запросов или выбранных категорий.

Для интеграции была проведена работа с библиотекой Alamofire для выполнения HTTP-запросов. Интеграция через Alamofire сделала процесс работы с API существенно проще, так как библиотека предлагает удобный интерфейс для написания запросов и получения ответов. Я изучила методы аутентификации и запроса данных от сервера, что позволило мне обеспечить корректное функционирование приложения.

В процессе создания проекта мы использовали систему контроля версий Git и платформу GitHub. Каждый раз, когда начиналась работа над новой функцией или исправлением, я создавалa новую отдельную ветку в Xcode, что позволяло изолировать изменения от основной ветки. Эта ветка затем синхронизировалась с нашим репозиторием на GitHub, предоставляя возможность руководителю проекта просматривать внесённые изменения и оставлять комментарии, что упрощало процесс обратной связи. После внесения всех необходимых улучшений и исправлений наш код проходил окончательную проверку и интегрировался в главную ветку проекта.

## 3.3 Анализ проделанной работы

Проанализировав процесс выполнения курсового проекта, я могу сказать, что работа оказалась для меня довольно сложной, но вполне выполнимой. Тем не менее, я считаю, что удачно справилась с возложенными на меня обязанностями. Наиболее объемными и ресурсозатратными задачами для меня были разработка бизнес-логики экрана поиск и создание SearchBar, так как прежде я никогда не сталкивалась с реализацией подобных вещей, и они потребовали от меня изучения обширного количества информации.

Что получилось:

* добавлены иконки в приложение;
* реализован модуль поиска с продуманной бизнес-логикой и внедренным API;
* разработан дополнительный функционал: возможность удаления введенного текста одной кнопкой, возможность очистить историю поиска.

Что вызвало некоторые трудности:

* недостаточный уровень знаний в области программирования, так как я только недавно начала обучение в университете;
* ошибки в коде, проблемы с интеграцией сторонних библиотек и работа с API;
* найти точную и корректную информацию в Интернете касательно решения моей проблемы не всегда выходило удачно, часто приходилось разбираться самой или обращаться за помощью к руководителю и команде;
* большой объем новой информации, которую нужно усвоить в кратчайшие сроки.

На протяжении всего проекта я старалась придерживаться намеченного плана, хотя некоторые задачи потребовали больше времени, чем планировалось. Основными факторами, мешавшими планомерной работе, стали неожиданные технические сложности и необходимость изучения большого количества нового материала в сжатые сроки.

Во время работы над курсовым проектом я существенно расширила свои знания в области программирования на языке Swift и работы с Xcode. Освоение UIKit и коллекционных видов представления данных позволило мне глубже понять принципы разработки интерфейсов для мобильных приложений. В результате я также научилась эффективно интегрировать внешние API в приложения.

Таким образом, этот проект стал для меня ценным опытом, который поспособствовал моему профессиональному развитию, а также он дал мне новые знания о современных методах разработки приложений.

# 4 Анализ взаимодействия с командой и руководителем проекта

## 4.1 Взаимодействие с командой

В ходе выполнения нашего курсового проекта мы создали дружескую рабочую атмосферу в команде. Все участники команды активно взаимодействовали, внося свой вклад в проект. Основным средством связи для нашей команды стал чат в Telegram. В нем мы оперативно обсуждали возникающие вопросы, обменивались мнениями и делились важной информацией.

Наряду с текстовой перепиской, мы регулярно организовывали видеозвонки в Google Meet. Эти созвоны помогли нам лучше понимать друг друга, более глубоко обсуждать текущие задачи и планировать дальнейшие шаги проекта.

Для обеспечения высокого уровня организации мы использовали общий репозиторий на GitHub. Это помогло нам отслеживать прогресс друг друга и своевременно вносить изменения в проект, если этого потребует руководитель.

## 4.2 Взаимодействие с руководителем

На протяжении всего периода работы над проектом мы постоянно поддерживали связь с нашим руководителем. Он всегда оказывал нам поддержку, помогал решать возникающие проблемы на каждом этапе работы. Руководитель не только давал ценные советы и рекомендации, но и проявлял терпение и снисходительность при разъяснении даже самых простых материалов, если мы испытывали трудности в их понимании. Часто он отмечал наши достижения, что мотивировало нас на дальнейшую работу. Благодаря его профессионализму и поддержке процесс работы оказался не только продуктивным, но и очень приятным.

Особенно ценным было его умение разъяснять материал, включая самые простые концепции, если мы испытывали затруднения в их понимании. Его подход заключался в приведении различного рода метафор и аналогий, чтобы показать нам, как именно всё работает и устроено. Нередко он созванивался с нами, чтобы лично объяснить, как работает определенный код, и как двигаться дальше.

## 4.3 Оценка работы руководителя

Наш руководитель продемонстрировал исключительный профессионализм и поддержку на протяжении всего проекта. Его терпение и готовность разбираться с любыми трудностями, которые мы испытывали, создали атмосферу доверия между нами. Благодаря его вовлечённости мы чувствовали себя уверенно и не боялись задавать любые вопросы. Он не только указывал на ошибки, но и отмечал наши достижения, что значительно повышало нашу самооценку.

В целом, руководитель проявил себя как бесценный наставник и лидер. Его профессионализм и чуткий подход способствовали нашей продуктивности и сделали процесс работы приятным и увлекательным. Благодаря его усилиям мы не только завершили проект успешно, но и приобрели ценные знания и навыки, которые точно будут полезны нам в будущем.

# ЗАКЛЮЧЕНИЕ

Проект «iOS-приложения для поиска изображений (Pinterest)» был начат 1 ноября 2024 года с целью создания прототипа, который позволит пользователям искать, просматривать и сохранять изображения, а также просматривать персональные подборки. Окончание разработки было запланировано на 17 декабря 2024 года. Поставленные цели также включали в себя интеграцию с API для получения изображений и обеспечение удобного пользовательского интерфейса.

В течение полтора месяца работы значительная часть функционала была успешно реализована. Были разработаны экран поиска, который позволяет пользователю вводить запросы и получать соответствующие результаты изображений через интеграцию с API Flickr; пользователи имеют возможность детального просмотра изображений, а также их сохранения для последующего использования; реализован алгоритм для формирования персональных рекомендаций на основе истории поиска пользователя.

К сожалению, некоторые из намеченных задач не были выполнены из-за ограничений во времени и нехватки опыта в сфере разработки: не были реализованы модули профиля и авторизации. Также, первоначально выбранная библиотека RxAlamofire не поддерживалась у одного из членов команды, и было принято решение перейти на Alamofire.

Мой личный вклад в проект заключается в реализации нескольких масштабных задач. Я была ответственна за полную разработку интерфейса и функциональности экрана поиска. Это включало работу с пользовательским вводом, отображение результатов и структурирование данных. Мной была реализована связь данных приложения с внешним API, что предусматривало как обработку данных с сервера, так и оптимизацию их отображения на экране устройства. Также я выполнила перенос всех необходимых изображений, созданных командой дизайнеров, в приложение.

Подводя итог, можно сказать, что наш проект достиг своей основной цели – создания работоспособного прототипа в ограниченные сроки. Несмотря на некоторые нереализованные функции, вся команда проделала значительную работу. В ходе работы над проектом мне удалось освоить новый для меня язык программирования, углубиться в незнакомую среду разработки и познакомиться с мобильной разработкой в целом, с которой ранее я не имела дела. Этот проект стал для меня важным этапом в развитии и дал ценнейший опыт, который я планирую использовать в дальнейшем.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архитектура VIPER: простыми словами — URL: <https://apptractor.ru/info/articles/arhitektura-viper-prostymi-slovami.html> (дата обращения 03.11.2024)
2. Язык программирования Swift: где используется, основы — URL: [https://blog.skillfactory.ru/glossary/swift/#:~:text=Swift%20—%20это%20язык%20программирования%20от,реже%20используется%20в%20других%20проектах.&text=Язык%20появился%20всего%20в%202014,Apple%20открыла%20его%20исходный%20код](https://blog.skillfactory.ru/glossary/swift/#:~:text=Swift%20%E2%80%94%20%D1%8D%D1%82%D0%BE%20%D1%8F%D0%B7%D1%8B%D0%BA%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%BE%D1%82,%D1%80%D0%B5%D0%B6%D0%B5%20%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D0%B5%D1%82%D1%81%D1%8F%20%D0%B2%20%D0%B4%D1%80%D1%83%D0%B3%D0%B8%D1%85%20%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%D1%85.&text=%D0%AF%D0%B7%D1%8B%D0%BA%20%D0%BF%D0%BE%D1%8F%D0%B2%D0%B8%D0%BB%D1%81%D1%8F%20%D0%B2%D1%81%D0%B5%D0%B3%D0%BE%20%D0%B2%202014,Apple%20%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D0%BB%D0%B0%20%D0%B5%D0%B3%D0%BE%20%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%BE%D0%B4). (дата обращения 03.11.2024)
3. UIKit | Apple Developer Documentation — URL: <https://developer.apple.com/documentation/uikit/> (дата обращения 10.11.2024)
4. iOS-разработка со SnapKit: спасаем консоль от простыни ошибок — URL: <https://habr.com/ru/companies/sravni/articles/719474/> (дата обращения 10.11.2024)
5. Работа с сервером с помощью Alamofire на Swift — URL: <https://habr.com/ru/articles/330760/> (дата обращения 27.11.2024)

# ПРИЛОЖЕНИЕ

**Техническое задание для iOS приложения для поиска картинок (Pinterest)**

[Описание проекта](#_heading=h.2bn6wsx)

[Модули МП](#_heading=h.ihv636)

[Описание модулей МП](#_heading=h.1pxezwc)

[Модуль авторизации/регистрации](#_heading=h.49x2ik5)

[Модуль рекомендуемых картинок](#_heading=h.32hioqz)

[Модуль поиска](#_heading=h.2p2csry)

[Модуль детального просмотра картинки с возможностью сохранить её на устройство](#_heading=h.147n2zr)

[Модуль профиля](#_heading=h.1hmsyys)

[Способы навигации](#_heading=h.41mghml)

[Возможные улучшения приложения](#_heading=h.3o7alnk)

[Добавить работу с пользовательскими бордами и пинами:](#_heading=h.23ckvvd)

[Добавить в модуль рекомендуемых картинок при открытии с поиска уточнения к поиску.](#_heading=h.2grqrue)

[Модуль входа по пинкоду и биометрии](#_heading=h.vx1227)

Основное назначение проекта

iOS приложения, 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

Модуль авторизации/регистрации

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](https://github.com/parkerdonat/ios-pdk?tab=readme-ov-file)

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Модуль рекомендуемых картинок

Модуль рекомендуемых картинок состоит из следующих секций:

Секция с названием бордов состоит только из текста с возможностью горизонтального скролла. 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.

Изображение выглядит как Красочность, Прямоугольник, прямоугольный, дизайн

Автоматически созданное описаниеИзображение выглядит как снимок экрана, Красочность, Прямоугольник, дизайн

Автоматически созданное описание

Модуль поиска

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.

Изображение выглядит как снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

Модуль профиля

Модуль профиля состоит 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

Изображение выглядит как снимок экрана, Красочность, Прямоугольник, дизайн

Автоматически созданное описание

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

Изображение выглядит как снимок экрана, Красочность, текст, Прямоугольник

Автоматически созданное описание

Модуль входа по пинкоду и биометрии

Модуль входа по пинкоду 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 пинкода, 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 пинкод”,** 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 соответствующе 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 пинкод”** состоит из заголовка, подзаголовка и двух кнопок: войти по биометрии и заново авторизоваться. Если пользователь ранее не добавлял биометрию, то кнопка “войти по биометрии” отсуствует.

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

Для реализации обязательна библиотека LocalAuthentication.

Изображение выглядит как текст, снимок экрана, Шрифт, круг

Автоматически созданное описание