

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Разработка UI iOS-приложения на основе SwiftUI

СОГЛАСОВАНО

Руководитель:

ФИО руководителя

\_\_\_\_\_  
(подпись руководителя)

Исполнитель:

ФИО исполнителя

\_\_\_\_\_  
(подпись исполнителя)

Санкт-Петербург

2024

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	4
ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ .....	5
3.1 Требования к функциональным характеристикам .....	5
3.2 Требования к интерфейсу.....	6
3.3 Требования к надежности .....	6
3.4 Требования к составу и параметрам технических средств.....	6
3.5 Требования к информационной и программной совместимости.....	6
СТАДИИ И ЭТАПЫ РАЗРАБОТКИ.....	8
4.1 Анализ и проектирование .....	8
4.2 Разработка интерфейса .....	8
4.3 Разработка функциональности .....	9
4.4 Тестирование .....	9
4.5 Внедрение .....	9

## ВВЕДЕНИЕ

Данное техническое задание предназначено для разработки мобильного приложения, предоставляющего пользователям основную информацию о библиотеках Петроградского района и ближайших мероприятиях, проводимых в этих библиотеках.

Целевая аудитория: посетители библиотек Петроградского района с устройствами на базе iOS 15 и выше.

Область применения: мобильное приложение, содержащее разделы с информацией о библиотеках, ближайших событиях и дополнительной информацией для читателей.

Объект применения: мобильное приложение предназначено для использования Центральной библиотечной системы (ЦБС) Петроградского района.

Для начальной версии приложения (MVP) предполагается реализация трех основных разделов:

- Мероприятия — предоставляет пользователю актуальный список мероприятий, проводящихся в библиотеках Петроградского района, с возможностью фильтрации по типу событий.
- Библиотеки — предоставляет информацию о библиотеках района, включая их расположение и актуальный график работы.
- Читателю — включает два подраздела:
  - Информация — ответы на часто задаваемые вопросы.
  - Мои книги — позволяет пользователям получить список книг, находящихся у них на руках.

## НАЗНАЧЕНИЕ РАЗРАБОТКИ

Функциональное назначение: предоставлять посетителям библиотек актуальные сведения о библиотеках и ближайших мероприятиях, проводимых в данных библиотеках.

Эксплуатационное назначение: приложение разрабатывается для мобильных устройств на базе iOS и будет доступно пользователям с iOS 15 и выше.

## ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

### *3.1 Требования к функциональным характеристикам*

Для начальной версии приложения должны быть реализованы три основных раздела: Мероприятия, Библиотеки, Читателю.

Основной функционал раздела мероприятия – получение данных о ближайших мероприятиях и представление их пользователю в виде вертикального списка карточек, с возможностью фильтрации по типу мероприятия. Каждая карточка должна являться кликабельным объектом и вести на отдельную страницу мероприятия. При открытии страницы должны запрашиваться актуальные данные и предоставляться пользователю.

Функционал раздела “Библиотеки” – получение актуальных данных о библиотеках и предоставление их пользователю в виде вертикального списка карточек, по аналогии с разделом “Мероприятия”. Каждая карточка должна являться кликабельным объектом и вести на отдельную страницу библиотеки. При открытии страницы должны запрашиваться актуальные данные о библиотеке и предоставляться пользователю. При открытии страницы библиотеки должны запрашиваться данные о мероприятиях, проводящихся в данной библиотеке и также предоставляться пользователю. Фильтрация списка ближайших мероприятий должна выполняться на клиенте.

Раздел “Читателю” должен содержать два подраздела: “Информация” и “Мои книги”. Подраздел “Информация” должен состоять из вертикального списка наиболее популярных вопросов. Каждый отдельный вопрос представляет собой разворачивающийся элемент. В развернутом состоянии должен отображаться ответ на вопрос. Подраздел “Мои книги” должен содержать поле ввода номера читательского билета. После ввода пользователь должен иметь возможность получить список книг, которые находятся у него на руках.

Навигация между разделами должна происходить с помощью навигационной панели расположенной внизу экрана. Навигация между подразделами должна осуществляться с помощью верхней навигационной панели, реализованной только для раздела “Читателю”. Возврат на прошлый экран со страниц библиотек и мероприятий должен осуществляться с помощью отдельной кнопки “Назад”. При открытии страницы библиотеки или мероприятия нижняя навигационная панель должна скрываться.

### *3.2 Требования к интерфейсу*

Интерфейс должен быть нативным и соответствовать гайдлайнам, которые предоставляет компания Apple для мобильной разработки. Интерфейс должен быть адаптивным и корректно отображаться на различных мобильных устройствах на базе iOS. Интерфейс должен корректно масштабироваться для экранов диагональю от 4.7 до 6.9 дюймов. Элементы страниц должны корректно отображаться на устройствах с кнопкой и без кнопки “Домой”.

### *3.3 Требования к надежности*

Приложение должно стабильно работать при устойчивом интернет-соединении. Должны корректно обрабатываться ошибки, возникающие на клиенте или сервере.

### *3.4 Требования к составу и параметрам технических средств*

Рекомендуемые технические средства: мобильное устройство с iOS 15 или выше, процессор Apple A13 Bionic или выше

### *3.5 Требования к информационной и программной совместимости*

Приложение должно поддерживать взаимодействие через API для предоставления актуальной информации о библиотеках и мероприятиях. Приложение должно использовать защищённый протокол HTTPS для всех операций, связанных с передачей данных между клиентом и сервером. Для

этого сервер должен поддерживать TLS (Transport Layer Security) с актуальной версией протокола. Используемые технологии для разработки нативного приложения: SwiftUI. SwiftUI будет использоваться для создания пользовательского интерфейса, что позволит эффективно строить декларативный UI, адаптируемый под различные размеры экранов и устройства. SwiftUI позволит ускорить процесс разработки, обеспечивая более простую работу с состояниями и анимациями.

## СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

### 4.1 Анализ и проектирование

На стадии анализа и проектирования проводятся следующие мероприятия:

*Сбор требований:* Анализируются требования заказчика, спецификации и пожелания пользователей. Оцениваются функциональные и нефункциональные требования, такие как производительность, безопасность и совместимость.

*Анализ существующих решений:* Производится оценка текущих решений для библиотек, а также анализ других приложений или сервисов, которые могут быть аналогичны по функционалу или технологии.

*Согласование форматов ответов сервера:* Выполняется согласование форматов данных, которые сервер будет отправлять на клиент. Это включает в себя выбор формата JSON, создание схемы для всех ответов API (например, использование Swagger для документирования API), а также проверку правильности кодировки и структуры данных. Важно, чтобы API соответствовало стандартам, и данные передавались в корректном и согласованном формате, что позволит избежать ошибок при интеграции клиента с сервером.

### 4.2 Разработка интерфейса

На этапе разработки интерфейса выполняются следующие задачи:

*Создание прототипов пользовательского интерфейса:* На основе согласованных, на первом этапе требований проектируется схема взаимодействия клиента и сервера, определяются функции, которые будут выполняться на клиенте, а также определяется дополнительный функционал, который будет реализован на клиенте, например фильтрация.

*Разработка экранов:* Разрабатываются основные части интерфейса, а так же реализуется функционал, который на данном этапе не подключается к серверу.



*Проверка интерфейса с использованием моков:* Проводится тестирование интерфейса и взаимодействия с пользователем с использованием моков (временных фиктивных данных).

#### *4.3 Разработка функциональности*

Этап разработки функциональности включает:

*Реализация основных функций обработки данных:* На этом этапе реализуется основная логика взаимодействия с сервером, такая как обработка запросов, к базе данных и API, для получения данных, которые будут отображены на клиенте.

*Интеграция с сервером и API:* Производится интеграция с сервером для получения актуальной информации. Проверяется корректность обработки получаемых данных и их отображения

#### *4.4 Тестирование*

Каждый из этапов должен сопровождаться smoke и sanity тестированием, в частности при разработке нового функционала. Это позволит проверять работоспособность нового функционала и влияние изменений на ранее разработанный функционал. Smoke тестирование должно применяться при проверке старого функционала. Sanity тестирование необходимо для проверки нового функционала. После завершения работы над MVP функционалом и подключения сервера, должно быть проведено регрессионное тестирование, которое позволит удостовериться в качестве продукта, а также исправить критические проблемы в случае их обнаружения.

#### *4.5 Внедрение*

После завершения разработки и тестирования приложение размещается в TestFlight для проведения бета-тестирования силами посетителей библиотеки.