

Техническое задание на разработку  
сервиса  
«Subbik»

Версия 1.2

**История изменения документа**

Версия	Дата	Автор	Изменения
1.1	1.12.2025	Пинчук М. А.	Начало написания документа
1.2	17.12.2025	Пинчук М. А.	Добавления основных положений ТЗ
1.3	12.01.2026	Слесарчук Д.Т.	Дополнение и исправление основных положений ТЗ

Общие Сведения.....	4
Плановые сроки начала и окончания работы по созданию приложения .....	4
Назначения и цели создания системе.....	4
Функциональные требования к системе .....	5
Нефункциональные требования к системе.....	9
Архитектура системы .....	13
Приложения .....	15
Цветовая палитра .....	15
Макет интерфейса .....	15

## Общие Сведения

Мобильное приложение «Subbik», сервис который представляет собой, инструмент для отслеживания подписок, их количества, цены и времени оплаты.

Заказчиком и исполнителем являются Пинчук Максим Андреевич и Слесарчук Дмитрий Денисович.

Плановые сроки начала и окончания работы по созданию приложения

Этапы создания информационной системы с плановыми сроками представлены в таблице.

Этап	Сроки
Проектирование системы	15 сентября – 26 октября (6 недель)
Разработка системы	27 октября – 28 декабря (8 недель)
Тестирование системы	12 января – 29 января (3 недели)

Стоимость разработки включает в себя три сосиски в тесте и чай, по возможности.

## Назначения и цели создания системе

Система предназначена для отслеживания подписок и трат на них.

Цели создания системы с точки зрения заказчика и пользователей:

- Предоставить возможность пользователям видение своих списков подписок.
- Упростить отслеживание оплаты подписок.

# Функциональные требования к системе

## Регистрация нового пользователя

- Любой пользователь может создать новую учётную запись в системе.
- При регистрации пользователь должен заполнить данные в форме:  
*Логин пользователя* — обязательное поле, должно быть уникальным в системе. *Электронная почта пользователя* — обязательное поле, используется для идентификации. *Пароль пользователя* — обязательное поле, должен быть подтверждён повторным вводом. *Подтверждение пароля* — обязательное поле, должно совпадать с паролем.
- Система должна проверять заполнение всех обязательных полей перед созданием учетной записи.
- Система должна проверять уникальность логина в базе данных.
- Система должна проверять совпадение пароля и подтверждения пароля.
- При успешной регистрации пользователь получает уведомление и может перейти к авторизации.

## Авторизация пользователя

- Пользователь может войти в систему, используя логин и пароль.
- При авторизации пользователь должен ввести данные в форму: *Логин пользователя* — обязательное поле. *Пароль пользователя* — обязательное поле.
- Перед авторизацией система должна проверить заполнение всех обязательных полей.

- Система должна проверить соответствие введенных данных учетной записи в базе данных.
- При успешной авторизации пользователь переходит на экран со списком подписок.
- При неудачной авторизации пользователь получает уведомление об ошибке.

## **Добавление новой подписки в список пользователя**

- Авторизованный пользователь может добавить новую подписку в свой список.
- При добавлении подписки пользователь должен заполнить данные в форме: *Название подписки* — обязательное поле, название сервиса подписки. *Стоимость подписки* — обязательное поле, указывается в числовом формате (например, 499,99). *Дата подписки* — обязательное поле, указывается в формате ГГГГ-ММ-ДД (например, 2024-01-15). *Значок подписки* — обязательное поле, выбирается из доступных вариантов (Netflix, Okko и другие).
- Система должна проверять заполнение всех обязательных полей перед сохранением подписки.
- Система должна проверять правильность формата даты (ГГГГ-ММ-ДД).
- Система должна проверять правильность формата стоимости (числовое значение).
- По умолчанию в поле даты устанавливается текущая дата.
- По умолчанию в поле стоимости устанавливается значение 0,0.

- При успешном добавлении подписка сохраняется в базе данных и отображается в списке пользователя.

## **Просмотр списка подписок в профиле пользователя**

- Авторизованный пользователь может просматривать список своих подписок.
- По умолчанию список подписок отсортирован по дате добавления, от новых к старым.
- Список подписок отображается в виде таблицы, содержащей следующую информацию: *Значок подписки* — визуальное представление сервиса подписки. *Название подписки* — название сервиса подписки. *Стоимость подписки* — стоимость подписки. *Дата подписки* — дата добавления подписки.
- Пользователь может просматривать все свои подписки на одном экране.
- При возвращении на экран со списком подписок данные автоматически обновляются из базы данных.

## **Изменение подписки в списке пользователя**

- Авторизованный пользователь может редактировать данные подписки, выбрав соответствующую опцию.
- Форма редактирования подписки аналогична форме добавления новой подписки.
- При редактировании пользователь может изменить следующие данные: *Название подписки* — обязательное поле. *Стоимость подписки* — обязательное поле. *Дата подписки* — обязательное поле. О *Значок подписки* — обязательное поле.

- Система должна проверять заполнение всех обязательных полей перед сохранением изменений.
- Система должна проверять правильность формата даты и стоимости.
- При успешном редактировании изменения сохраняются в базе данных и отображаются в списке.

### **Удаление подписки из списка пользователя**

- Авторизованный пользователь может удалить любую подписку из своего списка с помощью соответствующего инструмента.
- При удалении подписки полностью удаляется из базы данных пользователя.
- После удаления список подписок автоматически обновляется.

### **Валидация данных**

- Система должна проверять все обязательные поля перед сохранением данных.
- Система должна проверять формат даты (ГГГГ-ММ-ДД) при вводе.
- Система должна проверять формат стоимости (числовое значение) при вводе.
- Система должна проверять уникальность логина при регистрации.
- Система должна проверять совпадение паролей при регистрации.
- При обнаружении ошибок валидации пользователь получает уведомление с описанием ошибки.

### **Управление данными пользователя**

- Данные пользователя хранятся локально в базе данных SQLite на устройстве.

- Каждый пользователь имеет доступ только к своим данным.
- Подписки пользователя привязаны к его учётной записи через логин.
- При удалении пользователя все его подписки удаляются из базы данных (каскадное удаление).

## Нефункциональные требования к системе

### Требования к структуре и функционированию системы

- Описание перечня подсистем, их назначения, компонентов системы и их информационного взаимодействия между собой представлено в разделе «Архитектура системы».
- Система представляет собой мобильное приложение для Android, работающее локально на устройстве пользователя.
- Система использует локальную базу данных SQLite для хранения данных пользователей и подписок.
- Система должна работать автономно, без необходимости подключения к интернету для выполнения базовых функций.
- Система должна обеспечивать быстрый отклик на действия пользователя (время отклика не более 500 мс для стандартных операций).
- Система должна корректно работать на устройствах с Android версии 7.0 (уровень API 24) и выше.

### Требования к численности и квалификации персонала системы

- Заказчик системы единолично имеет полный доступ ко всем возможностям системы и обладает необходимой квалификацией для этого.

- Для разработки и поддержки системы требуется разработчик с квалификацией в области разработки под Android и работы с базами данных SQLite.
- Требований к режиму работы персонала нет.

## **Требования к безопасности**

- Все пароли пользователей должны храниться в базе данных в открытом виде (для MVP-версии).
- Для регистрации пользователей необходимо указать логин, адрес электронной почты и пароль.
- База данных приложения располагается локально на устройстве пользователя в защищенной директории приложения.
- Доступ к данным пользователя ограничен только его учетной записью — каждый пользователь видит только свои подписки.
- При удалении учетной записи пользователя все связанные с ней данные (подписки) автоматически удаляются из базы данных.

## **Требования к интерфейсу**

- Аккуратный, минималистичный дизайн.
- Цветовая схема должна соответствовать рекомендациям Material Design 3.
- Приложение должно поддерживать светлую и тёмную темы в соответствии с системными настройками устройства.
- Пользователь может изменить тему интерфейса в системных настройках Android.

- Все элементы интерфейса должны быть выполнены с использованием компонентов Material Design для упрощения разработки и обеспечения единства.
- Интерфейс должен быть адаптивным и корректно отображаться на экранах Android-устройств с разным размером.
- Все текстовые элементы должны быть читабельными и иметь достаточный контраст в обеих темах (светлой и тёмной).

## **Требования к производительности**

- Приложение должно запускаться не более чем за 3 секунды на стандартных устройствах Android.
- Операции с базой данных (добавление, редактирование, удаление подписок) должны выполняться не более чем за 500 мс.
- Загрузка списка подписок должна выполняться не более чем за 1 секунду.
- Приложение должно эффективно использовать ресурсы устройства (память, процессор, аккумулятор).

## **Требования к локализации**

- Интерфейс приложения должен поддерживать русский язык.
- Все текстовые сообщения, уведомления и подсказки должны быть на русском языке.
- Формат даты должен соответствовать стандартам локализации (ГГГГ-ММ-ДД).
- В формате числовых значений (стоимости) в качестве разделителя десятичных знаков должна использоваться точка.

## **Требования к совместимости**

- Приложение должно работать на устройствах с Android версии 7.0 (уровень API 24) и выше.
- Приложение должно корректно работать как на телефонах, так и на планшетах.
- Приложение должно поддерживать различные ориентации экрана (книжную и альбомную).
- Приложение должно корректно работать с различными разрешениями экрана и плотностью пикселей.

# Архитектура системы

## Основные сведения

Мобильное приложение «Subbik», сервис который представляет собой, инструмент для отслеживания подписок, их количества, цены и времени оплаты.

Архитектура системы основана на локальной клиентской модели и разделена на три слоя: базовый, дополнительный и база данных.

## Компоненты архитектуры

### Базовый слой

- Экран авторизации (MainActivity) — вход пользователя в систему
- Экран регистрации (AuthActivity) — создание новой учётной записи
- Список подписок (ItemsActivity) — отображение всех подписок пользователя
- Добавление подписки (NoteActivity) — создание новой подписки
- Адаптер списка (ItemsAdapter) — отображение элементов в RecyclerView

### Дополнительный слой

- Помощник БД (DbHelper) — работа с базой данных SQLite через SQLiteOpenHelper
- Модель пользователя (User) — структура данных пользователя
- Модель подписки (Note) — структура данных подписки из базы данных

- Модель элемента (Item) — структура данных для отображения в интерфейсе

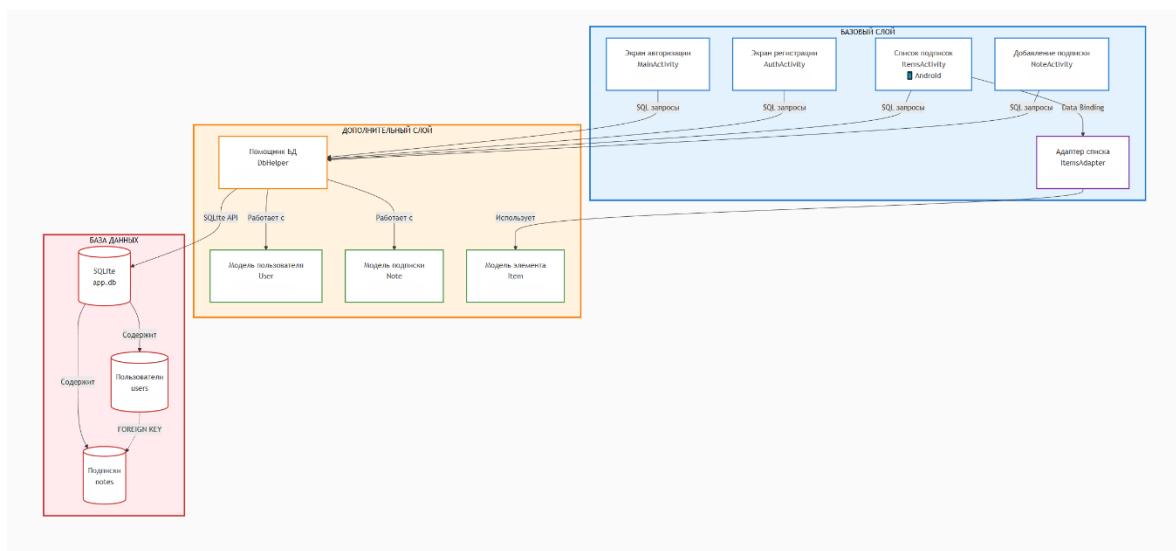
## База данных

- SQLite (app.db) — локальная база данных приложения
- Таблица пользователей (users) — хранение данных пользователей
- Таблица подписок (notes) — хранение подписок пользователей

## Протоколы взаимодействия

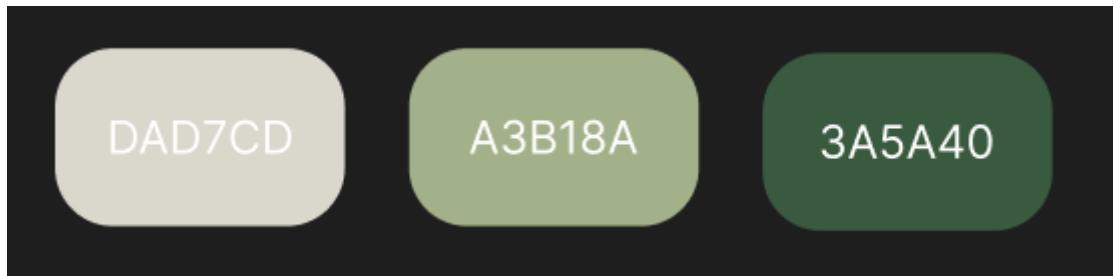
- SQL-запросы — действия взаимодействуют с DBHelper через SQL-запросы
- SQLite API — DBHelper использует SQLite API для работы с базой данных
- Привязка данных — ItemsActivity связывает данные с адаптером через привязку данных
- ВНЕШНИЙ КЛЮЧ — связь между таблицами users и notes

## Схема архитектуры



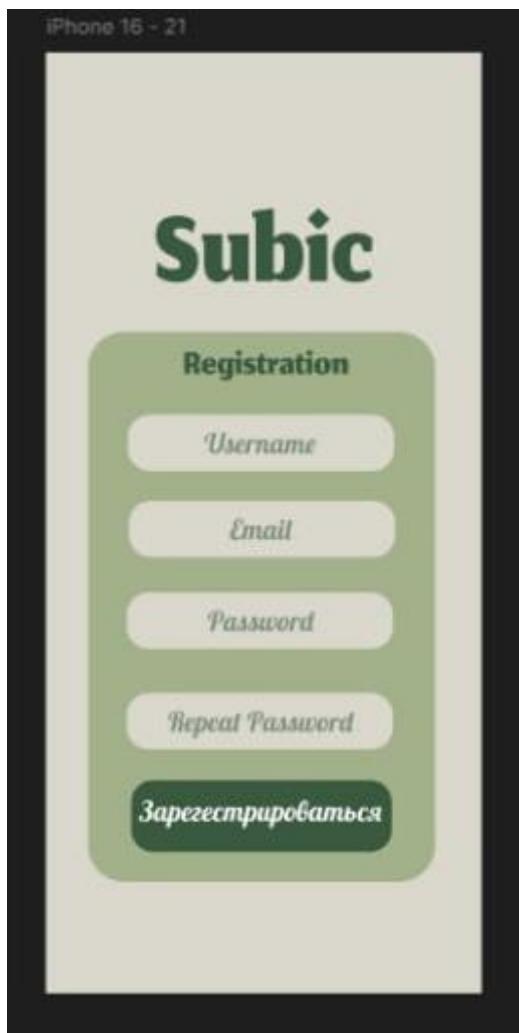
## Приложения

Цветовая палитра



Макет интерфейса

Страница регистрации



## Страница входа



## Страница с подписками



## Страница календаря подписок



## Страница создания трекера подписки

