|  |
| --- |
| Министерство науки и высшего образования РФ |
| Федеральное государственное автономное |
| образовательное учреждение высшего образования |
| **«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»** |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

|  |
| --- |
|  |
| **КУРСОВОЙ ПРОЕКТ** |
| Разработка веб-сайта с каталогом для компании «Крылатые Качели» на Django и React |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| Преподаватель | |  |  |  | П.В. Пересунько |
|  |  |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ23-17/2б, 032322546 |  |  |  | Е.А. Гуртякин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc199926887)

[Цель проекта 3](#_Toc199926888)

[Задачи 3](#_Toc199926889)

[1 Проектирование 4](#_Toc199926890)

[2. Разработка минимального стоящего продукта 6](#_Toc199926891)

[2.1 Первый спринт 6](#_Toc199926892)

[2.2 Второй спринт 8](#_Toc199926893)

[2.3 Третий спринт 9](#_Toc199926894)

[2.4 Четвертый спринт 10](#_Toc199926895)

[3. Подготовка к релизу 11](#_Toc199926896)

[3.1 Пятый спринт 11](#_Toc199926897)

[3.2 Шестой спринт 12](#_Toc199926898)

[3.3 Седьмой спринт 13](#_Toc199926899)

[4. Инструкция использования 14](#_Toc199926900)

[ПРИЛОЖЕНИЕ А 18](#_Toc199926901)

[ПРИЛОЖЕНИЕ Б 19](#_Toc199926902)

[ПРИЛОЖЕНИЕ В 20](#_Toc199926903)

## ВВЕДЕНИЕ

Для практически любого бизнеса нужен сайт. Такой сайт – «лицо» компании в современном мире. Без него, о бизнесе узнает гораздо меньше потенциальных клиентов, ведь они не смогут найти нужной (и что самое главное – бизнес-ориентированной) информации о нём.

Компания «Крылатые качели» – небольшая команда, специализирующаяся на проектировании и разработке детских площадок. Они уже спроектировали большое количество площадок в Абакане и собираются расширяться. Соответственно, им нужен хороший сайт. Нынешний сайт, к сожалению, не отвечает каким либо стандартам визуального дизайна и пользовательского опыта. По ссылке доступен старый сайт: https://kacheli19.ru/

Мне было предложено заняться разработкой нового сайта компании.

## Цель проекта

Целью создания сайта-каталога является получения новых навыков в веб-разработке и программировании, в частности, в дизайне и фронтенде. При работе с этим проектом я смогу получить опыт в full-stack разработке (разработке полного цикла) и деловой коммуникации с бизнес-заказчиком. Работа на настоящий бизнес является очень ценной для современного веб-разработчика, а опыт, полученный в работе с бизнесом, позволяет намного легче найти работу.

## Задачи

Для разработки сайта нужно выполнить много задач из разных сфер программирования: devops, frontend, backend и дизайна. Каждая из этих задач выполняется специалистом соответствующих профессий. При full-stack разработке придётся применить необходимые навыки из каждой профессии и сделать всё самому.

После опроса заказчика были составлены ключевые качества проекта, которые нужны бизнесу, а именно:

* простой и минималистский дизайн;
* удобство администрирования;
* гибкая реализация с заделом на возможное расширение функционала.

Таким образом, основными задачами будут создание дизайна и проектирование бэкенда с учётом требования удобства администрирования

## 1 Проектирование

1. Просмотр главной:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
2. Просмотр тематики изделий:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на плитку с тематикой.
   4. Сайт отображает страницу каталога тематик c плитками категорий изделий.
3. Просмотр группы изделий:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на плитку с тематикой.
   4. Сайт отображает страницу каталога тематик c плитками категорий изделий.
   5. Пользователь нажимает на плитку категории изделия.
   6. Сайт отображает страницу каталога группы изделий с плитками изделий.
4. Просмотр изделия:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на плитку с тематикой.
   4. Сайт отображает страницу каталога тематик c плитками категорий изделий.
   5. Пользователь нажимает на плитку категории изделия.
   6. Сайт отображает страницу каталога группы изделий с плитками изделий.
   7. Пользователь нажимает на плитку изделия.
   8. Сайт отображает страницу информации об изделии.
5. Просмотр страницы контактов:
   1. Пользователь открывает сайт по ссылке
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на ссылку «Контакты» в хэдере страницы
   4. Сайт отображает страницу контактов

Далее показана диаграмма использования

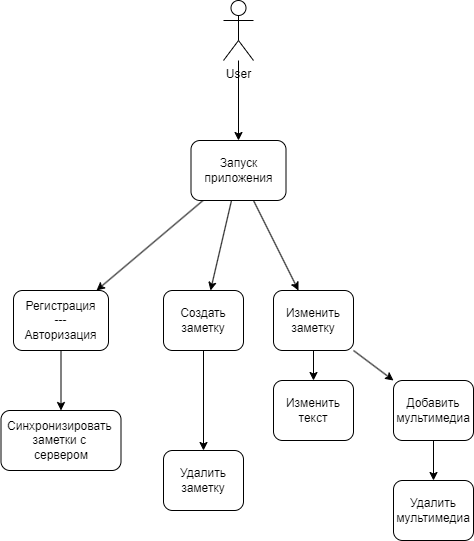


Рисунок 1 – Диаграмма использования

Сама разработка происходит в формате спринтов, где каждые две недели необходимо брать задание, выполнять его и переходить к следующему заданию.

## 2. Разработка минимального стоящего продукта

Далее представлены спринты, на которых был разработан минимально стоящий продукт, который может выполнять базовые операции с заметками.

## 2.1 Первый спринт

Задачи первого спринта были направлены на старт проекта. Проектирование приложения, выбор библиотек, создание структуры. Описание задач представлено в таблице 1.

Таблица 1 — Задачи на 1 спринт

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Приоритет |
| Продумать модульность приложения | Есть много разных способов разбить приложение на модули, но основные это горизонтальная или вертикальная. | Высокий |
| Выбрать архитектуру приложения | Clean Architecture – одна из самых популярных архитектур, но под большие проекты она плохо подходит | Высокий |
| Исследовать и выбрать библиотеку для работы с сетью | На выбор имеются Retrofit и OkHttp. Нужно узнать минусы и плюсы каждой | Средний |
| Создать базовую структуру проекта | Создание окружения, подбор нужной JDK и так далее | Высокий |

Первой задачей было продумывание модульности. В Android разработке присутствуют разные методы разделения кода, но по итогу был выбран метод разделения на независимые компилируемые модули со своими зависимостями.

Второй задачей был выбор архитектуры. Выбор был не большой, проект достаточно маленький, поэтому был выбран стандарт индустрии – Clean Architecture.

Третей задачей был выбор нужной библиотеки для работы с API того же проекта, но на беке. После недолгих рассуждений и обсуждения с беком, было выяснено, что стандартного набора более простой библиотеки «Retrofit» должно вполне хватить для моих задач.

## 2.2 Второй спринт

Задачи второго спринта были направлены на прикладные вещи для старта проекта. Они отражены в таблице 2.

Таблица 2 — Задачи на 2 спринт

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Приоритет |
| Реализовать базовые экраны приложения | Базовые экраны (Activity) должны иметь свои VM и должны быть указаны в манифесте приложения | Высокий |
| Настроить DI | Так как приложение многомодульное, необходим инструмент управления внутренними зависимостями | Высокий |
| Изучить подходящие библиотеки для работы с MarkDown | Узнать о базовых решениях, а также об опенсурсных | Средний |
| Инициализировать БД (SQLite) | Инициализация БД, написание первых SQL запросов для ее создания и подключения. | Средний |

Первым шагом была реализация основных экранов (Activity), каждый из которых связан со своей ViewModel (VM) для разделения логики и UI. Все Activity были зарегистрированы в AndroidManifest.xml. Для навигации между экранами использовался NavController из Jetpack Navigation.

Поскольку приложение многомодульное, был выбран Dagger — готовое решение для DI от Google. В каждом модуле были прописаны необходимые зависимости, а также созданы общие компоненты для повторного использования.

Были рассмотрены несколько библиотек, включая Markwon (популярное решение для Android) и CommonMark. После тестирования выбрано открытое решение с названием «compose-markdown» .

Для работы с локальной базой данных использовалась SQLite. Написаны первые SQL-запросы для создания таблиц и базовых операций.

## 2.3 Третий спринт

Задачи 3 спринта были направлены на создание CRUD-операций для работы с заметками и интеграция с API. Задачи отражены в таблице 3.

Таблица 3 — Задачи на 3 спринт

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Приоритет |
| Добавить возможность создания и редактирования заметок | Нужно добавить основные CRUD операции | Высокий |
| Написать unit-тесты для основных модулей | Так как библиотеки незнакомые, стоит писать простейшие тесты | Высокий |
| Протестировать работу с локальной базой данных | После появления базового интерфейса имеет смысл написать интеграционные тесты и протестировать БД | Средний |
| Формирование запросов к API (Retrofit, API). | Разработка начального представления о запросах к API, их обработка, подумать над асинхронщиной | Средний |

Реализованы основные функции: создание, чтение, обновление и удаление заметок. Для этого использовались методы SQLite и LiveData для автоматического обновления UI при изменении данных.

Тестирование проводилось с помощью JUnit и Mockito для изолированной проверки ViewModel и репозиториев. Были написаны тесты на корректность работы CRUD-операций и обработку ошибок.

Созданы интеграционные тесты с использованием AndroidX Test, проверяющие корректность сохранения и загрузки данных из Room.

Разработаны базовые API-запросы (GET, POST, PUT, DELETE) с помощью Retrofit. Для асинхронности использовались корутины (Kotlin Coroutines), чтобы избежать блокировки UI.

## 2.4 Четвертый спринт

Задачи второго спринта были направлены на улучшение пользовательского интерфейса: добавления анимаций, разработка дизайн и внедрение поддержки оффлайн-режима. Задачи отражены в таблице 4.

Таблица 4 — Задачи на 4 спринт

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Приоритет |
| Формирование дизайна приложения | Нужно определить тему приложения и представить начальный вариант в любом редакторе | Высокий |
| Создание анимаций | Какой бы дизайн не был хороший, без анимаций он будет выглядеть топорно | Высокий |
| Перенос работы с сетью и репозиториями в отдельные потоки | Асинхронность обязательна в нынешних приложениях, так как от нее может зависеть общее впечатление | Средний |
| Добавить поддержку оффлайн-режима (кеширование данных). | Приложение должно иметь возможность работать в оффлайн режиме, нужно кеширование медиа | Средний |

Дизайн создавался в Figma, учитывая Material Design 3. Определены основные цвета, шрифты и стили компонентов.

Добавлены анимации переходов между экранами (через Navigation Component) и микро-интеракции (нажатия кнопок, загрузка данных) с помощью Lottie и MotionLayout.

Все сетевые операции и работа с БД вынесены в фоновые потоки через корутины. Для обработки состояний загрузки/ошибок использовался StateFlow.

Реализовано кеширование данных через SQLite. При отсутствии интернета приложение показывает последние сохранённые данные, а при восстановлении соединения синхронизирует изменения с сервером.

## 3. Подготовка к релизу

Во время подготовки к релизу были проведены тесты, доработка внешнего вида приложения и выкладывания релизных версий продукта на различные площадки.

## 3.1 Пятый спринт

Проведена оптимизация приложения: добавлены диалоги подтверждения действий, протестирована работа на разных устройствах. Подготовлена релизная сборка с удалением отладочных логов. Задачи описаны в таблице 5.

Таблица 5 — Задачи на 5 спринт

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Приоритет |
| Добавление тулбара | Для работы с текстом необходимо иметь тулбар с базовыми возможностями редактирования | Высокий |
| Протестировать приложение на разных устройствах и версиях Android. | Имеется некоторое количество физических и неограниченное количество виртуальных устройств, нужно попробовать протестировать приложение там | Высокий |
| Создание предупреждений об удалении заметки и отмене изменений | В случае, если пользователь нажал случайно на удаление или хочет выйти без изменений, то стоит предупредить о его действии | Средний |
| Тестирование отмены изменений | Мультимедиа и их состояния могут вести себя некорректно, так как идет работа с кешем | Средний |

Создан кастомный тулбар с кнопками форматирования текста (удаление, сохранение, добавление мультимедиа) для редактирования заметок. Использовался MaterialToolbar с кастомными MenuItem.

Приложение протестировано на эмуляторах (Android Studio) и реальных устройствах (Realme, Samsung) с разными версиями ОС (от Android 8 до 14).

Добавлены AlertDialog перед критическими действиями (удаление, выход без сохранения).

Проверены сценарии отмены изменений в заметках, особенно при работе с медиа (изображения). Учтено поведение кеша и восстановление предыдущего состояния.

## 3.2 Шестой спринт

Основное внимание уделено подготовке к релизу. Создана оптимизированная сборка приложения, проведено тестирование производительности на различных устройствах. Выполнена очистка логов и окончательная проверка стабильности работы перед выпуском. Задачи отображены в таблице 6.

Таблица 6 — Задачи на 6 спринт

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Приоритет |
| Подготовка первой релизной версии | Проверка работы приложения в релизной версии | Высокий |
| Тестирование скорости работы приложения релизной версии | Релизная версия отличается своей скоростью работы, но нужно в ней убедиться | Высокий |
| Чистка логов | Google рекомендует удалять все логи при выпуске приложений | Средний |
| Запуск и проверка релизной версии на разных физических устройствах | Разные устройства имеют разную производительность, стоит проверить приложение на как можно большем количестве девайсов | Средний |

Создан release-билд с оптимизацией R8. Удалены debug-логи и лишние зависимости.

Замеры производительности через Android Profiler показали улучшение скорости на 20-30% по сравнению с debug-сборкой.

Все логи удалены или заменены на Timber с проверкой версии в конфигурации логгера.

Финальное тестирование на физических устройствах (разные производители, версии Android).

## 3.3 Седьмой спринт

Добавлены темы (светлая/тёмная), проведена презентация и бета-тестирование. Собран фидбэк и составлен план дальнейшего развития приложения. Задачи описаны в таблице 7.

Таблица 7 — Задачи на 7 спринт

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Приоритет |
| Добавление разных тем приложения | Разработка светлой и темной темы | Высокий |
| Подготовка к презентации приложения | Презентация приложения будет происходить на лекции | Высокий |
| Пиар и бета тест | Распространение приложения среди студентов | Высокий |
| Подведение итогов и планы на будущее | Составление ТЗ на будущее и создание TODO листов | Средний |

Реализована поддержка тем через специальный инструментарий в Android. Добавлены настройки переключения в самом приложении.

Создана короткая демонстрация ключевых функций (3-5 минут). Подготовлены скриншоты и презентация.

Приложение выложено на удаленном репозитории и телеграмм группы команды разработки (бека, фронта и мобильной команд). Собраны первые отзывы.

Cоставлен список доработок:

* добавление синхронизации через Firebase;
* расширение MarkDown-редактора;
* улучшение анимаций.

## 4**. Инструкция использования**

Для использования приложения его нужно установить на любой смартфон, поддерживающий версию Android выше седьмой. Скачать его можно с моего удаленного репозитория по ссылке <https://github.com/solutions-architects/anynote_android/releases/tag/boevoy>.

**ЗАКЛЮЧЕНИЕ**

В результате выполнения всех спринтов было успешно разработано и подготовлено к релизу приложение для работы с заметками. На начальных этапах была создана устойчивая архитектура, реализованы базовые функции CRUD и интеграция с API. Особое внимание уделялось качеству кода - написаны unit-тесты, проведено интеграционное тестирование, что позволило выявить и исправить потенциальные проблемы на ранних стадиях. В процессе разработки постоянно совершенствовался пользовательский интерфейс: добавлены анимации, поддержка темной темы и оффлайн-режим, что значительно улучшило UX. Финальные спринты были посвящены оптимизации производительности, тщательному тестированию на различных устройствах и подготовке релизной версии. В результате получилось стабильное, производительное приложение с продуманным интерфейсом, готовое к публикации.. Весь процесс разработки показал важность поэтапного подхода, постоянного тестирования и внимания к деталям для создания качественного продукта.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1) Layout basics <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-basics> (Дата обращения: 15.02.2025)

2) Material Components [https://developer.android.com/design/ui/mobile/guides/components/material-overview](https://developer.android.com/design/ui/mobile/guides/components/material-overview%20) (Дата обращения: 15.02.2025)

3) Sharing a file <https://developer.android.com/training/secure-file-sharing/share-file#java> (Дата обращения: 16.02.2025)

4) Device compatibility overview <https://developer.android.com/guide/practices/compatibility> (Дата обращения: 2.03.2025)

5) Screen compatibility overview [https://developer.android.com/guide/practices/screens\_support](https://developer.android.com/guide/practices/screens_support%20%20)  (Дата обращения: 2.03.2025)

6) Developing for older Android version <https://stackoverflow.com/questions/56572238/developing-for-older-android-version> (Дата обращения: 2.03.2025)

7) Develop a UI with Views <https://developer.android.com/studio/write/layout-editor> (Дата обращения: 14.03.2025)

8) How to delete internal storage file in android? <https://stackoverflow.com/questions/3554722/how-to-delete-internal-storage-file-in-android> (Дата обращения: 14.03.2025)

9) Android Debug Bridge (adb) <https://developer.android.com/tools/adb> (Дата обращения: 12.04.2025)

10) sqlite3 <https://developer.android.com/tools/sqlite3> (Дата обращения: 12.04.2025)

11) compose-markdown [https://github.com/jeziellago/compose-markdown](https://github.com/jeziellago/compose-markdown%20) (Дата обращения: 12.04.2024)

12) dagger [https://github.com/google/dagger](https://github.com/google/dagger%20) (Дата обращения: 15.04.2024)

13) Kotlin for Jetpack Compose <https://developer.android.com/develop/ui/compose/kotlin> (Дата обращения: 18.04.2024)

14) Quick guide to Animations in Compose <https://developer.android.com/develop/ui/compose/animation/quick-guide> (Дата обращения: 1.05.2024)

15) Navigation with Compose<https://developer.android.com/develop/ui/compose/navigation> (Дата обращения: 2.05.2024)

## ПРИЛОЖЕНИЕ А

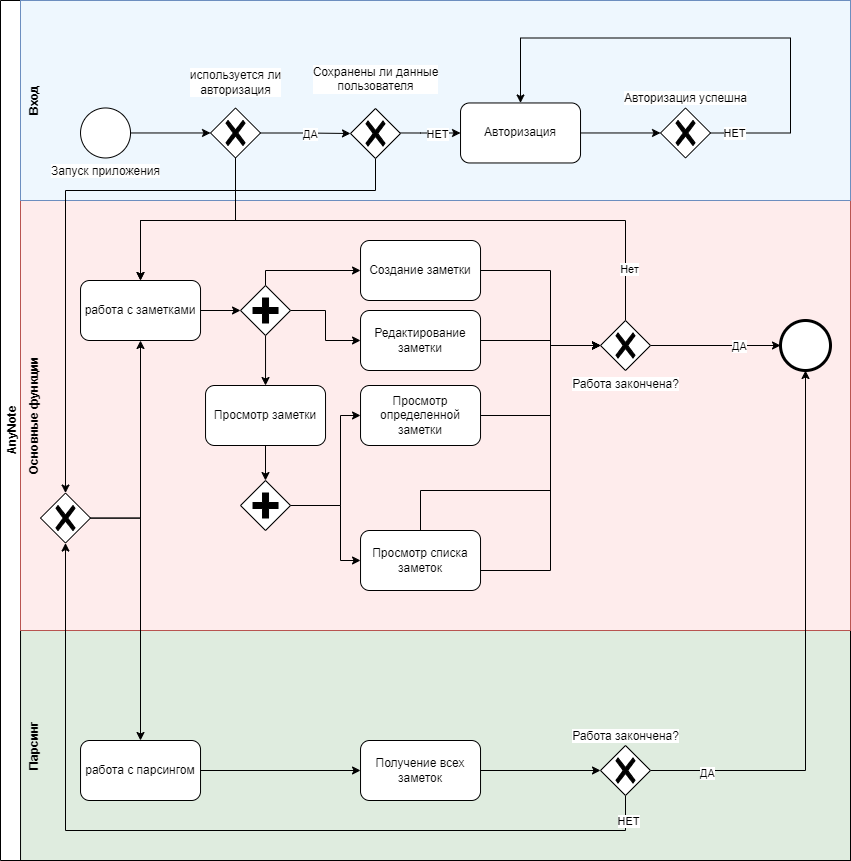


Рисунок 2 – Диаграмма Userflow

## ПРИЛОЖЕНИЕ Б

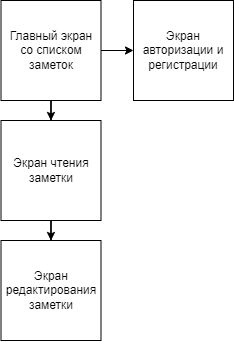


Рисунок 3 – Диаграмма окон

## ПРИЛОЖЕНИЕ В

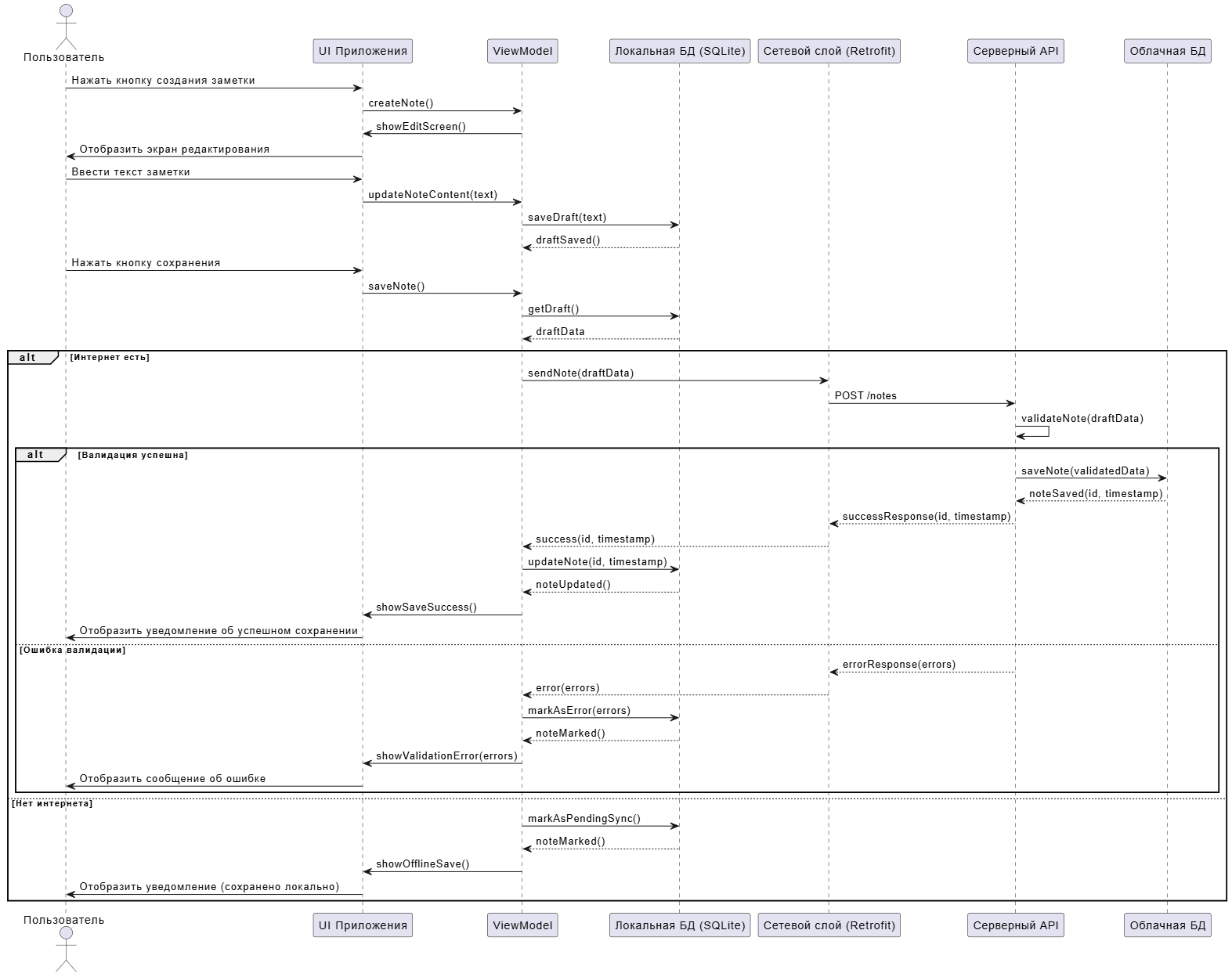


Рисунок 4 – Диаграмма последовательности