Medicine organizer

Архитектурный документ

Авторы документа:

Дрозд Софья Александровна (БПИ215)

Преподаватель группы:

Поваляева Елизавета Максимовна

Учебный ассистент:

Глазков Максим Сергеевич

0. Раздел регистрации изменений

Версия документа	Дата изменения	Описание изменения	Автор изменения
1.0.0	29.02.2024	Первоначальная версия документа	Дрозд Софья Александровна

1. Введение

1.1. Название проекта

Наименование программы на русском языке: «Сервис-помощник “Органайзер лекарств”».
Наименование программы на английском языке: «Assistant “Medicine organizer”».

1.2. Задействованные архитектурные представления

Для описания проекта были использованы следующие представления: прецедентов, логическое, архитектуры процессов, развертывания.

1.3. Контекст задачи и среда функционирования системы

«Сервис-помощник “Органайзер лекарств”» – это Android-приложение, созданное для автоматизации ведения учета купленных лекарственных препаратов, отслеживания их сроков годности, а также напоминаний о приёме лекарств. Пользователь может добавлять купленные медикаменты в базу данных и просматривать информацию обо всех лекарствах. Область применения – сфера здравоохранения. Приложение предназначено для личного пользования в качестве программы для организации домашней, автомобильной или других видов аптечек.

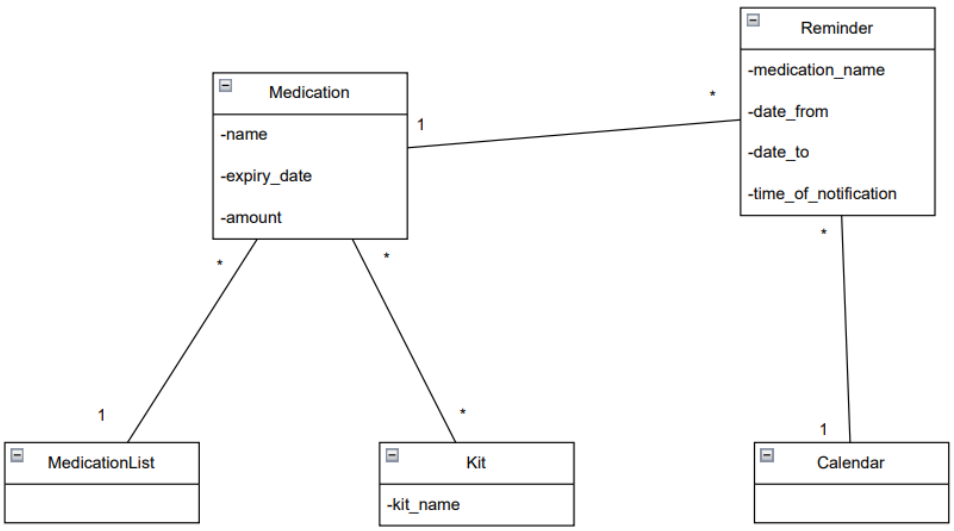

Рисунок 1 - Модель предметной области

1.4. Рамки и цели проекта

Проект "Сервис-помощник 'Органайзер лекарств'" включает в себя разработку Android-приложения для учета и управления лекарствами. Он включает функции добавления, отслеживания и управления информацией о лекарствах, а также предоставляет функционал напоминаний о приеме препаратов. Основная цель разработки заключается в создании удобного инструмента для пользователей, помогающего им эффективно управлять своими лекарственными препаратами и обеспечивающего своевременный прием медикаментов.

2. Архитектурные факторы

2.1. Ключевые заинтересованные лица

Действующее лицо	**Заинтересованность в системе**
Разработчик	Простота поддержки и введения новой функциональности
Архитектор	Удобство использования и доработки архитектурных документов

2.2. Ключевые требования к системе

Ключевые требования к системе:

- Реализована работа с медикаментами
    - Система должна сохранять данные о наименовании, сроке годности и количестве каждого медикамента.
    - Пользователь должен иметь возможность просматривать список всех лекарств, добавленных в базу данных.
    - Система должна позволять пользователю сортировать список медикаментов по сроку годности и по алфавиту.
    - Пользователь должен иметь возможность удалять и обновлять информацию о лекарственных препаратах.
- Реализована работа с аптечками
    - Система должна поддерживать функционал создания пользовательских категорий (аптечек) для организации лекарств.
- Реализована работа с напоминаниями о приеме лекарств
    - Система должна предоставлять пользователю календарь с отображением напоминаний о приеме лекарств.
    - Пользователь должен иметь возможность добавлять и удалять напоминания о приеме лекарств с указанием периода и времени напоминания.
    - Система должна отправлять push-уведомления пользователю для напоминания о приеме лекарств в установленные им время и даты.

2.3. Ключевые ограничения

Приложение должно быть разработано для платформы Android. Поскольку приложение предназначено для мобильных устройств, необходимо учитывать ограничения по производительности, размеру экрана и другим характеристикам мобильных устройств.

Необходимо обеспечить высокий уровень безопасности и защиты конфиденциальности данных, так как приложение работает с медицинскими данными пользователей. Приложение должно соответствовать законодательным требованиям о здравоохранении и защите персональных данных.

3. Общее архитектурное решение

Приложение состоит их трех основных смысловых разделов. Первый отвечает за функциональность, связанную с медикаментами, второй - за аптечки, а третий - за напоминания и их отображение на календаре. Каждый модуль реализует определенную функциональность и взаимодействует с базой данных приложения для хранения и получения необходимой информации.

Основная архитектура приложения строится на принципах модульности и разделения ответственности, что обеспечивает легкость поддержки и масштабирования приложения. Также предусмотрен механизм отправки push-уведомлений пользователю для напоминания о приеме лекарств.

Интерфейс пользователя разрабатывается с учетом удобства использования и интуитивной навигации.

4. Архитектурные представления

4.1. Представление прецедентов

Подробная модель прецедентов представлена на Рис. 2.

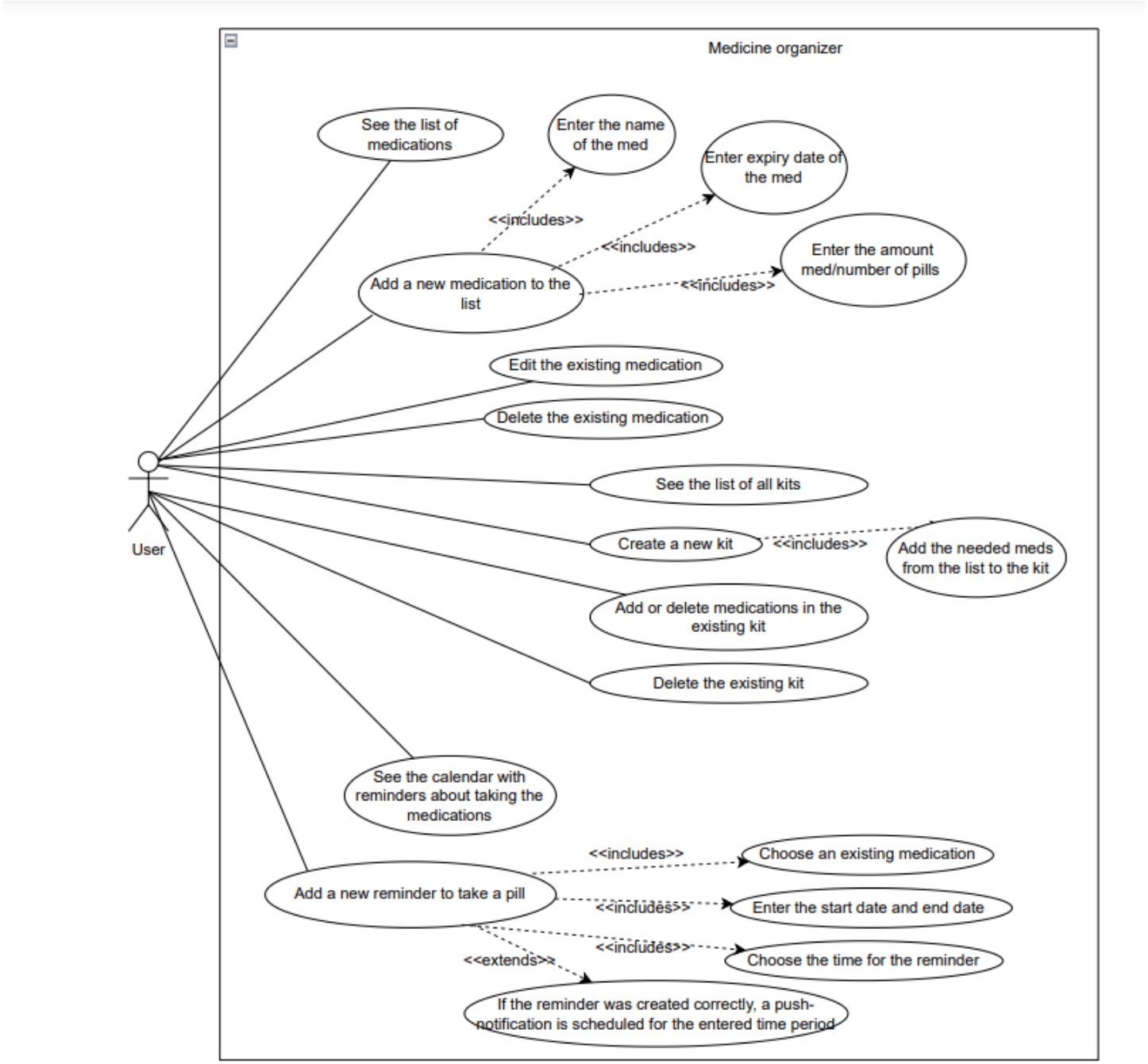

Рисунок 2 - Модель прецедентов

4.2. Логическое представление

Приложение разделено на три основных пакета: med, kit и calendar. Пакет "med" отвечает за функциональность, связанную с медикаментами, "kit" - за аптечки, а "calendar" - за напоминания и их отображение на календаре.

На диаграммах (Рис. 3-7) представлены основные пакеты и классы приложения, а также их связь и взаимодействия.

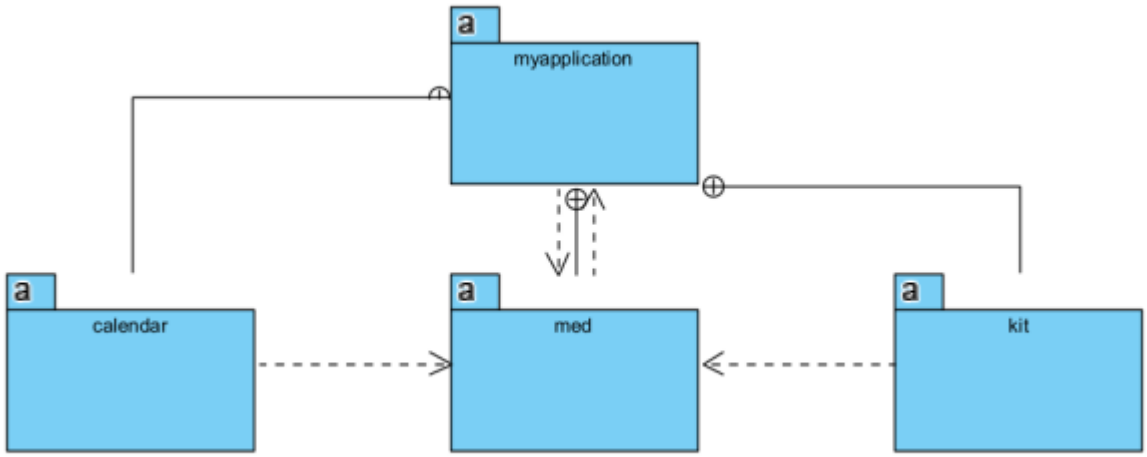

Рисунок 3 - Диаграмма пакетов

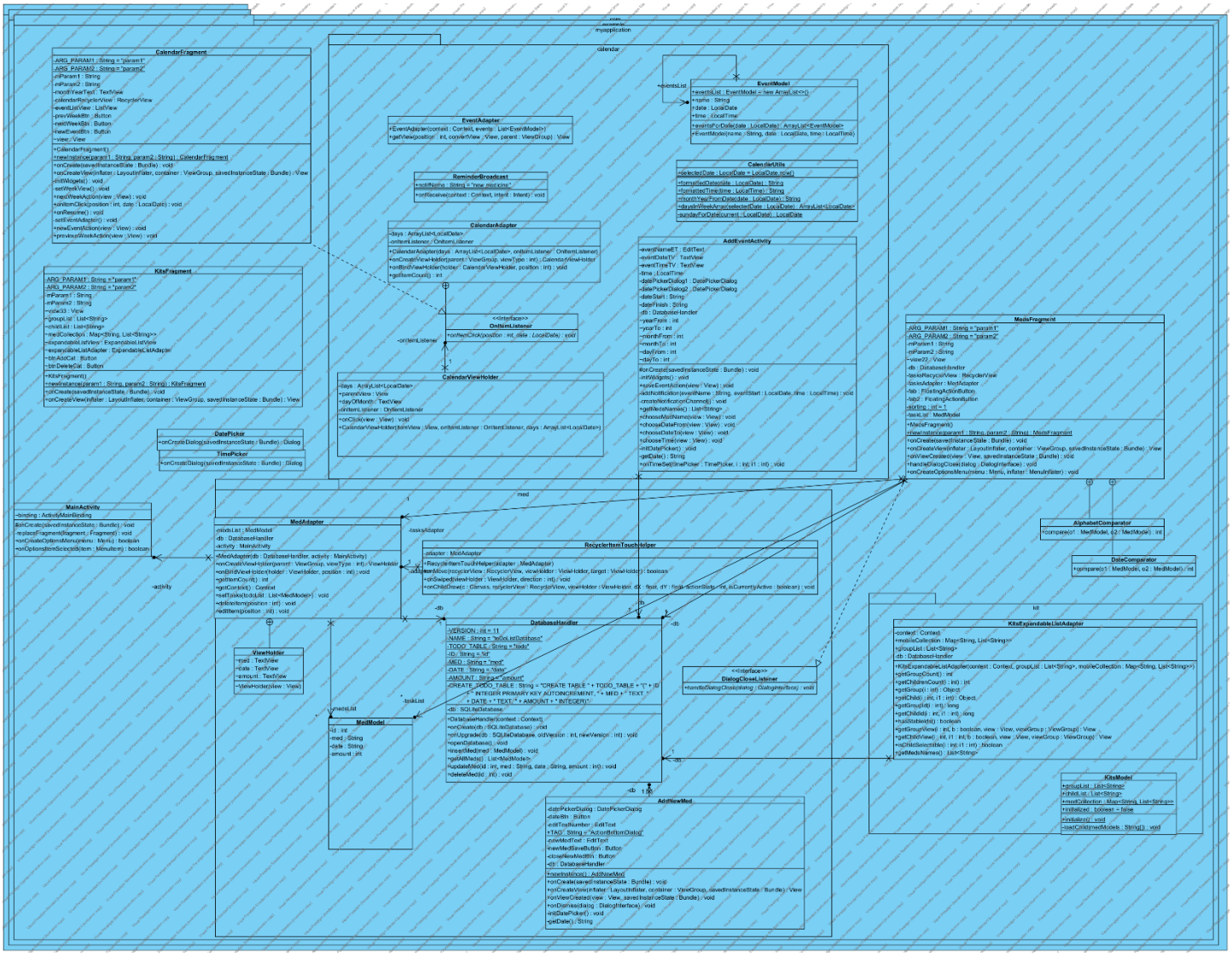

Рисунок 4 - Диаграмма классов

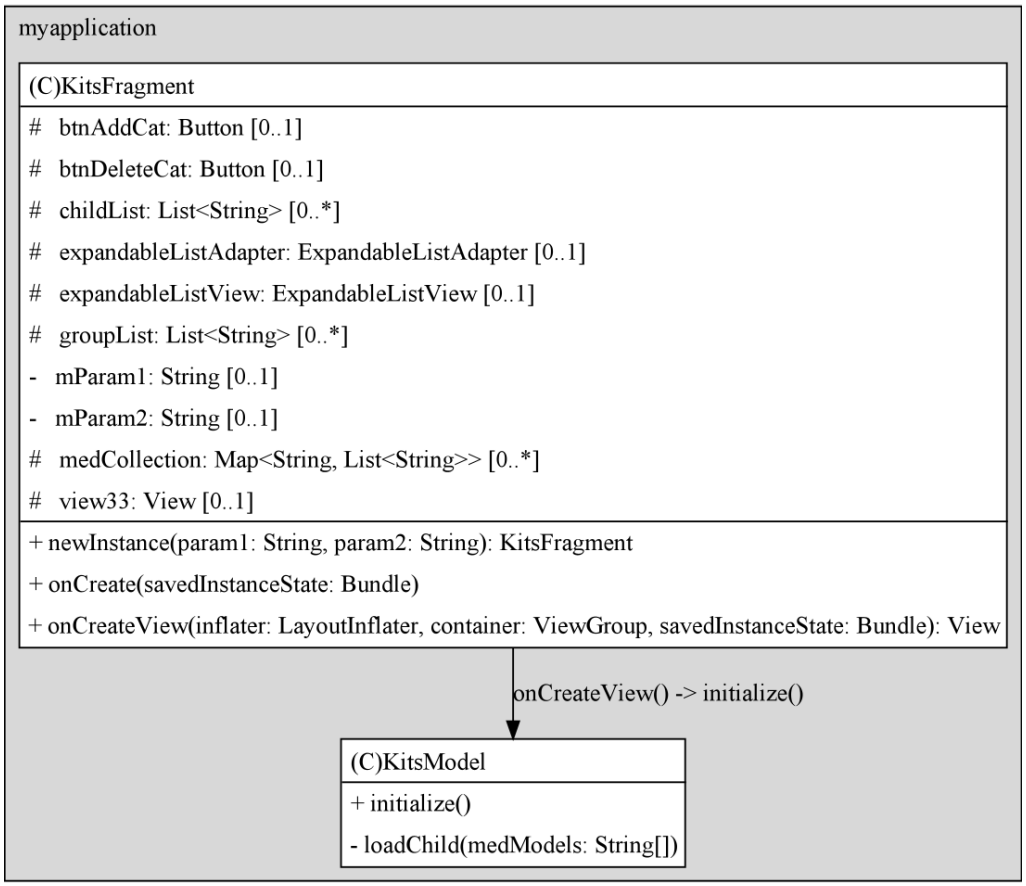

Рисунок 5 - Структурная диаграмма (пакет "kit")

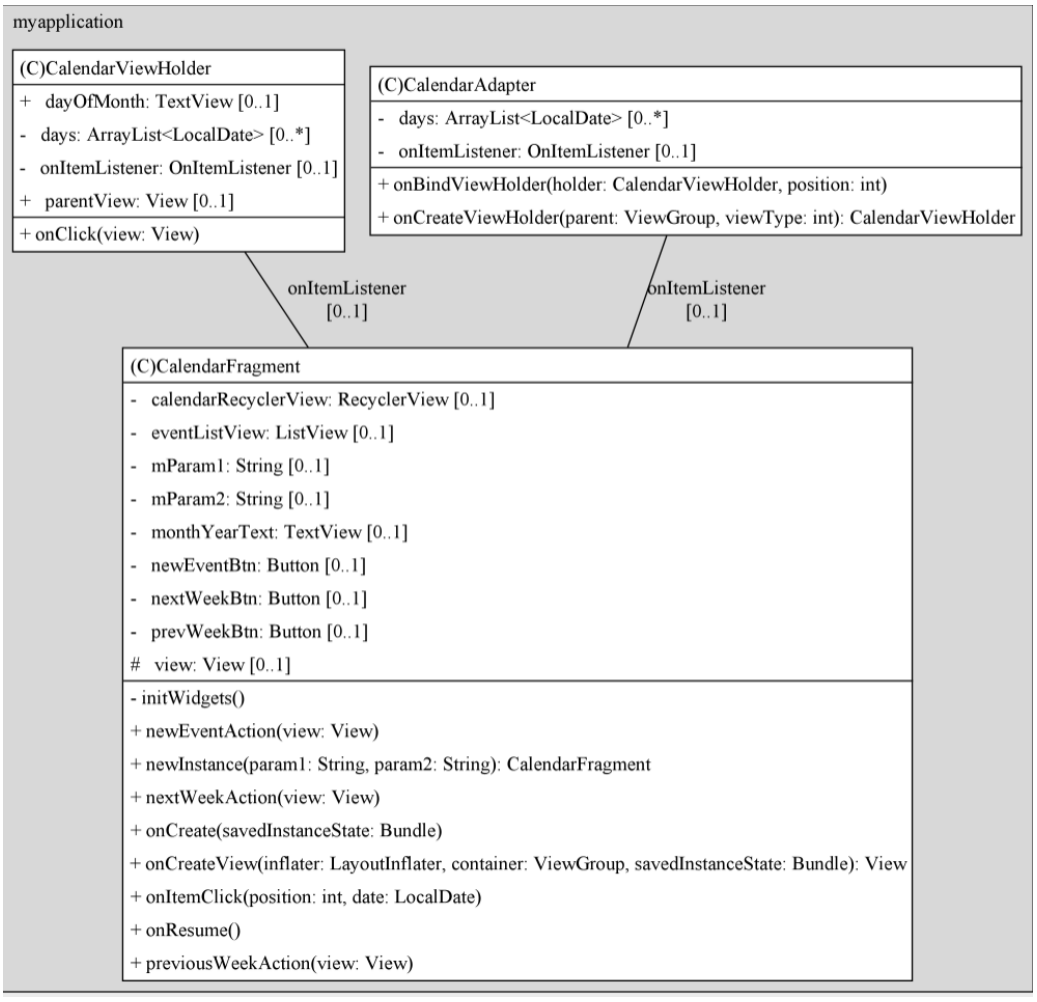

Рисунок 6 - Структурная диаграмма (пакет "calendar")

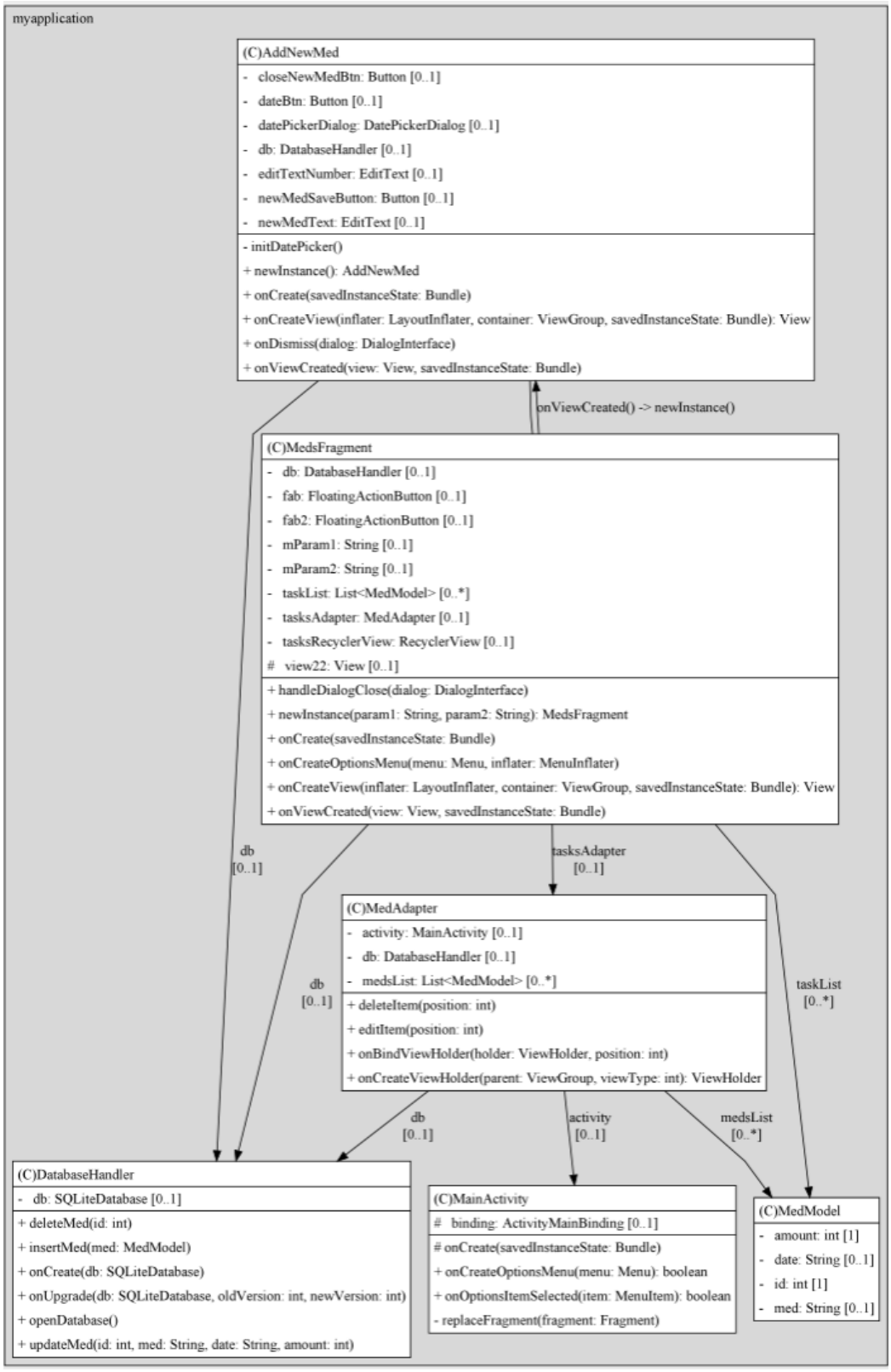

Рисунок 7 - Структурная диаграмма (пакет "med")

4.3. Представление архитектуры процессов

Диаграмма, описывающая основные процессы в приложении «Medicine organizer», представлена на Рис. 8.

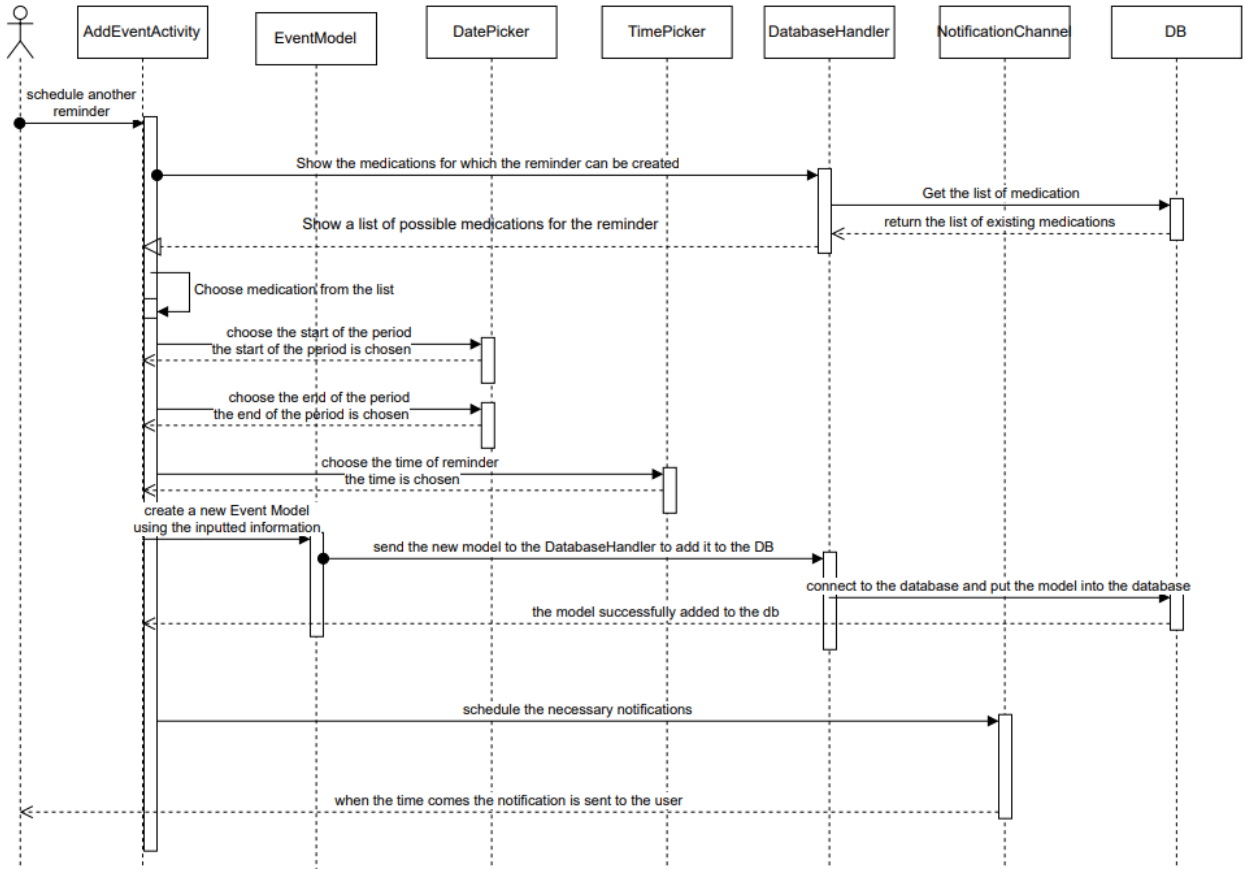

Рисунок 8 - Диаграмма последовательности

4.4. Физическое представление архитектуры

Для работы приложения необходимы планшетный компьютер, смартфон или другое устройство, на котором установлена Android версии 9.0 (Pie) и выше. Технические требования также включают наличие API версии 28 и выше, что обеспечивает совместимость с современными версиями Android и расширенные возможности разработки. Для надлежащей работы механизма push-уведомлений о приеме лекарств необходимо, чтобы устройство имело разрешение на отправку и получение уведомлений. Это обеспечивает своевременное уведомление пользователя о необходимости принятия медицинских препаратов. Кроме того, для установки и корректной работы приложения требуется наличие не менее 200 Мб свободного пространства на устройстве.

4.5. Представление развертывания

Приложение будет доступно для установки на мобильные устройства с операционной системой Android (версия 9.0 и выше). Компоненты и модули приложения будут упакованы в виде APK файлов.

Для обновления системы разработчики будут выпускать новые версии приложения в магазине Google Play. Пользователи будут получать уведомления о наличии обновлений, и их устройства будут автоматически загружать и устанавливать новые версии приложения.

4.6. Представление архитектуры данных

Система будет работать с различными данными, включая информацию о лекарствах, их сроках годности, дозировках, времени приема и других связанных параметрах. Для хранения данных будет использоваться локальная база данных SQLite.

Для интеграции данных о лекарствах система может использовать API онлайн-аптек или базы данных лекарственных препаратов.

Потоки данных будут включать операции добавления, обновления, удаления и чтения данных о лекарствах, а также управление пользовательскими настройками и напоминаниями (push-уведомлениями).

4.7. Представление архитектуры безопасности

Для обеспечения безопасности и сохранности данных в системе будут применяться следующие меры:

- Реализация механизмов аутентификации и авторизации пользователей для доступа к приложению и его функционалу.
- Шифрование чувствительных данных, таких как личная информация пользователей и медицинские данные.
- Защита от угроз безопасности, таких как вредоносные программы и атаки на приложение.
- Регулярные аудиты безопасности и обновления приложения для исправления выявленных уязвимостей.
- Резервное копирование данных для предотвращения и восстановления от потенциальных потерь данных в случае аварии или взлома.

4.8. Представление реализации и разработки

Программа должна быть разработана на языке программирования Java. Проектирование шаблонов интерфейса разрабатывается на XML. В качестве среды разработки должна быть использована среда Android Studio.
Хранение данных о лекарственных препаратах должно быть реализовано в базе данных SQLite.

4.10. Атрибуты качества системы

Нефункциональные аспекты проекта включают в себя характеристики, не связанные напрямую с его функциональностью, но важные для эффективной работы:

1. Производительность: система гарантирует отсутствие задержек для пользователей.

2. Надежность: система обеспечивает сохранность данных и надежную работу.

3. Масштабируемость: система может масштабироваться под рост пользователей и объема данных.

4. Безопасность: система защищает данные от несанкционированного доступа и использования.

5. Удобство использования: система обеспечивает простоту и интуитивно понятный интерфейс.

6. Поддерживаемость: система обеспечивает легкость восстановления и обновления.

7. Совместимость: гарантируется совместимость с другими системами для корректного взаимодействия.

4.10.1. Объем данных и производительность системы

Система должна эффективно обрабатывать информацию о лекарствах, их сроках годности и напоминаниях о приеме. В зависимости от количества пользователей и активности использования приложения, объем данных может значительно варьироваться. Система должна обеспечивать быстрый доступ к данным, быструю загрузку и отображение информации о лекарствах, а также мгновенную реакцию на запросы пользователя, особенно в контексте напоминаний о приеме лекарств.

4.10.2. Гарантии качества работы системы

Работоспособность системы можно проверить с помощью тестирования производительности, функционального тестирования, а также сценариев восстановления после сбоев. Регулярные тесты и мониторинг системы помогут выявить проблемы и предотвратить возможные отказы. В случае обнаружения каких-либо проблем можно обратиться к разработчикам, сообщив о найденной проблеме на платформе GitHub.

4.11. Другие представления

Визуальное представление программы в виде UI mockup представлено на Рис. 9.

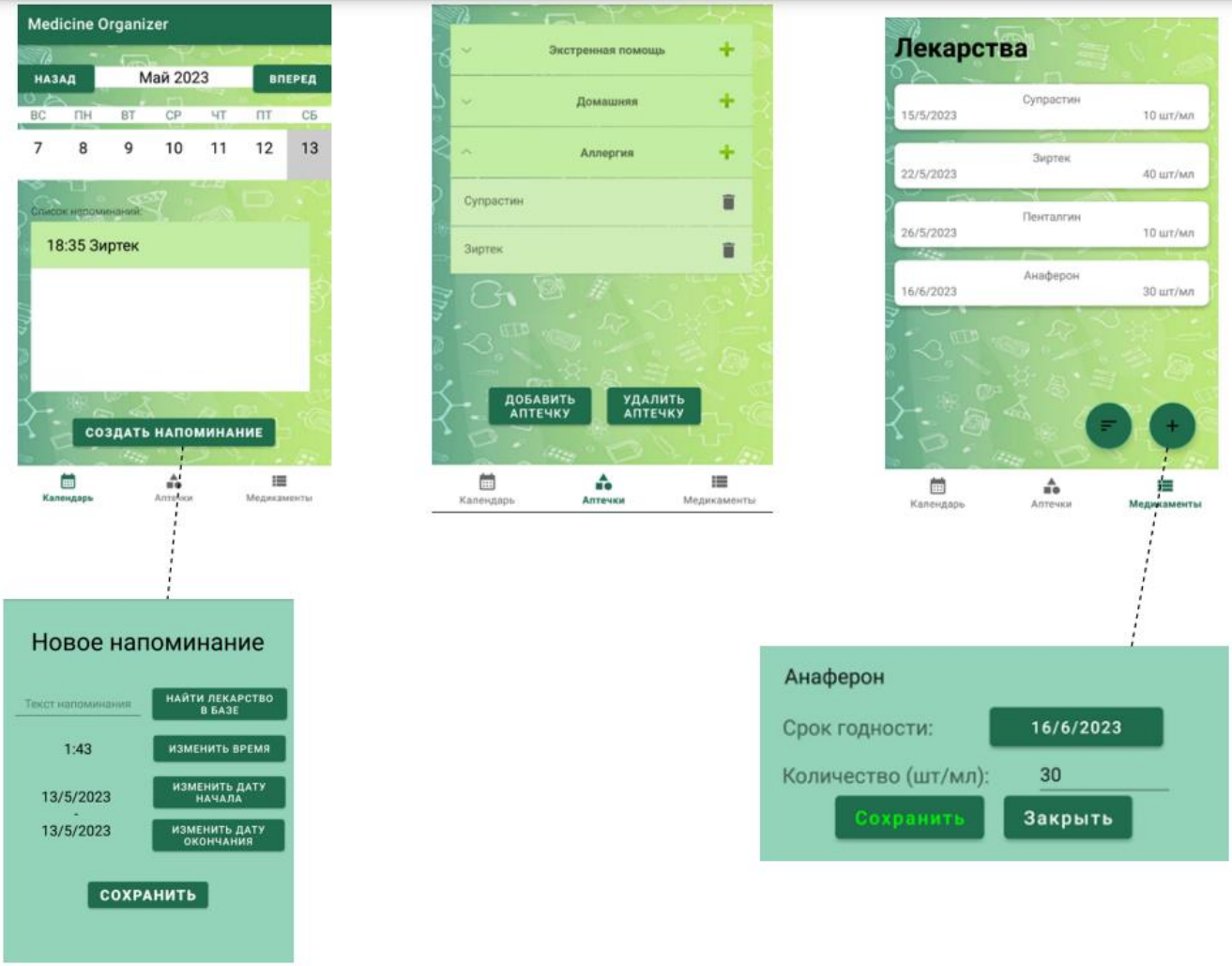

Рисунок 9 - UI mockup приложения

5. Технические описания отдельных ключевых архитектурных решений

5.1. Техническое описание решения №1: Хранение информации в системе

5.1.1. Проблема

Как обеспечить надежное и оптимальное хранение информации в системе?

5.1.2. Идея решения

Использование базы данных SQLite для хранения данных о лекарствах и напоминаниях.

5.1.3. Факторы

Требования ТЗ позволяют использовать базу данных SQLite из-за ее эффективности и интеграции с Android. Данные сохраняются надежно при корректной реализации.

5.1.4. Решение

Создание базы данных SQLite с таблицами для медикаментов, аптечек и календаря. Использование SQL запросов для вставки, выборки и обновления данных.

5.1.5. Мотивировка

SQLite - легковесная и быстрая база данных, хорошо подходящая для мобильных приложений. Ее интеграция с Android SDK делает использование ее предпочтительным.

5.1.6. Неразрешенные вопросы

Нет

5.1.7. Альтернативы

Рассматривались также реляционные базы данных, такие как MySQL или PostgreSQL, но они требуют больше ресурсов и сложнее в использовании на мобильных устройствах. Другой альтернативой может быть использование NoSQL решений, но они могут быть излишними для данного проекта.

5.2. Техническое описание решения №2: Обеспечение оптимальной производительности

5.2.1. Проблема

Как обеспечить оптимальную производительность системы при работе с базой данных лекарств?

5.2.2. Идея решения

Использование индексов в базе данных для оптимизации запросов и ускорения доступа к данным.

5.2.3. Факторы

Требования к производительности приложения позволяют использовать индексы для оптимизации работы с базой данных. Это особенно важно при обработке больших объемов данных о лекарствах и их сроках годности.

5.2.4. Решение

Создание индексов на полях, по которым часто выполняются запросы, таких как название лекарства, срок годности и дата приобретения. Это позволит ускорить поиск и фильтрацию данных в базе данных.

5.2.5. Мотивировка

Использование индексов позволяет сократить время выполнения запросов к базе данных, что повышает отзывчивость приложения и улучшает пользовательский опыт.

5.2.6. Неразрешенные вопросы

Нет

5.2.7. Альтернативы

Рассматривались также технологии кэширования данных для сокращения времени доступа к информации, однако в данном контексте использование индексов в базе данных оказалось более эффективным и простым в реализации.

6. Приложения

6.1. Словарь терминов

1. API (Application Programming Interface) - это набор инструкций и стандартов, которые определяют, как различные компоненты программного обеспечения могут взаимодействовать друг с другом. Обычно API используется для создания приложений, которые могут использовать функциональность или данные других приложений или служб.
2. APK файл - это файл формата, используемый в операционной системе Android для установки и распространения приложений. APK (Android Package) содержит все необходимые файлы и ресурсы, необходимые для установки и запуска приложения на устройстве Android.
3. Google Play - это официальный магазин приложений для устройств Android, предоставляемый компанией Google. В Google Play пользователи могут скачивать и устанавливать приложения, игры, фильмы, музыку и другие контенты на свои устройства Android.
4. Android SDK (Software Development Kit) - это набор инструментов и библиотек, предоставляемых Google для разработки приложений под операционную систему Android. Android SDK включает в себя различные инструменты, компоненты и документацию, необходимые для создания и тестирования приложений для устройств Android.
5. SQLite - это компактная реляционная база данных, которая широко используется в приложениях для операционной системы Android. SQLite обеспечивает простой и эффективный способ хранения и управления данными приложения на устройствах Android.
6. UI mockup - это набор изображений или дизайн-макетов, которые используются для визуализации пользовательского интерфейса (UI) приложения или веб-сайта. Mockup помогает разработчикам и дизайнерам лучше понять структуру и внешний вид интерфейса, прежде чем приступать к его реализации.