

Часть 1

1. Тип приложения – мобильное приложение с типом насыщенный клиент (бизнес-слой и слой сервиса данных располагается на самом устройстве.)
2. Стратегию развёртывания – нераспределенное развёртывание. То есть логика представления и логика-бизнеса размещаются непосредственно на клиенте



Рис.1.1

3. Так как при нераспределенном развёртывании вся функциональность и слои приложения, располагаются на одном сервере, как показано на рис. 1.1. Преимуществом данного подхода является простота и минимальные требования по количеству необходимых физических серверов. Также обеспечивается наилучшая производительность, поскольку взаимодействие между слоями осуществляется без пересечения физических границ между серверами или кластерами серверов. Но нельзя забывать, что использование одноуровневой архитектуры снижает общую масштабируемость и удобство обслуживания приложения, поскольку все слои физически располагаются на одном оборудовании.
4. Качества времени выполнения (надежность, безопасность), качества дизайна (удобство и простота обслуживания), производительность, качества взаимодействия с пользователем.
5. Пути реализации сквозной функциональности состоят из следующих областей:
 - 5.1 Безопасность. Проектирование соответствующей стратегии распространения исключений. В дизайне должны быть предусмотрены необрабатываемые исключения. Так же надо перехватывать внутренние исключения (только если можно их обработать). Приложение не должно оставаться в нестабильном состоянии после сбоя. Надо выработать соответствующую стратегию протоколирования и уведомления для критических ошибок и исключений, обеспечивая сохранение достаточно детальных сведений об исключении.
 - 5.2 Управление состоянием – это вопросы, связанные с хранением данных, представляющих состояние компонента, операции или этапа процесса. Сохраняться должны только необходимые данные.
 - 5.3 Резервное копирование и восстановление: приложение может предлагать функцию резервного копирования данных, чтобы пользователь мог восстановить свои заметки.
 - 5.4 Аутентификация и управление учетными записями (пользователи должны иметь возможность создавать учетные записи, входить в систему и управлять своими учетными записями).
 - 5.5 Обмен данными: приложение может предоставлять возможность экспортировать заметки в другие форматы (например, текстовый файл или PDF) или импортировать заметки из других источников.
6. Структурные схемы (диаграмма классов UML (рис. 1.2), типовая структура мобильного приложения (рис. 1.3)):

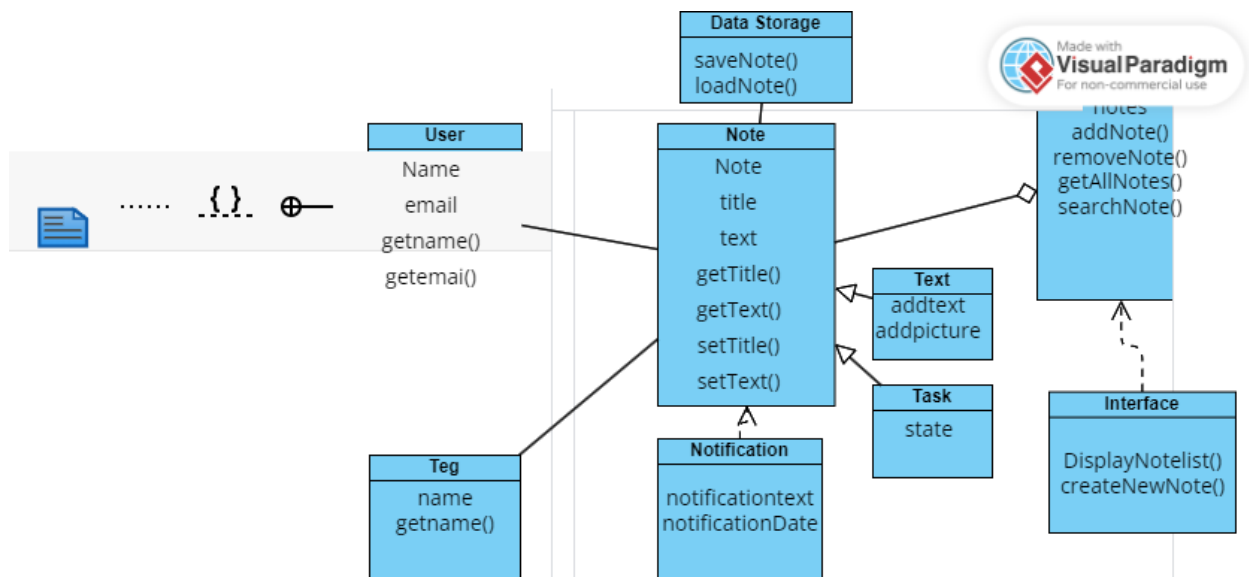


Рисунок 1.2 – диаграмма классов to be

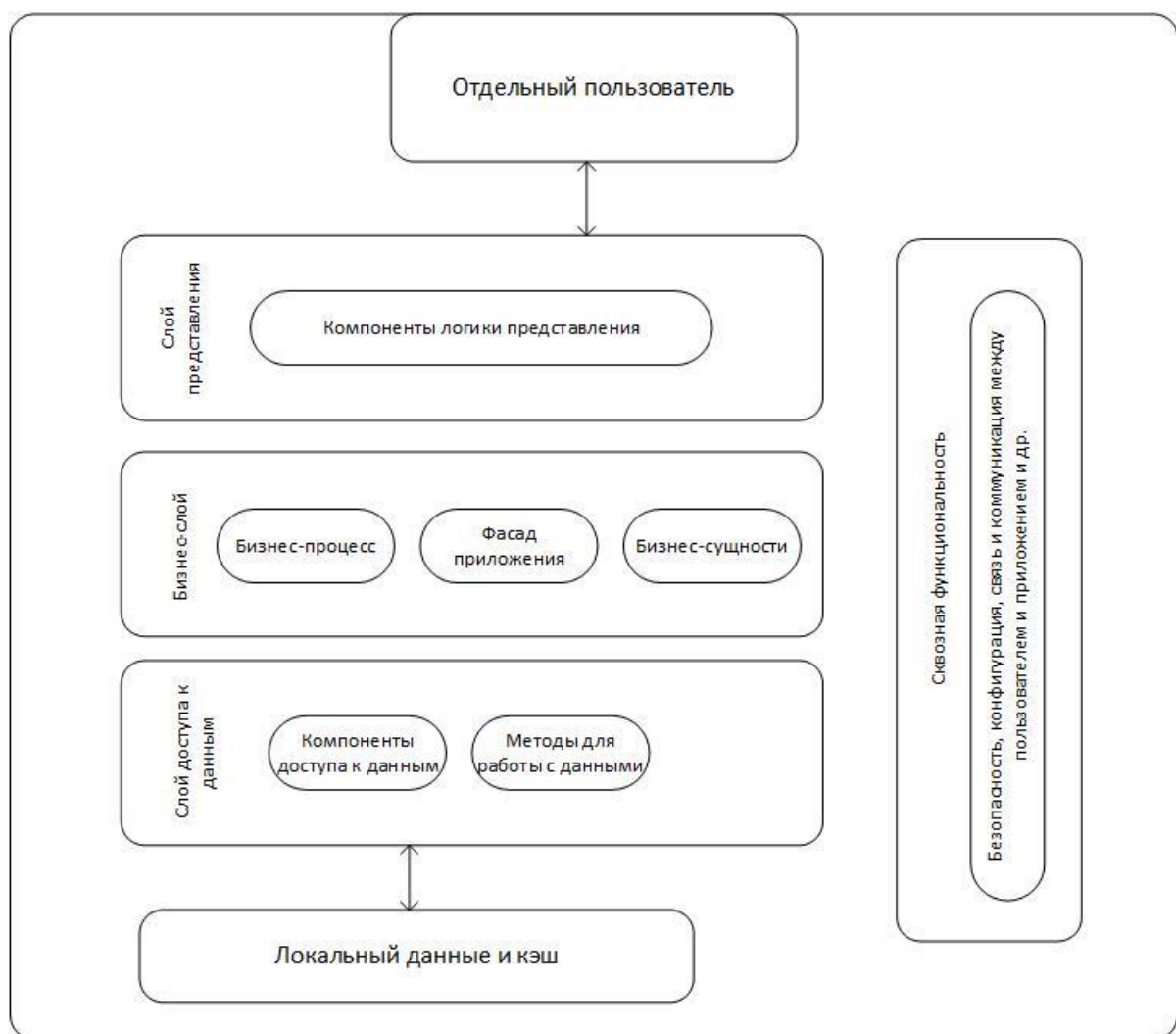


Рисунок 1.3 – Типовая структура мобильного приложения

Часть 2

Используя плагин Generator UML, была сгенерирована диаграмма классов (рис. 1.4)

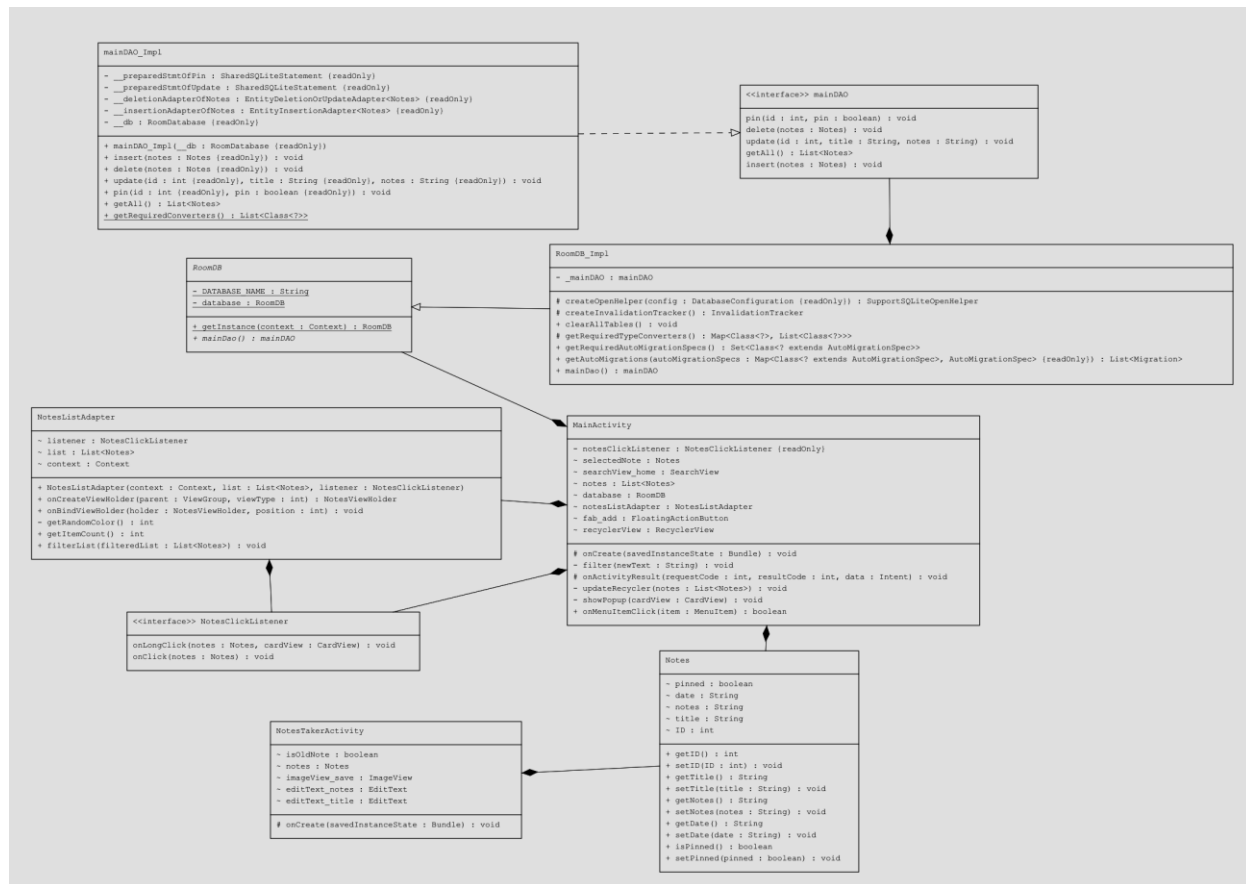


Рисунок 1.4 – диаграмма классов As is

Часть 3

Основные отличия двух архитектур связаны с незаконченной реализацией первой архитектуры.