

# **План демонстрация разработанного программного продукта**

## **Введение и актуальность**

### **Текст для выступления:**

Уважаемый председатель, уважаемые члены государственной экзаменационной комиссии! Вашему вниманию представляются результаты разработки мобильного приложения для гибкого планирования и приоритизации задач.

Актуальность работы обусловлена необходимостью создания инструмента, обеспечивающего бесперебойный доступ к данным в условиях нестабильного интернет-соединения, а также поддерживающего функции совместной работы. Целью работы являлось проектирование и реализация клиент-серверной системы с применением архитектурного паттерна Offline-First.

## **Технологический стек и архитектура**

### **Текст для выступления:**

В качестве основного инструмента разработки клиентской части был выбран фреймворк Flutter. Роль локального хранилища, выступающего единственным источником истины (Single Source of Truth) для пользовательского интерфейса, выполняет высокопроизводительная NoSQL база данных Isar. Серверная инфраструктура реализована на базе платформы Firebase.

Особое внимание в архитектуре уделено механизму синхронизации. В приложении реализована система Queue-Based Sync (синхронизация на основе очереди). Любые изменения данных первоначально фиксируются в

локальной базе и записываются в специализированную таблицу-очередь SyncAction. Фоновый процесс отслеживает состояние сети и при появлении связи асинхронно передает накопленные операции на сервер. Данный подход гарантирует сохранность данных и отсутствие блокировок интерфейса при потере соединения.

### **Демонстрация: Личное планирование (Действия на экране)**

Сценарий показа: Открыть главный экран, показать список задач, переключить статус одной задачи, открыть форму создания.

### **Текст для выступления:**

Перейдем к демонстрации функционала. На главном экране реализован интерфейс личного планирования. Пользователь имеет возможность управлять задачами, назначать им приоритет, категорию и настраивать правила повторения. Поддерживается декомпозиция сложных задач посредством создания чек-листов (подзадач).

Пользовательский интерфейс спроектирован с учетом правила распределения цветов 60-30-10, что обеспечивает визуальную иерархию и снижает когнитивную нагрузку при работе со списками.

### **Демонстрация: Совместная работа и синхронизация**

Сценарий показа: Открыть вкладку «Совместные задачи». Открыть задачу. Показать кнопку генерации кода. Нажать «Присоединиться» и показать форму ввода. Написать тестовый комментарий.

### **Текст для выступления:**

Ключевым модулем системы является подсистема совместной работы. Задачи разделены на изолированные пространства: личные и общие.

Для организации совместного доступа реализован механизм приглашений по уникальному идентификатору (коду). При вводе кода приложение обращается к глобальной коллекции в облачной базе данных, регистрирует пользователя в массиве участников и загружает данные в локальное хранилище.

Внутри совместной задачи реализован чат для обмена комментариями в реальном времени. Все изменения, включая статусы подзадач и новые комментарии, мгновенно синхронизируются между всеми участниками посредством реактивных потоков данных, сохраняя при этом заявленную оффлайн-устойчивость в случае обрыва связи.

## **Заключение**

### **Текст для выступления:**

Таким образом, в ходе выполнения работы спроектировано и успешно реализовано кроссплатформенное мобильное приложение, полностью отвечающее поставленным требованиям. Примененные архитектурные решения, такие как Offline-First и асинхронная очередь синхронизации, обеспечивают высокую отказоустойчивость системы. Поставленные цель и задачи выполнены в полном объеме. Доклад окончен, спасибо за внимание. Готов ответить на ваши вопросы.