

Глава 1

Введение

Собеседования по проектированию мобильных систем (mobile system design, MSD)¹ во многих компаниях стали ключевой частью процесса найма на инженерные должности. Такие собеседования, как правило, короткие — длятся не более часа, но при этом наполнены общими, не имеющими однозначного решения задачами: вас попросят спроектировать Facebook, YouTube или библиотеку пагинации. В условиях ограниченного времени и высоких ожиданий соискателю крайне важно задавать эксперту правильные вопросы, чтобы прояснить требования и сосредоточиться на наиболее важных аспектах.

Эта книга станет практическим помощником для всех: для инженера по мобильной разработке, готовящегося к собеседованию; для техлида², желающего отточить архитектурное мышление; для любого, кто интересуется разработкой мобильных систем. Благодаря реальным примерам из практики, универсальным фреймворкам³ и стратегиям внедрения вы научитесь четко и уверенно решать сложные проектные задачи.



¹ В этой книге вы часто будете встречать термин «дизайн». В данном контексте он, как правило, относится к проектированию системы и ее архитектуры, а не к визуальному оформлению интерфейсов. — *Примеч. науч. ред.*

² Техлид (от англ. «technical lead», «tech lead») — это ведущий технический специалист команды разработки, который отвечает за технические решения, архитектуру и качество реализации, но не является формальным менеджером в классическом смысле. — *Примеч. науч. ред.*

³ Фреймворк в этом контексте — это структурированный подход (мысленная рамка), который задает порядок шагов, фокус внимания и логику рассуждений при решении задачи. — *Примеч. науч. ред.*

Потребности компаний

Интервью по MSD заполняют существенный пробел в процессе приема на работу. Если на собеседованиях по структурам данных и алгоритмам оцениваются навыки программирования, то вопросы по проектированию систем оценивают компетенции уровня сеньора¹, которые сложно измерить другими способами. Такие собеседования помогают оценить ваши способности по нескольким ключевым параметрам:

- работа в условиях неопределенности, выявление требований и масштабов;
- подход к сложным и неоднозначным проектным задачам, разделение их на части;
- навыки решения проблем и глубина технических знаний;
- понимание опций или компромиссов при проектировании сложных систем;
- навыки коммуникации и работы в команде.

На собеседованиях по MSD специально рассматриваются сценарии с несколькими обоснованными подходами. При этом в одних компаниях эксперты² стремятся оценить ход рассуждений и аналитическое мышление соискателя, а в других ожидают конкретных решений, точно соответствующих устоявшимся отраслевым паттернам или же внутренним стандартам архитектуры. Поэтому на любом собеседовании важно продемонстрировать структурированный подход к решению задач, делать осознанный выбор и четко формулировать достоинства и недостатки принятых решений. Самые сильные соискатели показывают как глубину технических знаний, так и способность адаптировать принятые решения к требованиям или пожеланиям эксперта.

В отличие от собеседований по программированию, на собеседованиях по MSD редко приходится писать код. Хотя порой вас могут попросить набросать псевдокод³ или представить обмен данными между компонентами, но основное внимание в любом случае отводится решениям в области архитектуры и дизайна. Учитывая этот уникальный формат собеседования, давайте поговорим о том, чего ждут эксперты от соискателей с разными уровнями квалификации.

¹ Сеньор (от англ. «senior») — опытный специалист, обладающий глубокими знаниями в своей области и способный самостоятельно проектировать решения, принимать технические решения и менторить менее опытных коллег. В профессиональной среде также выделяют начинающих специалистов (джун, от англ. «junior») и специалистов среднего уровня (мидл, от англ. «middle»). — *Примеч. науч. ред.*

² Здесь и далее в книге под «экспертами» понимаются опытные специалисты, проводящие собеседования и обладающие необходимым набором компетенций для оценки кандидатов. — *Примеч. науч. ред.*

³ Псевдокод — это неформальный способ описания алгоритмов и логики работы системы, использующий элементы программного синтаксиса, но не привязанный к конкретному языку программирования. — *Примеч. науч. ред.*

Профессиональные уровни

Ожидания эксперта существенно различаются в зависимости от уровня должности. И хотя каждая компания устанавливает свои критерии оценки и уделяет особое внимание разным аспектам проектирования, можно назвать ряд общих параметров, актуальных для всей индустрии.



Важно отметить, что обсуждаемые здесь определения уровней основаны на опыте крупных технологических компаний, к примеру Google, Meta, Amazon и Microsoft, где инженерные лестницы четко структурированы и разделены на несколько ступеней. При этом в небольших компаниях и стартапах различия между должностями и уровнями более размыты.



ПРИМЕЧАНИЕ Едва ли претендентам на позицию джуниора придется проходить собеседование по MSD, поскольку оно оценивает продвинутые навыки, которые обычно приходят с опытом, а значит, менее показательно для новичков, делающих первые шаги по карьерной лестнице.

Мидлы

Соискатели на должность мидла должны продемонстрировать базовые знания в области проектирования мобильных систем и проявить инициативу во время собеседования. От них ожидается:

- четкое понимание проблем, умение задавать релевантные вопросы о требованиях и ограничениях;
- умение спроектировать комплексную высокоуровневую архитектуру с логически связанными компонентами;
- знание распространенных паттернов и технологий, подходящих для решения поставленной задачи;

- умение проектировать базовые структуры API, которые поддерживают требуемую функциональность;
- понимание основных подходов к управлению состоянием;
- умение оценить принятые решения с точки зрения производительности, безопасности и пользовательского опыта¹.

Задачи, которые ставятся на собеседованиях на этом уровне, обычно сводятся к комплексным функциям или сервисам средней сложности. Мидл-инженерам могут периодически требоваться подсказки, указывающие на те или иные аспекты, и это нормально. Интервьюеры ожидают этого и часто при необходимости направляют обсуждение к важным вопросам.

В процессе проектирования мидл-инженеры должны демонстрировать понимание проблем, характерных для мобильных устройств, например надежность работы сети, заряд батареи и расход памяти. Принятые решения могут не полностью охватывать все аспекты системы, однако эксперты оценивают в первую очередь умение создать работоспособную основу с четкими взаимосвязями компонентов.

Сеньоры

От соискателей на должность сеньора ожидают большей самостоятельности и технической глубины. Такие соискатели должны уметь:

- с минимальной внешней помощью определять требования к проектируемой системе;
- проактивно выявлять критические аспекты архитектуры;
- разработать комплексную высокоуровневую архитектуру, которая соответствует основным функциональным требованиям;
- распознавать и выполнять нефункциональные требования, в том числе по производительности, безопасности и масштабируемости;
- проектировать четко определенные компоненты со строго сформулированными контрактами API и паттернами взаимодействия;
- продуманно подходить к управлению состояниями приложения;
- принимать обоснованные решения при оценке преимуществ и недостатков, уметь обосновать сделанный выбор.

¹ Пользовательский опыт (User Experience, UX) — это совокупность впечатлений, ощущений и реакций пользователя, возникающих при взаимодействии с продуктом или системой. UX охватывает не только внешний вид интерфейса, но и удобство, понятность, скорость отклика, предсказуемость поведения системы и то, насколько эффективно пользователь может достичь своей цели. — *Примеч. науч. ред.*

На этом уровне соискатели могут столкнуться с теми же проблемами проектирования, что и мидл-инженеры, но требования к глубине и широте знаний будут более высокими. Естественно, сеньоры должны без подсказок эксперта учитывать такие вопросы, как обработка ошибок, пограничные случаи и потенциальные сценарии сбоев.

Сеньоры должны демонстрировать глубокое знание возможностей и ограничений мобильных платформ. Предложенные компоненты должны иметь четко определенные функции с понятными API, а общая архитектура — отражать лучшие практики создания мобильных приложений. Сеньоры должны демонстрировать понимание того, как принятые ими решения могут повлиять на пользовательский опыт, время работы без подзарядки батареи, объем сетевого трафика и другие аспекты, характерные для мобильных устройств.

Стафф+

Соискатели на должности уровня стафф⁺¹ в течение всего собеседования должны демонстрировать лидерство и техническое мастерство. В том числе они должны уметь:

- проводить тщательный и проактивный сбор требований с минимальным вмешательством эксперта;
- уверенно вести диалог и с легкостью находить решения в условиях неопределенности;
- проектировать эффективно масштабируемые системы и демонстрировать глубокое понимание проблем, возникающих в сложных системах, а также их последствий;
- грамотно и уверенно оперировать технической терминологией;
- быстро принимать обоснованные решения, опираясь на обширный опыт;
- предвидеть будущие проблемы и проектировать систему с учетом возможного масштабирования;
- демонстрировать стратегическое мышление, учитывать влияние принятых решений на бизнес и эволюцию продукта;
- разрабатывать стратегии оценки и снижения рисков.

Соискатели уровня стафф+ должны самостоятельно, без дополнительных указаний ставить и решать сложные вопросы, разбираться в изощренных страте-

¹ Стафф++ — условное обозначение уровней выше senior (Staff, Senior Staff, Principal и аналогичных), на которых специалист влияет не только на конкретную систему, но и на архитектурные решения, техническую стратегию и качество инженерных практик в масштабах команды, продукта или всей организации. — *Примеч. науч. ред.*

гиях кэширования, архитектуре offline-first¹, кросс-платформенности², оптимизации производительности и восстановлении после сбоев.

Принимаемые на этом уровне решения часто требуют сложных компромиссов между пользовательским опытом, производительностью, безопасностью и другими факторами. Поэтому эксперты ожидают от соискателей не только умения создавать эффективные архитектурные решения, но и демонстрации стратегического мышления в сложных ситуациях. Проектные решения соискателей должны отражать глубокое понимание паттернов мобильной архитектуры и возможностей оптимизации под те или иные платформы либо экосистемы, где будет работать мобильное приложение.

Интервьюеры могут намеренно вводить ограничения или пограничные случаи, после чего оценивать, как соискатели оптимизируют уже принятые решения. Поэтому критическую важность приобретает способность быстро адаптироваться, понимать последствия выбора той или иной опции для дизайнера всей системы, четко обосновывать принятые решения.

Как эффективно работать с этой книгой

В книге предлагается комплексный подход к собеседованиям по MSD. Она имеет четкую структуру, содержит подробные практические примеры. Чтобы извлечь из нее максимальную пользу, при чтении следует обратить внимание на следующие аспекты.

Последовательная структура

Предлагаемый к ознакомлению материал изложен в хорошо продуманной последовательности. В первых главах значительное внимание уделяется выбору базовой архитектуры и паттернам взаимодействия клиента и бэкенда³, объясняются ключевые принципы, лежащие в основе любых мобильных систем. В последующих главах акцент делается на новых проблемах, характерных для соответствующих практических примеров.

¹ Offline-first-архитектура — подход к проектированию приложений, при котором приложение остается работоспособным без сетевого подключения, используя локальное хранилище, а синхронизация с сервером выполняется при появлении сети. — *Примеч. науч. ред.*

² Кросс-платформенность — подход к разработке, при котором одна кодовая база и/или единые архитектурные решения используются для создания приложений под несколько платформ (например, Windows и macOS) с учетом их технических и пользовательских особенностей. — *Примеч. науч. ред.*

³ Бэкенд (от англ. «backend») — серверная часть приложения, отвечающая за бизнес-логику, хранение и обработку данных, а также взаимодействие с клиентскими приложениями. — *Примеч. науч. ред.*

Практические примеры из уникальных проблемных областей

Проектирование мобильных систем — сложная проблемная область, которая требует знания множества технических вопросов. Для их эффективного изучения можно воспользоваться табл. 1.1, которая соотносит те или иные задачи с главами, в которых они описаны. Такой подход позволит охватить широкий круг тем, важных для технических собеседований.

Таблица 1.1. Основные темы, рассматриваемые в книге

Глава	Темы
Глава 3. Лента новостей	REST API, пагинация, работа в режиме офлайн, оптимистичные обновления, нативный рендеринг, WebView
Глава 4. Мессенджер	WebSocket, проектирование баз данных, пуш-уведомления, генерация идентификаторов на сервере
Глава 5. Трейдинговая платформа	Визуализация графиков, сценарии постоянного подключения, WebView, буферизованные обновления интерфейса
Глава 6. Библиотека пагинации	Проектирование библиотек, дженерики, кэширование в оперативной памяти, приоритизация API-запросов, модульная архитектура, стратегии версионирования
Глава 7. Сервис бронирования отелей	Бронирование, синхронизация по времени, обработка платежей, автозаполнение при поиске, предварительная загрузка данных, полнотекстовый поиск
Глава 8. Google Drive	Управление файлами, стратегии хранения, блочная синхронизация, возобновляемая загрузка файлов, управление историей версий, шифрование данных
Глава 9. YouTube	Выбор полей в HTTP-запросе, медиастриминг, предварительная загрузка контента, улучшение пользовательского опыта при просмотре видеороликов
Глава 10. Стандартные блоки для проектирования мобильных систем	<p>Архитектура (распространенные паттерны, слои архитектуры, преобразование моделей данных, модульность, внедрение зависимостей, тестирование).</p> <p>Хранение данных (типы данных, видимость файлов).</p> <p>Сетевое взаимодействие (REST, аутентификация, CDN, пагинация).</p> <p>Разработка функциональности (поэтапные обновления, принудительное обновление, флаги функций, наблюдаемость, локализация, конфиденциальность, CI/CD, доступность, производительность, пуш-уведомления, размеры приложений).</p> <p>Поддержка устройств (форм-факторы, версии ОС)</p>

Помимо собеседования

Как можно заметить, **в каждой из глав приводится больше сведений, чем может потребоваться для часового собеседования.** Так сделано намеренно. Подробные объяснения помогут вам разобраться не только в том, какие решения следует принимать, но и почему они имеют смысл в условиях определенных ограничений и требований. Такое глубокое понимание позволит вам:

- уверенно объяснять эксперту причины принятых решений;
- адаптировать выбранные подходы к неожиданным требованиям и ограничениям;
- выявлять достоинства и недостатки, обосновывать необходимые компромиссы.

Дальнейшее обучение

В конце каждого кейса вы найдете дополнительные требования и предложения по функциям, которые не были рассмотрены во время обсуждения. Эти дополнения — отличная возможность для практики и проверки знаний: вы сможете самостоятельно принять решения, соответствующие этим требованиям, сравнив свой подход с рекомендациями книги.

Допустимые варианты

В этой книге мы предлагаем проектные решения, основанные на нашем собственном анализе и оценке требований. Вы можете не согласиться с некоторыми предложениями, и это нормально. Реальное проектирование мобильных систем часто подразумевает несколько возможных решений одной и той же задачи.

Если вы предпочитаете подход, не совпадающий с предложенным нами, попрактикуйтесь, подумайте, почему ваше решение будет работать лучше, учтите контекст и ограничения. Используйте эту книгу как руководство, а не как сборник готовых ответов. Такой подход помогает развивать критическое мышление, которое полезно при прохождении сложных собеседований и проектировании реальных систем.

Глава 2

План собеседования

Чтобы успешно пройти собеседование по проектированию мобильных систем, необходимо иметь четкую схему (план действий), иначе общение с экспертом станет непоследовательным и создаст трудности обеим сторонам. В отличие от задач по кодированию, которые предлагают соискателям на должность разработчика, задания по проектированию систем могут иметь несколько правильных решений, и это придает особую важность методологической стратегии.

Ниже описан примерный план собеседования, который состоит из пяти этапов, или шагов. Здесь мы лишь кратко рассмотрим каждый из них: подробные объяснения и практические советы будут даны в следующих главах.

Пошаговая схема эффективного собеседования

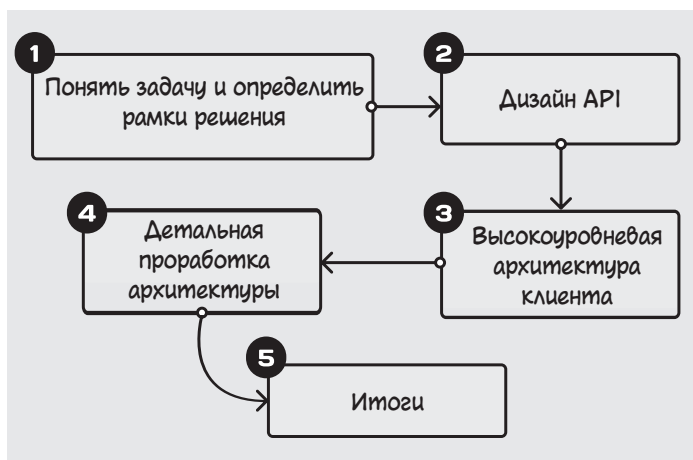


Рис. 2.1. Общий план собеседования

Шаг 1. Понять задачу и определить рамки решения

Обычно собеседование начинается с предложений типа «Спроектируйте ленту новостей» или же «Спроектируйте библиотеку пагинации». Казалось бы, тут все ясно. Однако на этом критически важном шаге необходимо точно понять, что именно предстоит проектировать.

Ваша задача — при помощи целенаправленных вопросов уточнить начальные условия и желаемый конечный результат, то есть определить масштаб (рамки) решения. Ответы эксперта помогут правильно подобрать нужные технологии и заложат фундамент вашего ответа. В этом смысле каждое собеседование уникально.

Какие же вопросы следует задать? Вот примерный список, с которого можно начать:

- **Что мы проектируем?** Этот вопрос прольет свет на требования к дизайну — функциональные и нефункциональные, а также позволит определить, какие аспекты можно оставить за его рамками. К примеру, можно спросить:
 - *Какие функции, экраны и критические пользовательские сценарии¹ (critical user journeys, CUJ) мы проектируем?*
 - *Нужна ли предварительная загрузка данных? Следует ли предусмотреть аутентификацию пользователей?*
- **Для кого мы проектируем?** Этот вопрос поможет понять масштабы и характер применения приложения, а значит — оценить, насколько производительным должно быть предлагаемое решение. К примеру, можно спросить:
 - *На какое количество активных пользователей в день мы рассчитываем? Ожидаем ли мы, что их число увеличится?*
 - *Мы создаем минимально жизнеспособный или же полностью готовый продукт?*
 - *На какой рынок мы ориентируемся²? Весь мир или какой-то отдельный регион? Каковы типовые сценарии использования? Дома или в поездке?*
- **Каковы системные ограничения?** Этот вопрос позволит определить границы системы и правильно выбрать технологии. К примеру, можно спросить:
 - *На какие мобильные платформы мы ориентируемся (например, iOS, Android или же и то и другое)?*
 - *Существуют ли API или системы, с которыми нужно интегрировать будущее приложение?*

Во время сбора информации вы заметите, что ответы эксперта раскрывают перед вами и другие аспекты будущей системы. Не бойтесь делать разумные

¹ Критический пользовательский сценарий (critical user journey) — ключевая последовательность действий пользователя, ведущая к получению основной ценности продукта. — *Примеч. науч. ред.*

² В контексте этих вопросов также используется термин «целевая аудитория» (ЦА) — группа пользователей, на которых ориентирован продукт и чьи потребности он должен учитывать при проектировании. Например: «На какую целевую аудиторию рассчитан этот продукт?». — *Примеч. науч. ред.*

предположения на основе услышанного. Старайтесь формулировать их как можно более четко, чтобы эксперт смог устранить любые недопонимания.

Однако ваша цель не в том, чтобы задать как можно больше вопросов. Делайте акцент на самых насущных проблемах, думайте о том, что важно уточнить, опираясь на здравый смысл и собственное суждение.



ПОМНИТЕ От полученных ответов зависит, какие технические решения вы примете и на какие компромиссы пойдете. В дальнейшем, когда вы будете объяснять свои действия эксперту, ссылайтесь на его ответы.

Шаг 2. Дизайн API

При проектировании API¹ необходимо зафиксировать контракт между клиентским приложением и внешними зависимостями. Этот шаг очень важен, поскольку он позволяет убедиться, что ваше понимание задачи совпадает с ожиданиями интервьюера, и закладывает прочный фундамент для последующей работы.

Проектирование приложения, которое должно взаимодействовать с бэкендом, необходимо начинать с выбора подходящих для конкретного случая протоколов и эндпоинтов². С другой стороны, при работе над мобильной библиотекой основное внимание следует уделить планированию публичного API, при помощи которого приложения будут сначала инициализировать и настраивать библиотеку, а затем и работать с ней.

Помимо самого API нужно продумать и модели данных, которые будут использоваться. Они должны четко соответствовать структурам данных и взаимосвязям между ними независимо от реализации (JSON-данные, структуры Swift, классы данных Kotlin или же Protocol Buffers для повышения эффективности).

Шаг 3. Высокоуровневая архитектура клиента

После API наступает очередь высокоуровневой схемы архитектуры клиента, которая показывает взаимосвязи между компонентами приложения.

Она определяет пространство для проектирования, позволяет наглядно судить о соответствии дизайна предъявленным требованиям и — что еще более важно — о соответствии всего потока данных необходимой функциональности

¹ API (Application Programming Interface) — интерфейс взаимодействия между программными компонентами, описывающий доступные операции и формат передаваемых данных. — *Примеч. науч. ред.*

² Эндпоинт (от англ. «endpoint») — элемент API, описывающий конкретную операцию и способ обращения. — *Примеч. науч. ред.*

системы. При построении схемы архитектуры следует выявить проблемные области, которые требуют дополнительного обсуждения на следующем шаге работы.

Шаг 4. Детальная проработка архитектуры

На этом шаге, действуя совместно с экспертом, вам предстоит выявить узкие места, а также тщательно проработать некоторые компоненты. В общем случае нужно наметить две-три темы для обсуждения, хотя все зависит от обстоятельств.

Не забывайте обновлять схему архитектуры по результатам диалога.

Кроме того, следует прислушиваться к реакции эксперта и его вопросам. Если его интересует несколько тем, старайтесь уделить внимание каждой из них, не выбиваясь при этом за рамки отведенного для собеседования времени.

Шаг 5. Итоги (необязательно)

Если позволит время, вы можете использовать этот последний шаг, чтобы продемонстрировать способность критически оценить предложенный дизайн и наметить его развитие в будущем. Вот несколько советов, как эффективно подвести итоги собеседования:

- **Обобщите основные решения.** Если во время обсуждения вы сравнивали варианты реализации, оценивая их плюсы и минусы, или же приняли непростые решения, попробуйте подвести краткий итог. Это поможет закрепить основные аспекты дизайна и убедиться в том, что эксперт правильно понял ваши доводы, особенно если вы говорили о технических деталях.
- **Продemonстрируйте навык критического мышления.** Если вас спросят о том, как можно улучшить предложенный вами дизайн, ни в коем случае не говорите, что он идеален и полностью завершен. Попробуйте обсудить возможные пути его развития. Это покажет, что вы обладаете навыками самоанализа и критического мышления — это высоко ценится экспертами.
- **Проанализируйте пограничные случаи.** Воспользуйтесь возможностью для обсуждения потенциальных сбоев и ошибок, которые не были рассмотрены до этого. Так вы покажете, что можете оценивать надежность системы и пользовательский опыт не только при благоприятных, но и при иных условиях.
- **Оцените возможности масштабирования.** Подумайте о том, сможет ли система работать при большем количестве пользователей и развиваться более многочисленной командой разработчиков. Так вы покажете, что думаете о будущем развитии дизайна и понимаете, с какими трудностями — техническими и организационными — можно столкнуться при масштабировании.