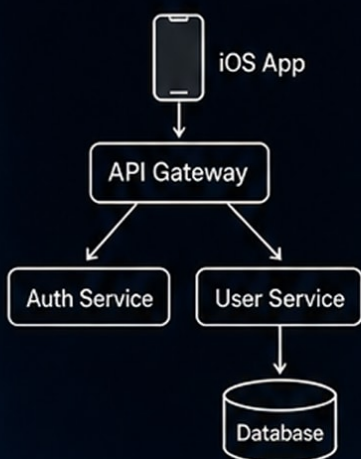


Share Your Screen.

Mastering the Senior iOS Live Coding and System Design Interview

Take-home projects, code review, and the communication strategies that pass senior and staff interviews, in Swift.

System Design



Take-Home Project Checklist

- ☒ Clear Requirements
- ☒ Architecture & Data Flow
- ☒ Core Functionality
- ☒ Edge Cases
- ☒ UI/UX Polish
- ☒ Testing & Documentation

Communication Strategies

- Think out loud.
- Ask clarifying questions.
- Align early, iterate together.
- Summarize and confirm.



REAL
STRATEGIES.
REAL
INTERVIEWS.
REAL
RESULTS.

MIKE SALARI

Contents

A note on research and examples	5
How to use the companion repository	6
Free updates and direct support	7
Preface	8
Introduction	10
Part 1: Mindset and foundations	13
Chapter 1: The new reality of iOS interviews in 2026	14
Chapter 2: Senior signals, what actually gets you hired	18
Chapter 3: The interviewer's perspective	28
Part 2: Core iOS skills	32
Chapter 4: Concurrency and race conditions	33
Chapter 5: Memory management and retain cycles	49
Chapter 6: Modern Swift in 2026	56
Chapter 7: Request lifecycle management	66
Chapter 8: iOS app architecture	74
Chapter 9: Data persistence and offline	80
Chapter 10: Networking and API integration	85
Chapter 11: Testing and testability	90
Part 3: Interview formats	96
Decoding the round: what "iOS coding exercise" actually means	97
Chapter 12: Live app building, strategy and mindset	105
Chapter 13: Code review and refactoring	117
Chapter 14: Mobile system design	123
Chapter 15: Take-home projects and review discussions	134
Chapter 16: Behavioral and leadership questions	144
Chapter 17: Debugging and diagnostics rounds	150
Part 4: Case studies, annotated end to end	156
Case study: a passed token-refresh round	157
Case study: an offline-first music app design	163
Case study: building a searchable list under time	170
Case study: hunting a memory leak live	175
Case study: a live SOLID refactor	180
Case study: a mixed senior tech round	185
Case study: an algorithm round for an iOS role	190
Case study: a staff-level behavioral round	194
Case study: a SwiftUI take-home review	198
Part 5: Preparation and practice	202

Chapter 18: How to prepare, a practical 4-week plan	203
Chapter 19: Putting it all together	208
Chapter 20: Working through the companion repository	214
Part 6: The practice workbook	219
Candidate worksheets: before, during, and after	220
Technical checklists	222
Timed drills and the practice map	224
Part 7: Modern iOS practices and deep dives	227
Modern testing and persistence, at senior depth	228
Architecture patterns and API design, compared	234
Performance profiling and UIKit depth	241
Part 8: The problem bank	247
Part 9: Career and positioning	452
Chapter 21: Resume, LinkedIn, and personal branding	453
Chapter 22: Rejection handling and follow-up	459
Chapter 23: The interviewer's edition	464
Chapter 24: Conclusion, thinking like a senior engineer	467
Appendix A: All scoring rubrics	469
Appendix B: Senior signals by format	476
Appendix C: Recovery phrases cheat sheet	478
Appendix D: Pre-interview and night-before checklist	480
Appendix E: Key frameworks summary	482
Appendix: Quick reference, answer ladders and checklists	484
Appendix: iOS fundamentals, answered at senior level	486
Appendix F: Full annotated mock interviews	489
Appendix: Android for iOS engineers, a bonus	540
Staying composed under pressure and interruption	544
Communicating and disagreeing under pressure	552
Codebase navigation drills	558
About the author	569
Recommended reading and tools	570
Acknowledgments	571
Updates, errata, and version history	572
Appendix K: Company loop maps	574
Also by Mike Salari	581

Chapter 1: The new reality of iOS interviews in 2026

The way senior iOS interviews work has shifted more in the last two years than in the previous five. Two forces drove the change: AI coding assistants became a normal part of the job, and Swift itself moved on with Swift 6, strict concurrency, and the Observation framework. If you are preparing with a mental model from 2022, you are preparing for an interview that no longer exists.

From the research. Across the interview sessions, first-hand accounts, and public interview reports studied for this book, live app building and debugging appeared far more often than algorithm puzzles for roles outside the largest tech companies. Many strong candidates had it backwards, grinding data-structure problems while under-practicing the build and debug rounds they were actually going to face. This book weights its practice the way real loops do.

Figure 1.1: The senior iOS interview formats at a glance.

iOS interview formats at a glance

Format	Typical length	Primary signals
Live app building	45 to 60 min	Communication, scope, safety
Code review	40 to 60 min	Prioritization, trade-offs
Mobile system design	45 to 60 min	Trade-offs, mobile constraints
Take-home + review	hours + 45 min	Decisions, ownership, testing
Behavioral	30 to 45 min	Ownership, reflection

This chapter sets the landscape. The rest of the book is how to win in it.

AI changed what a coding round can even measure

By 2026, AI assistants like Copilot, Cursor, and Claude are part of most engineers' daily workflow. That broke something in the traditional interview. When generating decent-looking code is trivial, watching a candidate produce decent-looking code tells you very little.

The effect is uneven across formats:

Share Your Screen

Senior iOS live coding and system design interview

Get the full book

581 pages · 75 solved Swift problems with tests
9 real case studies · 7 annotated mock interviews
Companion Swift package · free lifetime updates
PDF and EPUB · direct support from the author

Buy on Gumroad

<https://mikesalari.dev//share-your-screen-live-coding-ios>

Read more: shareyourscreen.dev

Author: salari.dev