

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

— SALESFORCE ARCHITECTURE REFERENCE

NexGen Living Complete Guide

Everything built across this conversation – flows, data model, picklists, Git setup, and interview prep – organized into one searchable reference.

Sales Cloud

Service Cloud

Experience Cloud

Revenue Cloud

Field Service

Agentforce

Marketing Cloud

— OVERVIEW

Business Model

NexGen Living is a for-profit PropTech marketplace that pairs seniors who have a spare room with young professionals or students who need affordable housing.

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

REVENUE STREAM 1

\$149/mo — Senior Host Platform Fee

Senior homeowners pay a monthly platform fee. Includes vetting, matching, service guarantee, quarterly coordinator check-in, 24/7 emergency support.

REVENUE STREAM 3

15% — Property Placement Fee

One-time fee at match signing. Equal to 15% of one month's notional rent savings. Recognized as Opportunity revenue on DocuSign completion.

REVENUE STREAM 5

\$500 — Partner Referral Bounty

Paid to universities, senior centers, real estate brokers per completed placement. Commission__c record auto-created on match activation.

REVENUE STREAM 2

\$89/mo — Renter Access Fee

Young renters pay a small fee — far less than their rent savings. Waived first month as acquisition incentive. Offset by credit if senior non-compliant on hours.

REVENUE STREAM 4

\$299 — Premium Vetting Upgrade

Optional upsell: enhanced background check, in-person interview by NexGen specialist, priority placement queue. One-time Product2 line item.

REVENUE STREAM 6

B2B — Corporate Housing Partnerships

Hospitals and universities pay annual licensing fee. Bulk placement capacity, branded portal skin. Account.Type = "Corporate Partner."

PLATFORM

Clouds Used

All accessible from a single free Developer Edition org.

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

SALES CLOUD

Lead acquisition, dual-sided pipeline, opportunity management, matchmaking engine, renewal automation, churn scoring, referral commissions.

EXPERIENCE CLOUD

Three portals: Haven (seniors), Base (renters), Partner Hub (universities/brokers). Screen Flows embedded per portal.

MARKETING CLOUD

Two separate journeys (seniors and renters), live Salesforce data in dynamic email content, Einstein Send-Time Optimization, NPS triggers.

SERVICE CLOUD

Case management, tiered SLA enforcement, IoT platform events, Agentforce case classification, escalation chains, mediation tracking.

REVENUE CLOUD

Subscription products, recurring billing, SLA breach credits, CPI escalation, invoice generation, dunning sequences.

FIELD SERVICE + AGENTFORCE

Property assessment dispatch, maintenance routing, technician mobile checklists. Agentforce reads case descriptions, suggests classification, surfaces knowledge articles.

SALES CLOUD

Acquisition & Matching Flows

Flow A — Dual-Sided Lead Acquisition

Screen Flow Record-Triggered Lead → Opportunity

01 Two — One targeting Persona__c or "Young Renter" . Record-Triggered Flow fires on Lead creation. Experience seniors, one renters.

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

	Cloud landing pages	Each embeds a Screen Flow creating a Lead with	Host"		Routes to separate Marketing Cloud journeys via Platform Event.
02	Lead Scoring	– Record-Triggered Flow calculates	Match_Readiness_Score__c	. Seniors scored on room availability, location, mobility, timeline. Renters on income, move-in date, background check consent.	Score ≥ 80 → auto-assign to specialist. Score < 40 → nurture sequence. Mid-range → SDR Task.
03	Property Assessment Opportunity	– Senior Host score ≥ 80: Screen Flow converts Lead, creates Opportunity Stage = "Site Visit Scheduled." Field Service dispatches assessor.		FSL Work Order linked to Opportunity. On WO completion, Opportunity auto-advances to "Property Listed."	
04	Renter Vetting Pipeline	– Renter Lead converts to Renter_Application Opportunity. Income doc upload, Checkr background check, reference Tasks loop.		Three gates (income, background, references) must clear. Each gate = Decision on custom checkbox. Failed gate → Task, not auto-rejection.	
05	Scheduled Flow Matchmaking Engine	– Queries both pools where Status = "Ready to Match." Scores compatibility across 8 criteria. Creates top-3 Match_Proposal__c records per open property.		This is the WOW pattern: declarative matchmaking without Apex. Runs nightly.	
06	Revenue Cloud Activation	– On match acceptance and DocuSign completion, activates two recurring subscriptions simultaneously: Senior Host \$149/mo and Renter \$89/mo.		One signed match = two active revenue streams. Start Date = Move-In Date.	

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

SERVICE CLOUD

SLA & Case Flows

Flow C — Tiered SLA Enforcement with Automatic Rent Abatement

- Record-Triggered
- Scheduled Path
- Service Cloud

EXPERIENCE CLOUD

Three Portals, One Org

PORTAL 1 — HAVEN

Senior Host Portal

Large fonts, high contrast, simplified navigation. Seniors view renter's hours log, submit maintenance requests, review monthly statements, access coordinator contact. Screen Flow for wellbeing check-ins replaces survey emails. Content personalization based on Mobility_Level_c.

PORTAL 2 — BASE

Young Renter Portal

Mobile-first. Swipe-style hours logging Screen Flow. Rent Savings Dashboard comparing effective rent vs. ZIP median (updated monthly by Scheduled Flow). Maintenance requests, coordinator messaging, Housing Agreement access. Push notifications via Notification Builder.

PORTAL 3 — PARTNER HUB

University & Broker Portal

For referral partners. View referred applicants, track conversion status, see commission earnings, download co-branded marketing materials. Self-

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

service commission inquiries via Cases tagged
Origin = "Partner Portal."

Architecture note: All three portals run from one org using different profiles, permission sets, and Experience Workspaces. Record visibility controlled by Sharing Rules – no code deployment differences between portals.

REVENUE CLOUD

Billing & Payment Flows

Flow B — Multi-Stream Revenue with Dynamic Rent Offset

Scheduled Flow

Record-Triggered

Revenue Cloud

SALES CLOUD

Renewal & Churn Flows

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

Flow E – 120-Day Renewal Engine with Churn Risk Scoring

Scheduled Flow

Record-Triggered

DATA MODEL

Custom Objects

Build these in Object Manager before creating any flows. Match__c is the spine of everything.

Match__c ← spine <table><tr><td>Senior_Host__c</td><td>Lookup(Contact)</td></tr><tr><td>Young_Renter__c</td><td>Lookup(Contact)</td></tr><tr><td>Status__c</td><td>Picklist</td></tr><tr><td>Compatibility_Score__c</td><td>Number</td></tr><tr><td>Churn_Risk_Score__c</td><td>Number</td></tr><tr><td>Monthly_Hours_Required__c</td><td>Number</td></tr><tr><td>Senior_Accepted__c</td><td>Checkbox</td></tr><tr><td>Renter_Accepted__c</td><td>Checkbox</td></tr><tr><td>Agreement_Signed__c</td><td>Checkbox</td></tr><tr><td>DocuSign_Envelope__c</td><td>Text</td></tr></table>	Senior_Host__c	Lookup(Contact)	Young_Renter__c	Lookup(Contact)	Status__c	Picklist	Compatibility_Score__c	Number	Churn_Risk_Score__c	Number	Monthly_Hours_Required__c	Number	Senior_Accepted__c	Checkbox	Renter_Accepted__c	Checkbox	Agreement_Signed__c	Checkbox	DocuSign_Envelope__c	Text	Housing_Unit__c <table><tr><td>Senior_Host__c</td><td>Lookup(Contact)</td></tr><tr><td>Status__c</td><td>Picklist</td></tr><tr><td>Market_Rent_ZIP__c</td><td>Currency</td></tr><tr><td>Effective_Renter_Rent__c</td><td>Currency</td></tr><tr><td>Property_Score__c</td><td>Number</td></tr><tr><td>NexGen_Certified__c</td><td>Checkbox</td></tr><tr><td>IoT_Device_ID__c</td><td>Text</td></tr><tr><td>Rooms_Available__c</td><td>Number</td></tr></table>	Senior_Host__c	Lookup(Contact)	Status__c	Picklist	Market_Rent_ZIP__c	Currency	Effective_Renter_Rent__c	Currency	Property_Score__c	Number	NexGen_Certified__c	Checkbox	IoT_Device_ID__c	Text	Rooms_Available__c	Number	Companion_Hours__c <table><tr><td>Match__c</td><td>MD(Match__c)</td></tr><tr><td>Activity_Type__c</td><td>Picklist</td></tr><tr><td>Activity_Date__c</td><td>Date</td></tr><tr><td>Duration_Hours__c</td><td>Number</td></tr><tr><td>Senior_Confirmed__c</td><td>Checkbox</td></tr><tr><td>Log_Channel__c</td><td>Picklist</td></tr></table>	Match__c	MD(Match__c)	Activity_Type__c	Picklist	Activity_Date__c	Date	Duration_Hours__c	Number	Senior_Confirmed__c	Checkbox	Log_Channel__c	Picklist
Senior_Host__c	Lookup(Contact)																																																	
Young_Renter__c	Lookup(Contact)																																																	
Status__c	Picklist																																																	
Compatibility_Score__c	Number																																																	
Churn_Risk_Score__c	Number																																																	
Monthly_Hours_Required__c	Number																																																	
Senior_Accepted__c	Checkbox																																																	
Renter_Accepted__c	Checkbox																																																	
Agreement_Signed__c	Checkbox																																																	
DocuSign_Envelope__c	Text																																																	
Senior_Host__c	Lookup(Contact)																																																	
Status__c	Picklist																																																	
Market_Rent_ZIP__c	Currency																																																	
Effective_Renter_Rent__c	Currency																																																	
Property_Score__c	Number																																																	
NexGen_Certified__c	Checkbox																																																	
IoT_Device_ID__c	Text																																																	
Rooms_Available__c	Number																																																	
Match__c	MD(Match__c)																																																	
Activity_Type__c	Picklist																																																	
Activity_Date__c	Date																																																	
Duration_Hours__c	Number																																																	
Senior_Confirmed__c	Checkbox																																																	
Log_Channel__c	Picklist																																																	
Monthly_Hours_Report__c <table><tr><td>Match__c</td><td>Lookup(Match__c)</td></tr><tr><td>Report_Month__c</td><td>Date</td></tr><tr><td>Hours_Logged__c</td><td>Number</td></tr><tr><td>Hours_Required__c</td><td>Number</td></tr><tr><td>Compliance_Status__c</td><td>Picklist</td></tr></table>	Match__c	Lookup(Match__c)	Report_Month__c	Date	Hours_Logged__c	Number	Hours_Required__c	Number	Compliance_Status__c	Picklist	Invoice__c <table><tr><td>Match__c</td><td>Lookup(Match__c)</td></tr><tr><td>Billed_Contact__c</td><td>Lookup(Contact)</td></tr><tr><td>Invoice_Date__c</td><td>Date</td></tr><tr><td>Total_Amount__c</td><td>Currency</td></tr><tr><td>Status__c</td><td>Picklist</td></tr></table>	Match__c	Lookup(Match__c)	Billed_Contact__c	Lookup(Contact)	Invoice_Date__c	Date	Total_Amount__c	Currency	Status__c	Picklist	Credit_Memo__c <table><tr><td>Match__c</td><td>Lookup(Match__c)</td></tr><tr><td>Case__c</td><td>Lookup(Case)</td></tr><tr><td>Credit_Type__c</td><td>Picklist</td></tr><tr><td>Amount__c</td><td>Currency</td></tr><tr><td>Applied_Invoice__c</td><td>Lookup(Invoice__c)</td></tr></table>	Match__c	Lookup(Match__c)	Case__c	Lookup(Case)	Credit_Type__c	Picklist	Amount__c	Currency	Applied_Invoice__c	Lookup(Invoice__c)																		
Match__c	Lookup(Match__c)																																																	
Report_Month__c	Date																																																	
Hours_Logged__c	Number																																																	
Hours_Required__c	Number																																																	
Compliance_Status__c	Picklist																																																	
Match__c	Lookup(Match__c)																																																	
Billed_Contact__c	Lookup(Contact)																																																	
Invoice_Date__c	Date																																																	
Total_Amount__c	Currency																																																	
Status__c	Picklist																																																	
Match__c	Lookup(Match__c)																																																	
Case__c	Lookup(Case)																																																	
Credit_Type__c	Picklist																																																	
Amount__c	Currency																																																	
Applied_Invoice__c	Lookup(Invoice__c)																																																	

OVERVIEW

• Business Model

• Clouds Used

FLows

• Sales Cloud Flows

• Service Cloud Flows

• Experience Cloud

• Revenue & Billing

• Renewal & Churn

DATA MODEL

• Custom Objects

• Picklist Values

• Relationships

ARCHITECTURE

• Clever Patterns

• Build Order

• Schema Builder Tips

DEVELOPER SETUP

• Dev Edition Setup

• VS Code & CLI

Credit_Generated__c	Currency	Stripe_Payment_ID__c	Text	Status__c	Picklist
Failure_Count__c	Number				
Match_Proposal__c		Wellbeing_Check__c		Commission__c	
Senior_Host__c	Lookup(Contact)	Match__c	Lookup(Match__c)	Partner_Account__c	Lookup(Account)
Young_Renter__c	Lookup(Contact)	Respondent__c	Lookup(Contact)	Match__c	Lookup(Match__c)
Compatibility_Score__c	Number	Comfort_Score__c	Number(1-5)	Amount__c	Currency
Rank__c	Number(1-3)	NPS_Score__c	Number(0-10)	Status__c	Picklist
Status__c	Picklist	Concerns__c	TextArea	Payout_Date__c	Date
Generated_By_Flow__c	Checkbox	Flag_For_Review__c	Checkbox		
Contact (extended)					
Persona__c	Picklist				
Match_Readiness__c	Number				
Churn_Risk__c	Number				
Latest_NPS__c	Number				
NPS_Trend__c	Formula				
Portal_Last_Login__c	DateTime				
Review_Score__c	Number				

DATA MODEL

Match__c Relationship Fields

Two lookup fields to Contact on the same object – each needs distinct names to avoid conflicts.

FIELD	CHILD RELATIONSHIP NAME	RELATED LIST LABEL	ON DELETE
Senior_Host__c	Senior_Host_Matches	Matches as Senior Host	Don't allow deletion

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

Young_Renter__c

Young_Renter_Matches

Matches as Young Renter

Don't allow deletion

Important: Set "Don't allow deletion of the lookup record if it has related Match records" on both fields. This prevents accidental deletion of a Contact with an active Match.

DATA MODEL

All Picklist Values

Match__c – Status__c

Tip: Set "Active" as default. After creating this field go to Chart Colors and assign green = Active, orange = At Risk, red = Suspended, grey = Ended statuses.

VALUE	MEANING
Proposed	System created proposal – neither party has responded
Pending Senior Acceptance	Renter accepted, waiting on senior
Pending Renter Acceptance	Senior accepted, waiting on renter
Both Accepted	Both said yes – intro meeting not yet scheduled
Intro Meeting Scheduled	Meeting on the calendar
Agreement Sent	DocuSign envelope out for signatures
Agreement Signed	Both signed – move-in pending
Active	Currently living together ← DEFAULT

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

At Risk	Churn score crossed threshold or hours warnings
On Probation	Third hours violation – formal probation
Suspended	Payment failure or serious violation
Ended – Renewed	Lease concluded and renewed
Ended – Natural Conclusion	Full term, not renewed, clean exit
Ended – Mutual Agreement	Both parties agreed to end early
Ended – Renter Violation	Terminated due to renter non-compliance
Ended – Senior Withdrawal	Senior chose to exit
Declined	One or both parties declined the match

Housing_Unit__c – Status__c

VALUE	MEANING
Assessment Scheduled	Field Service WO created, visit pending
Assessment Complete	Home visit done, unit scored
Listed – Available	On platform, visible in matching pool
Listed – Under Review	In active matching, not yet committed
Occupied	Active match in place
Temporarily Unavailable	Senior traveling, not accepting renters
Suspended	Removed from platform due to condition/violation
Delisted	Senior permanently exited platform

Invoice__c – Status__c

VALUE	MEANING
Draft	Generated but not yet sent

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

Sent	Delivered to customer
Payment Pending	Stripe charge initiated
Paid	Payment confirmed
Partial Payment	Some received, balance outstanding
Failed – Retry 1	First attempt failed
Failed – Retry 2	Second attempt failed, escalating
Failed – Final	Third failure, suspension triggered
Voided	Cancelled before payment
Written Off	Uncollectable, bad debt

Contact – Program_Status__c

VALUE
Inquiry
Application Started
Application Submitted
Background Check Pending
Background Check Cleared
Background Check Failed
References Pending
Interview Scheduled
Home Visit Scheduled
Approved – Ready to Match
Matched – Active
Between Matches

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

On Hold

Withdrawn

Declined

Churned

ARCHITECTURE

8 Patterns That Impress Interviewers

PATTERN 01 · SALES CLOUD

Scheduled Flow as Matching Engine

Instead of Apex, a Scheduled Flow queries both pools, scores compatibility using formula fields, auto-creates ranked Match_Proposal__c records nightly. Most candidates would reach for Apex here.

PATTERN 03 · REVENUE CLOUD

Contractual SLA → Automatic Invoice Credit

SLA breach flips a checkbox. Record-Triggered Flow calculates penalty, creates Credit_Memo__c, injects negative line item into next open Invoice. Connects Service Cloud to Revenue Cloud declaratively.

PATTERN 05 · SALES CLOUD

PATTERN 02 · SERVICE CLOUD

Platform Event as IoT Bridge

Smart home sensors publish Platform Events when triggered. Platform Event-Triggered Flow converts payload into Priority Critical Case. Genuinely real-time. Platform Event triggers are rare on resumes.

PATTERN 04 · EXPERIENCE CLOUD

Three Portals, One Org, Zero Code

Three Experience Cloud sites from one org using different profiles, permission sets, and Experience Workspaces. Record visibility via Sharing Rules. Same flows, three different experiences.

PATTERN 06 · MARKETING CLOUD

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

Declarative Churn Risk Score

Five behavioral signals – NPS trend, hours warnings, billing disputes, case volume, portal logins – each weighted, summed via Assignment elements. Score ≥ 60 fires retention Case. No ML, no Apex.

PATTERN 07 · AGENTFORCE

AI-Assisted Case Classification

Agentforce Agent (configured in Agent Builder) reads Case descriptions, suggests Type and Priority, surfaces knowledge articles. Human confirms or overrides. Built declaratively – no code.

Live SF Data in Email Dynamic Content

AMPscript in email template queries a Data Extension synced nightly from Housing_Unit__c. "Your neighborhood has [X] available hosts" is live data, not static copy.

PATTERN 08 · FIELD SERVICE

Reputation Score from Work Order Outcomes

Technician completes mobile checklist on-site. Scores feed into Housing_Unit__c.Property_Score__c via Record-Triggered Flow. Score $\geq 85 \rightarrow$ NexGen_Certified__c = true. Badge on portal listing.

SANDBOX SETUP

Build Order

Do not follow the flow guide tab order for building. Follow this dependency order instead.



Custom Objects in Schema Builder

– Create all 10 custom objects and draw all relationships as lines on the canvas. Then go into Object Manager for each to add picklist values and formula fields.

Match__c, Housing_Unit__c, Companion_Hours__c, Monthly_Hours_Report__c, Invoice__c, Credit_Memo__c, Match_Proposal__c, Wellbeing_Check__c, Commission__c + extend Contact

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

2

Picklist Values

– Add all Status picklist values per object. Assign chart colors on Match__c.Status__c (green = Active, orange = At Risk, red = Suspended, grey = Ended).

Also add the Validation Rule on Match__c preventing Status = "Active" without Agreement_Signed__c = true.

3

Products & Pricebook

– Create Product2 records: Senior Host Platform Fee (\$149), Renter Access Fee (\$89), Property Placement Fee (15%), Premium Vetting (\$299). Activate Standard Pricebook.

All billing flows depend on Products existing first.

4

Test Data

– Create 3–4 Contact records manually: one Senior Host, two Young Renters, one Partner Account. Create one Housing_Unit__c record.

Flows need real records to work with during testing.

5

Application Intake Screen Flows

– Build the senior and renter intake flows first. These create Contacts which everything else references.

Best first demo flow – most interview-visible.

6

Lead Scoring Record-Triggered Flow

– Depends on Contact fields existing from Step 1.

7

Match Proposal Scheduled Flow

– Depends on Match__c object and both Contact record types. This is the matchmaking engine.

8

Hours Logging Screen Flow

– Depends on active Match__c records existing.

9

Monthly Compliance Scheduled Flow

– Depends on Companion_Hours__c records existing.

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

10

Billing & SLA Flows

– Depends on Invoice__c and Cases existing. Build last after all source data objects are populated.

11

Renewal Scheduled Flow

– References almost every other object. Build last.

12

Dashboard

– Pipeline by Stage, Churn Risk by Match, Open Cases by Priority. Connects automation to business outcomes for interview demos.

Interviewers love when candidates connect flows to reporting.

SHORTCUTS

Schema Builder & Shortcuts

Schema Builder – What it can do

✓ Can Build

Custom objects (drag onto canvas) · Custom fields · Lookup and Master-Detail relationships (draw as lines) · See all relationships at once

✗ Cannot Build

Picklist values · Validation rules · Page layouts · Record Types · Formula fields (field created but formula must be written in Object Manager)

Other Shortcuts

01

Salesforce CLI
metadata

– Generate boilerplate Apex class files from Terminal:

```
sf apex generate class --name StripePaymentService --output-dir force-app/main/default/classes
```

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

templates

02	Data Import Wizard	– Setup → Data Import Wizard. Bulk-load test records from CSV instead of creating one-by-one. Build Contacts, Housing Units, and Matches in a spreadsheet then import all at once.
03	Clone existing flows	– In Flow Builder: Save As → New Flow. Reuse structural logic between similar flows instead of starting from scratch.
04	Quick Actions into Screen Flows	– Object Manager → Buttons, Links, and Actions → New Action → Flow. Creates a one-click button on any record page that launches a Screen Flow directly.
05	AppExchange Flow Templates	– Search "flow templates" on AppExchange. Salesforce Labs publisher releases free declarative examples to study and modify.

DEVELOPER ENVIRONMENT

Dev Edition Setup

Key fact: A free Salesforce Developer Edition org includes Sales Cloud, Service Cloud, and Experience Cloud features. You do not need to choose – all three are available in the same org. Storage limit is 5MB which is fine for demo data.

Object naming issue found: Custom objects created in this org have double underscores in odd places – Film_Package_c_c, Invoice_c_c. This happened because the Object Name field had _c typed in it before Salesforce

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

appended its own __c. Consider renaming these in Object Manager for cleaner API names before building flows that reference them.

DEVELOPER TOOLS

VS Code & Salesforce CLI

Install VS Code

Download free from code.visualstudio.com – available for Mac, Windows, Linux.

After installing, open VS Code → Extensions icon → search "Salesforce Extension Pack" → Install. This is the official Salesforce toolset.

Install shell command

To enable the `code .` command in Terminal: Cmd+Shift+P in VS Code → type "Shell Command: Install 'code' command in PATH" → click it.

Salesforce CLI

Download from developer.salesforce.com/tools/salesforcecli. Use `sf` commands (not the older `sfdx`).

```
# Verify installation
sf --version
```

```
# Update when prompted
sf update --force
```

```
# Create a new project
```

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

```
sf project generate --name NexGenLiving
cd NexGenLiving
```

```
# Authorize your Developer Edition org
sf org login web --alias nexgen-dev
```

```
# List authorized orgs
sf org list
```

package.xml – use this version

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types><members>*</members><name>CustomObject</name></types>
  <types><members>*</members><name>Flow</name></types>
  <types><members>*</members><name>ApexClass</name></types>
  <types><members>*</members><name>Layout</name></types>
  <types><members>*</members><name>PermissionSet</name></types>
  <types><members>*</members><name>CustomTab</name></types>
  <version>62.0</version>
</Package>
```

Pull metadata from org

```
sf project retrieve start --manifest manifest/package.xml --target-org nexgen-dev
```

OVERVIEW

- Business Model

- Clouds Used

FLOWS

- Sales Cloud Flows

- Service Cloud Flows

- Experience Cloud

- Revenue & Billing

- Renewal & Churn

DATA MODEL

- Custom Objects

- Picklist Values

- Relationships

ARCHITECTURE

- Clever Patterns

- Build Order

- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup

- VS Code & CLI

GitHub Backup Setup

GitHub no longer accepts passwords. You need a Personal Access Token. Go to github.com → Settings → Developer settings → Personal access tokens → Tokens (classic) → Generate new token (classic). Check the "repo" checkbox. Copy the token immediately – it starts with ghp_

One-time Git configuration

```
git config --global user.email "your@email.com"
git config --global user.name "Your Name"
git config --global credential.helper osxkeychain
```

First-time push to GitHub

```
# From inside your project folder
git init
git remote add origin https://github.com/YOURUSERNAME/REPONAME.git

# Verify the URL is correct (not YOURUSERNAME)
git remote -v

# If URL is wrong, fix it
git remote set-url origin https://github.com/ericzprice/NexGenLiving.git

git add .
git commit -m "Initial backup"
git push -u origin main

# Use your Personal Access Token as the password
```

Ongoing backup – 3 commands

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

```
sf project retrieve start --manifest manifest/package.xml --target-org nexgen-dev
git add .
git commit -m "describe what you built today"
git push
```

Pro tip: Flows are stored as XML files after retrieval. Open them in VS Code to read the flow logic as code. You can diff changes between commits to see exactly what changed in a flow – worth mentioning in interviews.

AUTOMATION

One-Command Backup Script

Create the script

```
code ~/Documents/NexGenLiving/backup.sh
```

Paste this into the file:

```
#!/bin/zsh
echo "Pulling from Salesforce..."
sf project retrieve start --manifest manifest/package.xml --target-org nexgen-dev

echo "Staging changes..."
git add .

echo "Enter a commit message:"
read msg
```

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

```
git commit -m "$msg"

echo "Pushing to GitHub..."
git push

echo "Done. Backup complete."
```

Make it executable (run once)

```
chmod +x ~/Documents/NexGenLiving/backup.sh
```

Run it any time

```
# Navigate to your project first
cd ~/Documents/NexGenLiving
./backup.sh
```

Optional – Nightly cron backup: Run `crontab -e`, press i, paste the line below, press Escape then :wq to save.
Backs up automatically every night at 10pm.

```
0 22 * * * cd /Users/ericprice/Documents/NexGenLiving && sf project retrieve start --manif
```

OVERVIEW

- Business Model
- Clouds Used

FLOWS

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

— INTERVIEW PREPARATION

Opening Story

Always start with the business problem, not the technology.

*"NexGen Living is a PropTech startup I designed a full Salesforce architecture for. The business model is an intergenerational housing marketplace – seniors with a spare room are matched with young professionals or students who need affordable housing. It's been proven at scale in the Netherlands and Sweden, and the U.S. market is untapped. **The technical challenge is that it's a two-sided marketplace,** which means I had to build two completely different acquisition funnels, two distinct Experience Cloud portals, two separate Marketing Cloud journeys – and then have them converge at a matching engine. The matching engine itself is a Scheduled Flow that queries both pools, scores compatibility across eight criteria, and auto-creates ranked match proposals. No Apex. **The piece I'm most proud of is the contractual enforcement chain:** when a Service Cloud SLA is breached, a Record-Triggered Flow automatically calculates the financial penalty, creates a Credit Memo, and injects a negative line item into the next invoice through Revenue Cloud. The renter's contract enforces itself. That kind of cross-cloud data flow, done entirely declaratively, is what I think separates a Salesforce architect from someone who just builds flows."*

— INTERVIEW PREPARATION

Common Questions & Answers

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

"How did you handle matching without a custom algorithm?"



"What's the most creative declarative thing you built?"



"How does Revenue Cloud fit into this?"



"Tell me about your Experience Cloud work."



"What would you add if you had more time?"



INTERVIEW PREPARATION

Demo Tips

Start with the business problem

"The grooming staff was manually calling every owner the day before pickup..." Always open with pain, not technology. Interviewers need to see you

Build Flow A first for demos

The dual acquisition flow touches Lead, Contact, Opportunity, Products, and Cases – more interview-relevant topics in one demo than any

OVERVIEW

- Business Model
- Clouds Used

FLows

- Sales Cloud Flows
- Service Cloud Flows
- Experience Cloud
- Revenue & Billing
- Renewal & Churn

DATA MODEL

- Custom Objects
- Picklist Values
- Relationships

ARCHITECTURE

- Clever Patterns
- Build Order
- Schema Builder Tips

DEVELOPER SETUP

- Dev Edition Setup
- VS Code & CLI

understand why automation matters before you describe how it works.

Shorten Scheduled Paths in sandbox

Set Scheduled Path timers to +5 minutes in sandbox for live demos. Tell the interviewer "in production this would be 24 hours, but I've shortened it for the demo." Shows awareness of environments.

Connect automation to reporting

Build at least one Dashboard: Pipeline by Stage, Churn Risk by Match, Open Cases by Priority. Interviewers love when candidates connect automation to business outcomes.

other flow. Practice narrating each screen as you click through it.

Always gate flows with entry conditions

When asked about safety/correctness: "I always have a Decision element at the top of every flow checking record type, status, and conditions before any action fires." Shows defensive thinking.

Mock payment in sandbox

Skip live Stripe integration in demos – use a "Simulate Payment Approved" checkbox screen instead. Tell interviewer the prod version uses External Service. Shows pragmatism.