63LABS

Restauranteur Learn & Build

SECTION 05 OF 16

Facilities Equipment and Resources

Equipment and software registers, facilities records, resources, utilities and approved-service reminders.



SEPARATE TOPIC PDF / FULL RESTAURANT EDITION PRESERVED

Read this section

Two page-number systems

The section contents below use PDF page positions in this file. Original lesson footers keep their page numbers from the complete 619-page Restaurant Edition. "Master p/pp" still refers to the original 5,410-page Learn & Build source.

Same lessons, smaller files

Content has been separated, not rewritten. Nine shared-boundary pages were divided between their topics so every lesson keeps its ending. The complete Restaurant Edition and original master remain unchanged.

Start with the shared guide

The original scope guide is included below. Project sections also include the shared build starter and spreadsheet setup. These are learning and build specifications, not finished apps or certified operating instructions.

Moving between sections

Local contents and bookmarks open pages in this PDF. Cross-section links use the companion filenames and may require all files in one folder and a supporting PDF reader. If a link does not open, use the numbered section list and the printed edition-page label. Section 01 retains the original full-edition contents and source index.

The 16-section collection

01 Start Here and Source Index

02 Business Planning and Money

03 Stock Suppliers and Food Records

04 Team Operations and Procedures

05 Facilities Equipment and Resources

06 Catering Guests and Delivery

07 Marketing Publishing and Websites

08 Restaurant Products and Services

09 Careers and Team Learning

10 AI Tools and Task Guides

11 Operations Methods and Data

12 Evidence and Reasoning Cases

13 Integrated Restaurant Projects

14 Guest Clubs and Event Ideas

15 Supplier and Product Ideas

16 Concepts and Quality Checks

Contents for this PDF

Numbers on the right are PDF page positions in this section. Retained lesson footers show their complete-edition page labels.

How to use this edition (shared guide) 4

Shared build starter (shared guide) 5

Spreadsheet and document setup (shared guide) 6

Facilities, equipment and resources 7

P034 Restaurant equipment register 8

P039 Restaurant software and recurring-service renewals 10

P082 Restaurant document, agreement, manual, and renewal index 12

P067 Restaurant housekeeping task and recurrence record 14

P068 Restaurant premises and equipment-service reminder register 16

P069 Restaurant fixtures, smallwares, furniture, and event-equipment inventory 19

P070 Storeroom or service-station organization plan 21

P071 Restaurant relocation or pop-up move checklist 23

P077 Restaurant delivery-vehicle service history and reminder record 25

P089 Restaurant emergency-contact and communications binder 27

P094 Restaurant vendor, venue, menu-project, or workflow decision record 29

P095 Restaurant fit-out or renovation quote-and-change register 31

P096 Restaurant herb-garden, patio planting, or edible-growing observation record 33

P097 Restaurant uniform, apron, and event-outfit readiness index 35

P098 Restaurant utility usage and bill-normalization workbook 37

How to use this edition

This is a restaurant-focused curation of the actual 5,410-page 63Labs Learn & Build combined master. It keeps useful original lessons, projects, worked reasoning and tool guidance, then adds clearly labeled restaurant applications. The original file remains unchanged.

A reference library rather than a course

Start with the problem you are solving. Operators can use the money, stock, supplier, team and guest workflows; restaurant-industry builders can use the service, software, content and commercial project material. You do not need to implement every project or buy every named tool.

Broad restaurant use, not just daily operations

The edition retains all 50 original business projects and 43 adaptable planning, facilities, food-record, event and learning projects. Optional supplier, catering, career, venue-event and creative applications are identified as such. Private health, medication, sleep, fitness, mood and veterinary-care projects are not converted into employee-management tools.

What was changed

Repeated setup instructions and repeated practice layouts are consolidated. Restaurant application paragraphs are new adaptations. Source lessons and project requirements are identified by their original master page numbers. Necessary source corrections are labeled rather than silently presented as original text.

What you receive

A learning and build-specification PDF. It does not include generated apps, connected POS systems, live forms, installed automations, completed spreadsheets, legal policies, or a restaurant-specific food-safety program. The projects explain what to build and how to test it.

Source references

Master p or pp always means the physical page position in 63Labs_Combined_Ready.pdf, not a page in this edition. P001-style labels preserve the original project IDs. The contents and PDF bookmarks navigate this edition; the final index traces its source material.

Keep consequential decisions reviewed

Use fictional data while building. Keep private guest, staff and payment data out of shared examples. Use approved restaurant procedures and current authoritative guidance for food safety, employment, taxes, accessibility, privacy and other regulated matters. This edition does not certify compliance or safe equipment work.

Shared build starter

Use this once per build, followed by either the spreadsheet setup or document setup, then the chosen project specification. These consolidated instructions preserve the common safeguards from the source rather than repeating them in every project.

Copy this starter

Act as a patient build partner. Ask the project intake questions in one message, then wait for my answers. I may answer not sure. Separate my facts, goals, assumptions and research still needed. Never invent business facts, sources, permissions, credentials, customers, results or regulatory requirements. Use fictional sample records. Produce the complete requested artifact, not merely an outline. Explain how to copy, set up, use, test and revise it. Do not claim you created or saved a file unless you actually did. If long, stop at a section boundary and resume when I say continue without omissions. End with a requirement-by-requirement checklist. Do not connect services, publish, send messages, spend money or change live records as a side effect of drafting this build.

Add my restaurant context

My restaurant or industry role: [role]. Intended user: [person or role]. Exact task: [task]. Scope and exclusions: [limits]. Approved sources: [documents and dates]. Tools and access available: [tools]. Currency, locale and units: [values]. Owner and reviewer: [roles]. Deadline or work budget: [limit]. Ask about missing information rather than assuming it.

If a response or formula fails

Continue from the exact point where you stopped; keep the same structure, tab names, column order and references. For a formula issue, inspect the actual formula, cell, expected result, source values, locale, dates, units and copied ranges. Explain the proposed correction and rerun the original test. A polished answer is not proof of correctness.

Spreadsheet and document setup

Spreadsheet setup

For Google Sheets, confirm locale and units. The source examples use English functions, comma separators, real dates displayed YYYY-MM-DD, headers in row 1 and data rows 2:1000. Preserve named tabs and column order. Supply actual tab-separated headers, fictional sample rows and exact paste cells. Leave computed sample cells empty. Give each formula with its exact cell, explanation, fill range and expected result. Keep unused rows blank and distinguish missing inputs from zero. Specify validation, dropdowns, formats, filters, frozen headers and formula protection. Keep raw data tables unmerged. Do not add scripts, paid add-ons or external connections by default. Explain how every related range extends beyond row 1000.

Document setup

Create copy-ready pages with clear headings, useful tables only where appropriate, fill-in fields and a completed fictional example. Keep the reusable blank version separate from the example. Supply the actual wording and templates requested, not instructions to write them later. Name owners, actions, dates and review points. Use approved policy content; for changing rules, cite current primary sources only when verified, otherwise list a precise verification task.

First use

Create a safe copy. Enter the source example and compare every expected value. Test blank inputs, explicit zero, duplicate IDs, refunds/reversals, period boundaries and a changed value. For dates and percentages, inspect the underlying values as well as the display. Save evidence before replacing fictional records with authorized live information.

Limits of the supplied specifications

The arithmetic examples have been reviewed, but this PDF is not an executed application test. Generated formulas, data validation, access controls, links, integrations and regulatory treatment still need testing in the actual implementation. Editorial clarifications identify known source gaps, not a claim that all possible edge cases are covered.

Facilities, equipment and resources

Use the projects below independently. P numbers are the original master project IDs, retained for traceability. The short source lessons and worked reasoning are followed by project-specific build specifications.

P034 Restaurant equipment register

P039 Restaurant software and recurring-service renewals

P082 Restaurant document, agreement, manual, and renewal index

P067 Restaurant housekeeping task and recurrence record

P068 Restaurant premises and equipment-service reminder register

P069 Restaurant fixtures, smallwares, furniture, and event-equipment inventory

P070 Storeroom or service-station organization plan

P071 Restaurant relocation or pop-up move checklist

P077 Restaurant delivery-vehicle service history and reminder record

P089 Restaurant emergency-contact and communications binder

P094 Restaurant vendor, venue, menu-project, or workflow decision record

P095 Restaurant fit-out or renovation quote-and-change register

P096 Restaurant herb-garden, patio planting, or edible-growing observation record

P097 Restaurant uniform, apron, and event-outfit readiness index

P098 Restaurant utility usage and bill-normalization workbook

P034 Restaurant equipment register

Facilities, equipment and resources

Source project: Asset Management Project. Master pp 1045, 1251-1252, 3236. Restaurant application below is an editorial adaptation.

Restaurant application

Catalog durable equipment, manuals, custody/location, warranty and documented service history.

Source based lesson

Track individually owned business equipment, custody, service records, and disposal evidence.

Gather and build

Asset ID, serial number, location, custodian, purchase, service. Assign IDs; record transfers and service; retain disposal evidence without deleting the history.

Check before relying on it

A location change must not create a second asset. Trace one asset from purchase through its current custodian.

Source based practice with fictional inputs

Asset A-01 moves from storage to employee E-02.

Worked result: There is still one asset with an updated custodian and transfer history. Preserve the asset ID and record the movement.

Try a changed condition: A second identical-looking item arrives with a different serial number.

Check your answer: Create a separate asset ID; appearance alone does not establish duplication.

Restaurant boundary

Not a repair guide. Never invent maintenance intervals; original purchase cost is not present value or depreciation.

P034 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

Which durable assets are in scope? What tag format will identify them? Who holds each asset? What purchase and service records exist? Which review and return checks are needed?

Structure

Assets: AssetID, Description, SerialRef, PurchasedDate, Cost, Location, Custodian, Status, WarrantyEnd, ManualLink. Events: EventID, AssetID, Date, Type, From, To, Evidence. Service: ServiceID, AssetID, Date, Work, Cost, NextDue, Source.

Build Requirements

Treat assets as individual durable items, distinct from saleable inventory. Use In use/Available/In repair/Retired status. Never merge two similar restaurant appliances because their descriptions match. Record transfers, returns, service, and disposal with permanent EventIDs and evidence.

Create an issue/return checklist including condition, accessories, authorized recipient, date, and acknowledgement. Preserve history when custody changes; the Assets tab is current state and Events is the audit trail. Store sensitive serial numbers only in an appropriately restricted file if needed; examples are fictional.

Build views for equipment by custodian/location, warranty dates, service due, and missing tags. Provide exact COUNTIFS/SUMIFS formulas for active asset count and original purchase cost by category. Do not label original cost as current value or calculate tax depreciation without a separately specified method.

Test three assets costing 600, 900, and 300, with the 300 asset retired. Active asset count is 2 and original cost of active assets is 1,500; historical total purchase cost is 1,800. Manufacturer or approved service instructions determine intervals, not invented schedules. Add an annual physical-verification checklist and a reviewed retirement process.

Acceptance Check

Active count is 2 and active original cost is 1,500; retired items remain in history. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

D03: Append Category to Assets after ManualLink, preserving existing field order. Define active count as In use/Available/In repair, excluding Retired and incomplete status rows. Use Category and status for the requested summaries.

Source extensions and troubleshooting

Master p1253. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt fields for [equipment type] and relevant identifiers.

Change the workflow: Add a checkout and return workflow for shared equipment.

Add a useful view: Show assets with missing custodians, warranties ending, and overdue service.

Check the inputs. Do not confuse original purchase cost with current market or accounting value.

Check the workflow. Maintain transfer history when the current custodian changes.

Check the result. Use supplied manuals for maintenance intervals.

P039 Restaurant software and recurring-service renewals

Facilities, equipment and resources

Source project: Subscriptions Project. Master pp 1050, 1271-1272, 3296. Restaurant application below is an editorial adaptation.

Restaurant application

Track POS, ordering, website and other recurring business services and responsible owners.

Source based lesson

Track recurring business services, owners, billing, notice deadlines, and renewal decisions.

Gather and build

Service owner, billing amount, renewal date, notice deadline, use. Separate billing and cancellation deadlines; record actual usage; assign a keep-or-cancel decision.

Check before relying on it

A reminder date must fall before the decision deadline. Review the next three renewals and document the owner's decision.

Source based practice with fictional inputs

An annual service costs $240, renews June 30, and has a verified notice deadline of May 31.

Worked result: Show a $20 monthly equivalent but the actual $240 cash event; schedule the decision before May 31. Use the decision deadline, not only the billing date.

Try a changed condition: Cancellation is requested May 20 but unconfirmed.

Check your answer: Track the request and follow-up separately from confirmed cancellation.

Restaurant boundary

Monthly equivalent is not payment timing; the source does not cancel services or store credentials.

P039 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

Which services are recurring? What currency and billing frequencies apply? What renewal and notice dates are verified? Who uses and approves each service? What cost center and review horizon matter?

Structure

Subscriptions: SubscriptionID, Service, Owner, AmountPerBill, Frequency, NextBillDate, RenewalDate, NoticeDays, Decision, Status. Add UsageReview and Calendar.

Build Requirements

Frequency values Monthly/Quarterly/Annual/Other. Give normalized monthly cost as amount, amount/3, or amount/12 for the first three; Other requires a user-entered conversion and basis. This comparison is not the cash payment schedule. Keep actual invoice dates in a separate billing log.

Compute NoticeDeadline=RenewalDate-NoticeDays only when both are known. Unknown notice terms show Verify terms. Do not assume all agreements use calendar days; if business days or special clauses apply, adapt from the actual terms. Status Active/Trial/Cancelled/Expired; cancellation requires confirmation evidence and does not erase history.

UsageReview records seats purchased, seats used, last review, owner decision, and evidence. Show renewal/notice dates within a chosen horizon, unowned services, and unused capacity. Do not store credentials or cancel services automatically.

Test a 120 annual service has normalized monthly cost 10 but a 120 cash payment in its actual billing month. A renewal on November 30 with 30 calendar days' notice has an October 31 deadline. Provide a review checklist: confirm need, usage, cost, alternatives, notice terms, authorized decision, and confirmation.

Acceptance Check

Annual 120 normalizes to 10; notice deadlines use verified terms; inactive services are excluded from ongoing cost. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1273. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Add cost centers for [teams] and assign one owner per service.

Change the workflow: Adapt notice calculations to these verified contract terms: [paste].

Add a useful view: Show actual billing cash flow beside normalized monthly cost.

Check the inputs. Do not substitute monthly equivalent cost for actual payment timing.

Check the workflow. Verify notice terms rather than assuming a fixed number of days.

Check the result. Retain cancellation confirmation and final billing records.

P082 Restaurant document, agreement, manual, and renewal index

Facilities, equipment and resources

Source project: Personal Records Project. Master pp 1093, 1443-1444, 3812. Restaurant application below is an editorial adaptation.

Restaurant application

Map personal categories to actual restaurant licenses/agreements/training records/manuals as appropriate; retain secure storage references, owners, expiry status, access groups, and last verification.

Source based lesson (restaurant wording adapted)

Create a searchable index for restaurant agreements, manuals, permits, approved training records, and other business documents without copying restricted contents.

Gather and build

Document category, secure location, owner, expiry, access method. Index locations and renewal dates; restrict access; avoid storing secret credentials in a general-purpose sheet.

Check before relying on it

A public link must not expose private records. Locate three documents using only your index and repair unclear labels.

Source based practice with fictional inputs

A restaurant document index lists a document category and authorized storage location without reproducing restricted contents.

Worked result: The index helps retrieval while limiting copied information. Use location references and appropriate access methods rather than storing secrets in a general index.

Try a changed condition: A location link is inaccessible to its intended authorized user.

Check your answer: Repair the retrieval process; do not make the whole index public as a shortcut.

Restaurant boundary

An index is not a security implementation or compliance determination. Do not invent retention periods or embed credentials/sensitive identifiers.

P082 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant agreements, manuals, permits, training documents, or other records need an index; where originals/copies are stored; authorized owners/access groups; verified issue/expiry/review dates; naming convention; and backup method. Index references only, not secrets or unnecessary personal information.

Structure

Folder map; document index; naming convention; access matrix; renewal review; backup check; emergency access instructions.

Build Requirements

Build categories chosen by the user, such as agreements, manuals, permits, approved training documents, equipment records, and restaurant administration. The index stores references, not passwords, full account numbers, or document images. Use fictional examples when asking an AI for help.

Index fields: DocumentID, Category, Description, Owner, IssueDate, ExpiryDate, StorageRef, OriginalOrCopy, AccessGroup, LastVerified, NextAction. Missing expiry can mean Not applicable or Unknown; distinguish them. A storage path is not proof that a file still exists, so include a verification date.

Create a naming convention with date, category, description, and version, plus a small folder tree described in a table. Keep sensitive filenames discreet if shared. Define who may view, edit, and manage access using the user's chosen tools; do not claim encryption or security features without verification.

Do not invent legal retention periods. Add a retention-decision field based on current official guidance or the user's qualified adviser, with source and checked date. Provide a backup test checklist: locate a sample, open it, confirm readability, and verify authorized access.

Example (restaurant adaptation): an index entry points to a supplier agreement in authorized storage and lists its review date, while a team view shows only the document type, responsible owner, and authorized retrieval reference. Finish with a quarterly review and an emergency-access note without embedding secrets.

Acceptance Check

The index, folder map, access decisions, renewal review, and backup test are complete without embedded secrets. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1445. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the categories and access groups to [restaurant record and access needs].

Change the workflow: Create a migration checklist from [current system] without losing originals or version history.

Add a useful view: Add an expiry review view that distinguishes Unknown from Not applicable.

Check the inputs. Keep credentials and full sensitive identifiers out of the index.

Check the workflow. Verify files and backups instead of assuming a link proves availability.

Check the result. Use sourced retention decisions rather than invented universal periods.

P067 Restaurant housekeeping task and recurrence record

Facilities, equipment and resources

Source project: Home Cleaning Project. Master pp 1078, 1383-1384, 3632. Restaurant application below is an editorial adaptation.

Restaurant application

Replace home rooms with defined restaurant areas; use owner-approved task standards, product references, owners, due occurrences, and completion evidence.

Source based lesson (restaurant wording adapted)

Record restaurant housekeeping tasks, accountable owners, approved frequencies, and occurrence completion without treating the planner as a sanitation program.

Gather and build

Restaurant areas, source-approved tasks and frequencies, trained roles, available time, governing procedures. Enter the actual approved frequencies and methods; assign responsibility; preserve each scheduled occurrence and its completion evidence.

Check before relying on it

Overdue tasks should stay visible rather than vanish at week-end. Time an approved routine, identify capacity problems, and escalate any mandatory work that cannot fit. Do not silently reduce required cleaning or sanitation.

Source based practice with fictional inputs

An approved restaurant housekeeping task is due Monday and remains incomplete when the next week begins.

Worked result: The overdue occurrence remains visible. Preserve occurrence status and create the next occurrence separately.

Try a changed condition: An authorized manager explicitly cancels the old occurrence under the applicable procedure and records the reason.

Check your answer: Record the cancellation and reason, not completion. If the task is mandatory, its requirement must still be resolved through the applicable approved procedure.

Restaurant boundary

The household schedule is not a commercial sanitation program. Do not invent chemical mixing, contact times, food-contact procedures, or required frequencies; those need authoritative approved sources.

P067 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant areas and approved housekeeping tasks are in scope; trained/authorized participants; source-approved frequencies and methods; available time; access needs; product labels; and manufacturer or commercial-sanitation procedures that govern the work.

Structure

Task list; weekly/monthly schedule; room checklist; supply list; quick-reset option; completion log.

Build Requirements

Build a room-by-room task inventory and record frequencies from approved restaurant procedures and applicable source guidance. Separate routine cleaning from repairs and maintenance. Include estimated minutes, owner, required supplies, and a completion standard that is specific without demanding perfection.

Create a weekly schedule within the available time and a separate rotating list for less frequent tasks. Provide an optional 10-minute reset only for approved nonmandatory tasks and a way to request authorized deferral. Do not omit or shorten mandatory cleaning, sanitation or verification steps to fit a time budget. Do not mix cleaning chemicals or invent surface-treatment instructions; follow product labels and manufacturer guidance.

Completion log fields: TaskID, PlannedDate, Owner, Status, CompletedDate, Notes. Status Planned/Done/Skipped/Rescheduled/Cancelled. A cancellation requires reason and authorized approval; it does not erase an unresolved mandatory requirement. Keep a rescheduled task linked to its original occurrence rather than counting it twice. Add a supply checklist with item, on-hand state, and buy-next flag.

Test a 60-minute weekly budget with tasks estimated at 15, 20, 10, and 15 minutes: total 60. Adding a 20-minute task requires additional capacity, authorized rescheduling of eligible work, or sharing with an appropriately trained person. Mandatory work still has to be resolved under the approved procedure. Provide a short team workload-review script for adjusting responsibilities without assigning work by stereotypes.

Acceptance Check

The example schedule totals 60 minutes and includes realistic owners, supplies, and skip/reschedule states. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

E10: Add Cancelled and reason/authorized decision reference. Cancellation is not completion; mandatory requirements must still be resolved under the governing procedure.

Source extensions and troubleshooting

Master p1385. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the schedule to [rooms] and [minutes/week].

Change the workflow: Create a reduced optional-work version while preserving all mandatory approved tasks. If those tasks do not fit the available resources, flag the capacity problem and required management decision.

Add a useful view: Add a shared responsibility view with agreed owners and task definitions.

Check the inputs. Keep total scheduled effort within available time.

Check the workflow. Use product and surface instructions instead of improvised chemical advice.

Check the result. Record skipped or rescheduled work distinctly from Done.

P068 Restaurant premises and equipment-service reminder register

Facilities, equipment and resources

Source project: Home Maintenance Project. Master pp 1079, 1387-1388, 3644. Restaurant application below is an editorial adaptation.

Restaurant application

Map systems to identified restaurant assets or premises services; retain manuals, responsible person, last service, sourced interval, provider, and history.

Source based lesson (restaurant wording adapted)

Track restaurant premises systems, equipment maintenance sources, service history, and verified upcoming reminders.

Gather and build

System, model, manual reference, last service, interval, provider. Use manufacturer or qualified-service guidance; record completed work; calculate reminders from the stated interval.

Check before relying on it

Do not invent a maintenance interval because a generic template contains one. Collect documentation for one system before scheduling its tasks.

Source based practice with fictional inputs

A restaurant equipment-maintenance template contains an interval but no applicable manufacturer manual or qualified-provider source.

Worked result: The interval is unverified and should not drive a confident schedule. Record the applicable source and confirm the interval before relying on it.

Try a changed condition: The source uses both elapsed time and usage.

Check your answer: Capture both conditions and the actual rule; do not replace it with a guessed date-only interval.

Restaurant boundary

No hazardous repair steps or invented intervals. Property/landlord responsibilities and time-versus-usage rules must be verified.

P068 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant premises systems and equipment are in scope; who owns, leases, or services each; applicable manuals and verified tasks/intervals; previous service records; responsible role; budget; and qualified-provider contacts.

Structure

Systems: SystemID, Name, Location, ManualLink, WarrantyEnd, ResponsiblePerson. Tasks: TaskID, SystemID, Task, Source, LastDone, IntervalMonths, NextDue, Owner, Status, NextDateOverride, ScheduleBasis. Add ServiceLog and Costs.

Build Requirements

Use manufacturer, installer, landlord, or qualified-provider instructions for task frequency. If no source is available, mark Verify interval rather than inventing a schedule. Separate ordinary reminders from work requiring qualified help; do not give hazardous repair steps.

A verified real NextDateOverride takes precedence only with its ScheduleBasis and applicable Source recorded. Otherwise NextDue uses EDATE(LastDone,IntervalMonths) when both inputs are present and the interval is positive; otherwise show Set schedule. Preserve the entered override separately from the calculated NextDue. For seasonal or condition-based work, retain the actual source conditions rather than substituting an invented date. Do not automatically mark a task completed when its due date passes.

ServiceLog fields: ServiceID, SystemID, Date, Work, Provider, Cost, ReceiptLink, NextRecommendation. Preserve prior service history. Summary shows due soon, overdue, unknown schedules, and actual costs by period. Keep planned budgets separate from spending.

Test a task last completed January 15 with a three-month interval gives April 15. A missing LastDone is not interpreted as an ancient date. Provide a seasonal review checklist based on user-supplied needs, a contractor question list, and a record of operator versus property-owner responsibility. Emergency issues are handled through appropriate local services, not this reminder planner.

Acceptance Check

The sample due date is April 15; without a verified source-backed override, missing dates show Set schedule; service sources and history are retained. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

E04: Append NextDateOverride and ScheduleBasis/SourceRef, keep computed date separate, and retain actual source/manual conditions.

Source extensions and troubleshooting

Master p1389. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the planner to [property type] using these manuals and responsibilities: [list].

Change the workflow: Add a yearly maintenance budget with planned and actual costs separated.

Add a useful view: Create a contractor/service contact page with scope and receipt references.

Check the inputs. Do not invent intervals when the source is missing.

Check the workflow. Use real dates and explicit interval units.

Check the result. Keep completed service records even when the next due date changes.

P069 Restaurant fixtures, smallwares, furniture, and event-equipment inventory

Facilities, equipment and resources

Source project: Household Assets Project. Master pp 1080, 1391-1392, 3656. Restaurant application below is an editorial adaptation.

Restaurant application

Replace rooms with restaurant locations; preserve record IDs, quantities, evidence, value basis, and unvalued-item counts.

Source based lesson (restaurant wording adapted)

Catalog restaurant fixtures, smallwares, furniture, or event equipment using stable IDs, locations, evidence, quantities, and a stated value basis.

Gather and build

Item, location, serial number, photo, receipt, user-supplied value. Create a room-by-room inventory; link evidence securely; distinguish purchase cost from estimated current value.

Check before relying on it

Duplicate photos do not mean duplicate possessions. Verify five entries against the actual items and correct missing details.

Source based practice with fictional inputs

Two photographs show the same restaurant-owned laptop with one serial number.

Worked result: The inventory contains one item with two images. Link evidence to a stable item identifier.

Try a changed condition: A second laptop has the same model but a different serial number.

Check your answer: Create a separate item record.

Restaurant boundary

Useful distinction from individual Asset Management: grouped physical quantities and separate value bases. Not appraisal, insurance coverage, or claim guarantee.

P069 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant rooms, fixtures, smallwares, furniture, or event equipment are included; inventory purpose; chosen value basis; verified quantities; available evidence; and the authorized storage/access/backup arrangement.

Structure

Items: ItemID, Room, Category, Description, Quantity, UnitValue, ValueBasis, PurchaseDate, PhotoRef, ReceiptRef, SerialRef, Notes. Add Summary and ReviewLog.

Build Requirements

Use permanent ItemIDs and separate similar items when their identity or evidence matters. UnitValue is user supplied and labeled Purchase cost/Estimated replacement/Other; do not sum unlike bases into one unlabeled total. Line value=Quantity*UnitValue only when both are known; missing value remains unknown, not zero.

Create a room-by-room capture checklist with description, model if known, location, photo, receipt, and condition note. Avoid asking an AI to hold sensitive serial numbers or private documents; references can point to the user's chosen secure location. This inventory does not establish insurance coverage or guarantee a claim outcome.

Summary shows item count, quantity, known value by basis, and number of unvalued records. Count records and physical quantity separately. Test two chairs at 80 each and one lamp at unknown value: known value 160, three physical items, two inventory records, one unvalued record.

Include an update process for additions, moves, gifts, disposal, and corrections, keeping dated evidence where useful. Provide a backup and export checklist and an annual room review. Do not generate appraisal values without an identified basis.

Acceptance Check

The sample shows 160 known value, three items, two records, and one unvalued record. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1393. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt fields for [collections/electronics/furniture] with appropriate evidence references.

Change the workflow: Add a move-location log without duplicating the item itself.

Add a useful view: Show known values separately by purchase cost and estimated replacement basis.

Check the inputs. Do not turn missing values into zero.

Check the workflow. Keep record count separate from physical quantity.

Check the result. Label value basis and avoid implying coverage or appraisal certainty.

P070 Storeroom or service-station organization plan

Facilities, equipment and resources

Source project: Home Organization Project. Master pp 1081, 1395-1396, 3668. Restaurant application below is an editorial adaptation.

Restaurant application

Map small home spaces to restaurant storage areas; use ownership, keep/move/review categories, location labels, outgoing-item log, and completion checks.

Source based lesson (restaurant wording adapted)

Plan bounded restaurant storage or service-station sorting sessions and assign approved locations to retained items.

Gather and build

Defined work area, session limit, authorized decision-maker, item ownership, approved storage and disposal constraints. Choose a small area; sort into keep, relocate, authorized disposal, and review. Confirm authorization and applicable retention or storage requirements before acting.

Check before relying on it

Do not discard another owner's property or restaurant records without the appropriate authorization. Record the approved layout for authorized team use and schedule a maintenance check.

Source based practice with fictional inputs

A restaurant storage drawer contains business items and another owner's belongings.

Worked result: Sort or make decisions only within the agreed ownership scope. Obtain agreement and preserve an undecided/review category.

Try a changed condition: An undecided item has no review date.

Check your answer: Assign a review action rather than claiming the area is fully resolved.

Restaurant boundary

Do not discard another person's property or apply unsupported waste/retention rules. Food storage suitability and safety separation require approved guidance, not decluttering logic.

P070 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant storage or service-station area needs organizing; desired retrieval outcome; authorized decision-maker; time; item ownership; approved storage constraints; and verified disposal, retention, recycling, or relocation options.

Structure

Space priorities; decision guide; session plan; sorting labels; outgoing-item log; storage map; maintenance routine.

Build Requirements

Define one small target area per session and a visible stopping point. Create decision categories Keep here/Move/Donate/Sell/Recycle/Dispose/Unsure. Do not pressure the user to discard sentimental, shared, or legally important items. Shared belongings require agreement.

Write a decision guide based on usefulness, fit, condition, duplication, space, and documented operational purpose. Add an Unsure box with a review date rather than an indefinite hidden pile. Keep donation/sale tasks realistic; a decision is not complete until the item leaves or reaches its authorized storage location.

Session plan fields: SessionID, Area, Goal, Minutes, Supplies, DecisionLimit, ExitTask, ReviewDate. Outgoing log: ItemRef, Destination, Owner, NextAction, Deadline, CompletedDate. Special waste, records, and potentially hazardous items require current local disposal or retention guidance; do not invent it.

Create a storage map linking category to location, label, access needs, and capacity. Example: a 30-minute drawer session allocates 5 minutes setup, 15 sorting, 5 organizing, and 5 removing outgoing items. Add a five-minute reset routine and a review focused on whether belongings are easier to find.

Acceptance Check

The sample session totals 30 minutes and includes a complete exit path for outgoing items. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1397. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the plan to [space] and [session length].

Change the workflow: Create decision prompts for sentimental items without pressure to discard them.

Add a useful view: Add an outgoing-items dashboard with destination and next action.

Check the inputs. Do not sort more than you can reset within the session.

Check the workflow. Check ownership before discarding shared items.

Check the result. Verify disposal and record-retention requirements for special items.

P071 Restaurant relocation or pop-up move checklist

Facilities, equipment and resources

Source project: Moving Project. Master pp 1082, 1399-1400, 3680. Restaurant application below is an editorial adaptation.

Restaurant application

Map rooms and boxes to kitchen-office-service areas and equipment crates; retain stable IDs, load/arrival states, access, service confirmation, and deposits.

Source based lesson (restaurant wording adapted)

Coordinate a restaurant site move or pop-up relocation with dates, owners, packing IDs, confirmed services, access, and arrival tasks.

Gather and build

Move dates, addresses, contacts, box IDs, services, responsibilities. Work backward from move day; label boxes by destination; track critical documents and arrival essentials.

Check before relying on it

Packing complete does not mean transport, access, or utilities are confirmed. Rehearse the first planned setup or service at the destination and identify missing equipment, access arrangements, or unresolved dependencies.

Source based practice with fictional inputs

Restaurant relocation boxes K-01 and K-02 have contents lists; the transport booking is still proposed.

Worked result: Packing status can be complete while transport remains unconfirmed. Track packing, access, transport, and essential services separately.

Try a changed condition: K-02 is relabeled B-04 without updating its record.

Check your answer: Preserve or map the identifier change so the contents remain traceable.

Restaurant boundary

Not a safe food/equipment transport or utility commissioning plan. Use qualified providers and verified property requirements for specialist work.

P071 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant site move or pop-up relocation is planned; confirmed dates; origin/destination; people and equipment; verified access and transport arrangements; critical service dependencies; accessibility needs; budget; and actual notice terms.

Structure

Move timeline; task register; box inventory; service/address checklist; essentials bag; budget; move-day plan; arrival checks.

Build Requirements

Build a backward timeline from the confirmed move date with booking, notices, sorting, supplies, packing, utilities, travel, handover, and unpacking tasks. Mark unconfirmed dates and responsibilities. Verify lease/provider notice requirements from the actual documents rather than inventing standard periods.

Box inventory fields: BoxID, OriginRoom, DestinationRoom, ContentsSummary, Fragile, Priority, Loaded, Arrived. Avoid listing sensitive documents in an openly shared box file; keep an essentials and important-records plan separately. Label boxes consistently and record missing/damaged items without overwriting packing details.

Budget distinguishes estimates, quotes, deposits, final charges, and paid amounts. Do not add a deposit twice to a total quote. Example: a quoted service costs 600 and a 150 deposit is paid; remaining 450, total cost still 600.

Create a move-day run sheet with times, contacts, access, parking/booking checks, meter/key handover if relevant, breaks, and fallback contact. Address changes should be performed only through verified official channels; never ask for passwords in the tracker. Include arrival essentials, condition photos where appropriate, and a seven-day unpacking priority plan.

Acceptance Check

The budget example leaves 450 due; boxes have unique IDs; move-day dependencies and unconfirmed tasks are visible. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1401. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the plan to [local/long-distance] and these restaurant site and team requirements: [list].

Change the workflow: Create a room-by-room packing schedule within [days/hours].

Add a useful view: Add a quote/deposit/final-payment reconciliation table.

Check the inputs. Distinguish confirmed bookings from requested quotes.

Check the workflow. Avoid double counting deposits.

Check the result. Keep essentials, keys, and critical documents accessible.

P077 Restaurant delivery-vehicle service history and reminder record

Facilities, equipment and resources

Source project: Vehicle Care Project. Master pp 1088, 1423-1424, 3752. Restaurant application below is an editorial adaptation.

Restaurant application

Use actual restaurant/catering vehicle IDs, units, mileage, manuals, verified time/distance intervals, and provider invoices.

Source based lesson (restaurant wording adapted)

Keep restaurant delivery or catering vehicle records with verified service details, mileage or kilometers, source-based reminders, and unresolved evidence gaps.

Gather and build

Vehicle identity, mileage, service records, official schedule. Record date and mileage for each service; link invoices; plan next tasks using the exact vehicle's guidance.

Check before relying on it

A maintenance reminder is not a diagnosis of a warning light. Build a history for one vehicle and list records you still need.

Source based practice with fictional inputs

A restaurant delivery-vehicle record says serviced but has no date, odometer reading, or work details.

Worked result: The event's scope and timing remain uncertain. Request supporting records and describe the gap accurately.

Try a changed condition: An invoice confirms one specific task at a known mileage.

Check your answer: Update only that task; do not infer unrelated work was completed.

Restaurant boundary

Not diagnosis, hazardous repair training, or universal maintenance schedule. Time/distance basis must come from the exact vehicle's guidance.

P077 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant delivery/catering vehicles are included; miles or kilometers; exact vehicle identifiers; verified manuals/service schedules; last service dates/readings; current readings; and provider/invoice records.

Structure

Vehicles: VehicleID, Name, DistanceUnit, CurrentOdometer, ManualLink. Tasks: TaskID, VehicleID, Task, LastDate, LastOdometer, MonthsInterval, DistanceInterval, NextDate, NextOdometer, Status. Add ServiceLog.

Build Requirements

Use manufacturer or qualified-provider schedules and label their source. Do not invent universal service intervals. Some tasks are due by time or distance, whichever comes first; others follow condition or a provider instruction. Record the chosen basis explicitly.

NextDate=EDATE(LastDate,MonthsInterval) when known and positive; NextOdometer=LastOdometer+DistanceInterval when known and positive. Flag due if either applicable threshold is reached. Unknown thresholds remain Verify schedule, not overdue from a zero date. Keep miles and kilometers separate unless a conversion is explicitly defined.

ServiceLog includes ServiceID, VehicleID, Date, Odometer, Work, Provider, Cost, ReceiptRef, Recommendation. Preserve history and validate nondecreasing odometer readings unless a documented replacement/correction explains the change. Do not provide hazardous repair instructions.

Test a verified 5,000-distance-unit interval after odometer 20,000 gives next 25,000. At 25,100 it is due even if the time threshold has not arrived. Add cost summaries by vehicle and period, a pre-service question list, and a record of recalls or notices only when verified through official sources.

Acceptance Check

The sample threshold is 25,000 and triggers at 25,100; missing source intervals remain unconfirmed. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1425. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt schedules to these manual instructions: [paste].

Change the workflow: Add multiple vehicles with separate distance units and histories.

Add a useful view: Show upcoming time-based and distance-based tasks without combining unlike units.

Check the inputs. Use the actual manual rather than generic intervals.

Check the workflow. Check distance units before comparing readings.

Check the result. Retain service history and explain any odometer correction.

P089 Restaurant emergency-contact and communications binder

Facilities, equipment and resources

Source project: Emergency Planning Project. Master pp 1100, 1471-1472, 3896. Restaurant application below is an editorial adaptation.

Restaurant application

Map household participants to actual site roles and approved contacts; preserve verified official guidance, accessible copies, offline contact plan, drill, and correction log.

Source based lesson (restaurant wording adapted)

Organize restaurant-site communication contacts, accessible plan copies, approved supply checks, and current site-specific emergency guidance.

Gather and build

Verified site contacts, approved meeting arrangements, chosen operational accessibility needs, official alerts, approved supply records. Use applicable current official and site-specific guidance; record the authorized plan owner and communication options; review changes and practice only as directed by the approved plan.

Check before relying on it

The plan is incomplete if an intended participant cannot access or understand it. A general template does not validate a venue's emergency procedures. Ask the authorized site plan owner to verify current local requirements and emergency-management guidance for this restaurant; record the source and review date.

Source based practice with fictional inputs

A restaurant-site communication plan is stored on a device one intended participant cannot access.

Worked result: The plan is not usable by every intended participant. Arrange accessible, understood plan copies and check the communication workflow against current approved site guidance.

Try a changed condition: A contact number changes.

Check your answer: Update the relevant copies and check that participants know the current information.

Restaurant boundary

Not an evacuation, fire-code, food-emergency, hazardous-response, or supply-prescription authority. Adopt site-specific official guidance and qualified approvals; do not invent routes or shelter locations.

P089 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant site and hazards are covered by actual official/site-specific guidance; authorized plan owner; staff/guest communication and accessibility requirements; verified contacts/meeting arrangements; existing approved supplies; and current source references. Do not request clinical histories or invent emergency routes.

Structure

Restaurant-site profile; contact/communication plan; meeting points; official guidance index; supply inventory; document references; practice log; review checklist.

Build Requirements

Use current official local emergency guidance. In the United States, start with https://www.ready.gov/ and local emergency management; outside the U.S., use the appropriate official agencies. Do not invent evacuation routes, shelter locations, hazard-specific instructions, or supply quantities. Record source and checked date for each adopted instruction.

Create a communication plan with primary and alternate contacts, an out-of-area contact if chosen, agreed meeting points, and a method for checking in. Verify meeting locations and accessibility with the authorized site plan owner and participants. Keep an offline copy where participants can access it.

Supply inventory fields: ItemID, Item, Unit, RecommendedQtyFromSource, OnHand, Gap, ExpiryOrCheckDate, Storage, Owner. Gap=MAX(0,Recommended-OnHand) only when a verified target exists. Record only operational accessibility and communication arrangements the participants choose to provide; keep clinical details and animal-care instructions outside this shared plan.

Create an important-records reference page without embedding passwords or full sensitive identifiers. Add a plan for maintaining charged communication devices and checking supplies based on manufacturer guidance. If an actual emergency occurs, follow official alerts and emergency services rather than this generic binder.

Run a simple communication drill: one participating staff member cannot reach the primary contact, so they try the agreed alternate and meeting procedure. Record what failed and assign a correction. End with review after moves, site or team changes, and updated official guidance.

Acceptance Check

Contacts and meeting points are verified, supply targets have sources, and a drill with corrective actions is included. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1473. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the binder to [location] using these current official sources: [links/text].

Change the workflow: Create an accessible communication plan for these approved site communication needs: [list].

Add a useful view: Add a supply-check view with verified target quantities and expiry/review dates.

Check the inputs. Use locally relevant official instructions, not invented routes or universal quantities.

Check the workflow. Test contact details and offline access.

Check the result. Keep clinical details out of the shared plan; use only approved operational accessibility arrangements.

P094 Restaurant vendor, venue, menu-project, or workflow decision record

Facilities, equipment and resources

Source project: Decision Making Project. Master pp 1105, 1491-1492, 3956. Restaurant application below is an editorial adaptation.

Restaurant application

Use real options, weighted criteria, hard constraints, evidence, tradeoffs, owner decision, and revisit trigger.

Source based lesson (restaurant wording adapted)

Document a restaurant operational decision with real alternatives, owner-defined criteria, hard constraints, evidence, and a review trigger.

Gather and build

Decision, priorities, options, evidence, tradeoffs, hard constraints, decision owner, review date. Reject infeasible options before comparing the rest against the same stated criteria; record the decision-maker's reason and what new evidence would change it.

Check before relying on it

An attractive score must not hide a non-negotiable constraint. Revisit a decision after new evidence and explain whether your reasoning changed.

Source based practice with fictional inputs

Restaurant option A scores higher overall but violates a stated non-negotiable delivery schedule.

Worked result: It is infeasible under the current constraint despite the score. Apply feasibility constraints before ranking remaining options.

Try a changed condition: The schedule limit is genuinely changed by the decision-maker.

Check your answer: Reassess feasibility and scores with the updated constraint documented.

Restaurant boundary

A high score cannot override safety/legal/capacity constraints; not a substitute for qualified high-stakes advice.

P094 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant decision is in scope; real alternatives; documented hard constraints; owner-defined criteria; available evidence; unresolved uncertainty; decision deadline; and review trigger. Avoid unnecessary personal disclosures.

Structure

Values reflection; decision brief; facts/assumptions table; options matrix; tradeoff notes; decision record; review page.

Build Requirements

Help the user describe values through actual choices and behaviors rather than supplying a supposedly correct set. Create prompts about meaningful experiences, obligations, boundaries, and tradeoffs. Do not request unrelated personal or clinical details.

Decision brief fields: DecisionID, Question, Deadline, Options, Constraints, Evidence, Unknowns. Build a criteria matrix with user-selected weights and 1-5 scores anchored to concrete meaning. Require weights to total 100% and keep missing scores incomplete. A weighted score is a thinking aid, not an instruction to choose the highest number.

Test option A scores 4 and 2 on equally weighted criteria: weighted score 3. Option B scores 3 and 4: 3.5. A nonnegotiable constraint can still rule out B; show constraints before rankings. Do not use this generic framework as a substitute for qualified advice in high-stakes legal, medical, or financial decisions.

Write a decision record with chosen option, reasons, tradeoffs accepted, uncertainty, next action, and revisit trigger. The review asks what was known at the time and what was learned, avoiding hindsight-based self-blame. Include a small reversible experiment when appropriate and chosen by the user.

Acceptance Check

The sample scores 3 and 3.5, weights reconcile, and constraints and uncertainty remain visible. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1493. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the criteria to these values and constraints: [list].

Change the workflow: Create a small reversible experiment to test [uncertain assumption].

Add a useful view: Add a decision-review page separating outcome from the quality of information available at the time.

Check the inputs. Do not let numerical scores override nonnegotiable constraints.

Check the workflow. Keep assumptions distinct from evidence.

Check the result. Use the framework to clarify your judgment rather than outsource the decision owner's stated priorities.

P095 Restaurant fit-out or renovation quote-and-change register

Facilities, equipment and resources

Source project: Home Projects Project. Master pp 1106, 1495-1496, 3968. Restaurant application below is an editorial adaptation.

Restaurant application

Map home work to restaurant premises scope; retain equivalent quote comparison, verified measurements, approvals, dependencies, changes, evidence, and closeout.

Source based lesson (restaurant wording adapted)

Record restaurant renovation or fit-out scope, comparable quotes, dependencies, authorized changes, and acceptance checks without supplying hazardous work instructions.

Gather and build

Scope, measurements, quotes, dependencies, budget, acceptance criteria. Define the desired result; compare equivalent quotes; identify permits or licensed work to verify locally.

Check before relying on it

Do not treat a low quote with missing work as automatically cheaper. Write a completion checklist before approving work.

Source based practice with fictional inputs

For a restaurant redecoration project, quote A includes preparation, paint, and cleanup; quote B includes paint application only.

Worked result: The prices do not yet cover equivalent scopes. Compare the same work, materials, and exclusions.

Try a changed condition: B adds the missing tasks and remains lower in price.

Check your answer: Then compare timing, quality evidence, terms, and applicable qualifications as well as price.

Restaurant boundary

No structural/electrical/gas DIY instructions, assumed permit exemptions, or automatic contract/payment authorization. Commercial requirements need current qualified verification.

P095 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant fit-out/renovation scope and finish criteria are proposed; verified measurements/conditions; budget/dates; required qualified assessments and local/property approvals; comparable quotes; and the authorized change approver.

Structure

Project brief; requirements; quote comparison; work sequence; selections; budget/change log; acceptance checklist; closeout records.

Build Requirements

Define included work, exclusions, finish criteria, and what needs professional assessment. Do not provide structural, electrical, gas, or other hazardous DIY instructions. Required permits and approvals must be checked with current local authorities and property rules; do not assume they are unnecessary.

Create a quote comparison with scope, materials, exclusions, schedule, payment milestones, warranty terms as quoted, and unresolved questions. Lowest price is not automatically equivalent scope. Separate estimates, accepted contract amount, approved changes, paid, and remaining balance.

Build a dependency schedule: assessment, approvals, design/selections, ordering, preparation, work, inspection, punch list, handover. Mark lead times as supplier-confirmed or assumed. Include service disruption, approved temporary operating arrangements, dust/noise controls to verify, access, and protection of restaurant assets where relevant; do not invent safe-operation procedures.

Test accepted scope 5,000 plus approved change 400 gives revised total 5,400; payments 2,000 leave 3,400. An unapproved requested change is excluded from committed cost and shown separately. Acceptance checklist records requirement, evidence, inspector/owner, status, and correction. Finish with manuals, warranties, receipts, and a final payment review based on actual terms.

Acceptance Check

The budget example leaves 3,400; scope, approvals, dependencies, and acceptance evidence are explicit. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1497. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the plan to [room/project] with these professional assessments and quotes: [paste].

Change the workflow: Create a like-for-like quote comparison that exposes exclusions.

Add a useful view: Add a change approval log showing requested versus committed cost.

Check the inputs. Verify local approvals and technical requirements with the responsible sources.

Check the workflow. Do not compare quotes without aligning scope.

Check the result. Keep unapproved changes separate from committed totals.

P096 Restaurant herb-garden, patio planting, or edible-growing observation record

Facilities, equipment and resources

Source project: Gardening Project. Master pp 1107, 1499-1500, 3980. Restaurant application below is an editorial adaptation.

Restaurant application

Use verified plant identity, measured site, actual care sources, local seasonal guidance, tasks, and harvest observations when relevant.

Source based lesson (restaurant wording adapted)

Record an actual restaurant patio-landscaping or identified growing-area project using observed conditions and appropriate local sources.

Gather and build

Plant identity, location, light observations, local season, care sources. Map the space; record locally appropriate guidance; log care and observations without guessing treatments.

Check before relying on it

Verify plant identity before relying on species-specific guidance. The organizer does not establish edibility, pesticide suitability, or safe food use. Consult a local extension resource for one unresolved care question.

Source based practice with fictional inputs

Two plants in a restaurant patio project share pot size but have different verified identities and light exposure.

Worked result: The organizer should not assign identical care solely from pot size. Record identity, location, observations, and an appropriate care source.

Try a changed condition: One plant's identity remains uncertain.

Check your answer: Mark it unresolved and obtain reliable identification before relying on species-specific guidance.

Restaurant boundary

No inferred edibility, pesticide directions, safe harvest/food handling, watering/frost assumptions, or claim that garden output is approved for commercial food use.

P096 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask whether the restaurant project is patio landscaping or an identified growing area; measured site and observed conditions; verified plant identity; local authoritative guidance; source-based care requirements; available time/budget; and any separate approvals needed for commercial food use.

Structure

Site map; plant list; planting calendar; care log; supplies; observations; season review.

Build Requirements

Record the site using user-provided measurements and observed light, not guessed conditions. Plant fields: PlantID, CommonName, VerifiedSpeciesOrVariety, Location, PlantingDate, Source, CareReference, Notes. Flag uncertain plant identity before giving species-specific guidance.

Use current local extension, botanical, or supplier guidance for planting windows and care requirements. Do not invent frost dates, watering intervals, pesticide directions, or plant edibility. Care tasks should be observation- and source-based; product applications follow the actual label and local rules.

Calendar fields: TaskID, PlantID, Task, DueWindow, Source, Owner, Status. Log watering or other care only as performed, with quantity/unit when useful. Track observations such as growth, leaf changes, flowering, or harvest without diagnosing a cause from a short description.

Create a materials list and a season budget. Test a bed 2 meters by 3 meters has 6 square meters; plant spacing calculations require the chosen crop's verified spacing and layout, not a universal density. Provide a season review of what grew well, what conditions were observed, and what to change next time.

Acceptance Check

The site area example is 6 square meters; care and seasonal advice retain their verified source. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1501. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the plan to [containers/beds/indoor space] with these measurements and light observations: [list].

Change the workflow: Build a planting calendar from these local official or extension sources: [links].

Add a useful view: Add a harvest or bloom log with consistent units and dates.

Check the inputs. Confirm plant identity before specific care or edibility claims.

Check the workflow. Use local seasonal guidance instead of invented dates.

Check the result. Do not treat an observation as a confirmed diagnosis.

P097 Restaurant uniform, apron, and event-outfit readiness index

Facilities, equipment and resources

Source project: Personal Style Project. Master pp 1108, 1503-1504, 3992. Restaurant application below is an editorial adaptation.

Restaurant application

Map wardrobe items to actual owned uniform/appearance choices and approved role requirements; preserve available/laundry/repair/stored states, care-label references, and purchase review.

Source based lesson (restaurant wording adapted)

Organize approved restaurant uniforms or work clothing and distinguish ready owned items from proposed purchases.

Gather and build

Owned garments, verified condition, approved role/occasion requirements, compatible combinations, actual use. Inventory usable approved clothing; identify workable combinations; record a genuine gap only after checking what is available. Do not invent PPE or hygiene requirements.

Check before relying on it

A saved shopping item is not an owned garment. Plan five relevant service or off-site-event uniform combinations and verify which actual items are ready under the applicable approved requirements.

Source based practice with fictional inputs

A restaurant work-clothing inventory contains ten owned garments and four saved shopping items.

Worked result: Owned quantity is ten; saved items are not available garments. Separate owned, considered, and acquired states.

Try a changed condition: One saved item is purchased and received.

Check your answer: Owned quantity becomes eleven, with the item's actual status updated once.

Restaurant boundary

Do not invent PPE, commercial sanitation, gendered dress rules, body standards, or employment policies. This is clothing availability and care-record organization.

P097 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which approved restaurant roles/events the clothing plan supports; actual owned uniform items; fit/comfort/access needs; climate; care-label instructions; laundry/repair availability; and approved purchasing limits. Do not invent PPE, food-sanitation, or employment dress requirements.

Structure

Uniform/work-clothing index; outfit combinations; weekly plan; laundry/care references; repair/donate list; gap and purchase review.

Build Requirements

Index fields: ItemID, Category, Description, ColorOptional, SeasonOrUse, FitNotes, Status, CareLabelRef. Use Available/Laundry/Repair/Stored/Outgoing status. Do not impose body ideals or gendered rules; use the user's comfort and expression preferences.

Create outfit records linking actual ItemIDs, occasion, weather assumptions, shoes/accessories if useful, and readiness. Do not recommend buying a new item for every gap before checking existing combinations. Clothing care instructions must come from labels or verified manufacturer guidance.

Build a weekly outfit plan around real activities and laundry availability. A garment in Laundry is not ready merely because it appears in the index. Add a repair/alteration task list and a purchase decision worksheet: need, existing substitutes, expected use, budget, and wait/review date.

Test five planned outfits using three reusable base items: count outfits and owned pieces separately. If one required piece is unavailable, mark the affected outfit Needs substitute. Provide a seasonal review focused on usefulness, comfort, and keeping records light enough to maintain.

Acceptance Check

Outfits reference real items, unavailable pieces are flagged, and purchase decisions check existing alternatives first. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1505. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the planner to [activities/style/comfort needs].

Change the workflow: Create an off-site-event uniform pack from these existing items: [list], without assuming new purchases.

Add a useful view: Add a repair and alteration queue with cost estimates and next actions.

Check the inputs. Owned does not mean currently available.

Check the workflow. Use actual care labels for garment treatment.

Check the result. Distinguish outfit combinations from item counts.

P098 Restaurant utility usage and bill-normalization workbook

Facilities, equipment and resources

Source project: Utility Management Project. Master pp 1109, 1507-1508, 4004. Restaurant application below is an editorial adaptation.

Restaurant application

Use actual restaurant utility bills, period convention, same-unit daily consumption, verified charge components, meter identity, and cost-versus-usage comparison.

Source based lesson (restaurant wording adapted)

Record restaurant utility consumption and bills in consistent units and billing periods, separating daily use from price changes.

Gather and build

Meter unit, usage, billing days, rate details, fixed charges. Log bills consistently; compare usage per day; separate usage changes from price changes.

Check before relying on it

Do not add electricity kWh to gas therms as if they were the same unit. Explain one bill change using both consumption and pricing information.

Source based practice with fictional inputs

Restaurant utility period A records 300 kWh over 30 days; period B records 330 kWh over 33 days.

Worked result: Both average 10 kWh/day. Compare the same units and time basis.

Try a changed condition: B's bill is higher because the rate changed.

Check your answer: Separate consumption from price effects; the unchanged daily-use rate does not explain the entire bill.

Restaurant boundary

Do not add unlike utility units, infer equipment caused changes, or treat simple normalization as an energy audit. Occupancy/weather/service volume can limit comparisons.

P098 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant utilities and unit types are included; exact billing periods; bill usage/charge components; meter identity; service-volume or weather context; and whether usage, cost, or both will be compared.

Structure

Bills: BillID, Utility, PeriodStart, PeriodEnd, Usage, Unit, FixedCharge, UsageCharge, OtherCharge, Total, Days, UsagePerDay. Add Readings and Summary.

Build Requirements

Define whether PeriodEnd is exclusive; use a consistent convention so Days=End-Start for an exclusive end. Require end>start. UsagePerDay=Usage/Days. Total=FixedCharge+UsageCharge+OtherCharge, with zero explicit and missing charges flagged. Record provider-stated totals separately if reconciling them.

Compare the same utility and units; never add kWh, cubic meters, and gallons into one usage total. If bills vary in length, daily usage is more comparable than raw totals, but restaurant operations and weather changes still limit conclusions. Do not claim a device caused savings from a simple before/after comparison.

Readings fields: ReadingID, Date, Utility, MeterRef, Reading, Unit. Difference between successive readings requires the same meter/unit and handling of resets or replacements. Do not assume all meters are cumulative or infer pricing from one total bill.

Test 300 kWh over 30 days gives 10 kWh/day; 280 kWh over 28 days also gives 10. Charges 15 fixed, 45 usage, and 5 other total 65. Provide monthly views for cost and unit-specific usage and a question list for unexplained bill changes. No provider account connection is included.

Acceptance Check

Both usage examples equal 10 kWh/day and the charge example totals 65; invalid periods are flagged. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1509. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the tracker to [utilities] with their actual units and billing conventions.

Change the workflow: Add a meter replacement/reset log before calculating reading differences.

Add a useful view: Show cost and normalized usage separately, with restaurant operating-context notes.

Check the inputs. Do not combine unlike utility units.

Check the workflow. Use consistent inclusive/exclusive date rules.

Check the result. Separate fixed charges from usage charges before interpreting cost changes.