63LABS

Restauranteur Learn & Build

SECTION 06 OF 16

Catering Guests and Delivery

Relationships, sales, guest support, catering, events, travel and delivery records.



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

Catering, guests and commercial relationships 7

P004 Restaurant-industry relationship CRM 8

P005 Qualified restaurant prospect research 10

P006 Restaurant client engagements 13

P007 Catering and wholesale sales pipeline 15

P013 Restaurant service onboarding 18

P022 Restaurant-industry sales playbook 20

P023 Evidence-based restaurant outreach 22

P036 Guest feedback and improvement log 24

P037 Guest and supplier support workflow 26

P038 Restaurant merchandise returns and refunds 28

P047 Restaurant partnerships and sponsorships 30

P048 Restaurant membership or professional community 32

P072 Chef research trip, trade-show visit, or off-site catering travel organizer 34

P073 Off-site catering or restaurant-team packing checklist 36

P078 Catering-client gift, staff appreciation, or event-favor plan 38

P079 Restaurant-hosted wedding and wedding-catering coordination workspace 40

P080 Restaurant launch, tasting, private dining, or catering event plan 42

P093 Restaurant community-kitchen, fundraiser, or volunteer-session record 44

Restaurant delivery and catering records 46

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.

Catering, guests and commercial relationships

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.

P004 Restaurant-industry relationship CRM

P005 Qualified restaurant prospect research

P006 Restaurant client engagements

P007 Catering and wholesale sales pipeline

P013 Restaurant service onboarding

P022 Restaurant-industry sales playbook

P023 Evidence-based restaurant outreach

P036 Guest feedback and improvement log

P037 Guest and supplier support workflow

P038 Restaurant merchandise returns and refunds

P047 Restaurant partnerships and sponsorships

P048 Restaurant membership or professional community

P072 Chef research trip, trade-show visit, or off-site catering travel organizer

P073 Off-site catering or restaurant-team packing checklist

P078 Catering-client gift, staff appreciation, or event-favor plan

P079 Restaurant-hosted wedding and wedding-catering coordination workspace

P080 Restaurant launch, tasting, private dining, or catering event plan

P093 Restaurant community-kitchen, fundraiser, or volunteer-session record

P004 Restaurant-industry relationship CRM

Catering, guests and commercial relationships

Source project: Customer Relationships Project. Master pp 1015, 1131-1132, 2876. Restaurant application below is an editorial adaptation.

Restaurant application

Track corporate catering clients, restaurant buyers and agency accounts with dated interactions, owner and next action.

Source based lesson

Create a customer relationship manager for contacts, interaction history, and follow-up tasks linked by permanent contact IDs.

Gather and build

Contact ID, name, organization, source, interaction date, next action. Create separate contact and interaction lists; connect them by ID; add a view of overdue follow-ups.

Check before relying on it

Renaming a customer must not disconnect their history. Import five sample contacts and resolve a deliberate duplicate.

Source based practice with fictional inputs

C-01 has three interactions and C-02 has one. C-01 changes its display name.

Worked result: There remain two contacts and four interactions; C-01 retains its three linked interactions. Use the stable identifier for links; edit only the display name.

Try a changed condition: A new C-03 record shares C-01's office email but names a different person.

Check your answer: Do not merge solely on shared email; verify identity and link both people to the organization if appropriate.

Restaurant boundary

A recorded contact or event attendance is not permission for marketing. Deduplicate people and organizations; do not invent follow-up completion.

P004 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

Ask for industry; individual or team use; locale and currency if needed; contact relationship types; usual interaction channels; follow-up needs and any data I want excluded.

Base Tabs And Columns

Contacts: A Contact ID, B Name, C Company, D Email, E Phone, F Relationship, G Owner, H Added Date, I Contact Preference, J Notes, K Last Completed Contact. Activities: A Activity ID, B Contact ID, C Activity Date, D Channel, E Status, F Summary, G Owner. Tasks: A Task ID, B Contact ID, C Task, D Due Date, E Status, F Owner, G Completed Date, H Due Flag. Lists: relationships, contact preferences, channels, activity statuses, task statuses, and owners in separate columns. Dashboard: an all-time contact overview and a task snapshot as of today. Give explicit cells for every summary and chart.

STRUCTURE AND RULES Use manually assigned stable IDs such as C-001, A-001, and T-001. Contacts contains one row per person. Activities and Tasks can contain many rows with the same Contact ID, but each activity and task has its own unique ID. Two people at the same company remain separate contacts. Store phone numbers as text to preserve leading zeros and plus signs.

Default relationships: Prospect, Customer, Partner, Other. Contact preferences: Unknown, Email, Phone, Do Not Contact. Activity statuses: Planned, Completed, Cancelled. Channels: Email, Phone, Meeting, Other. Task statuses: Not Started, In Progress, Done, Cancelled. Preferences guide internal handling; Unknown does not mean permission to market.

Use Contacts IDs as dropdown values for Activities B2:B1000 and Tasks B2:B1000. Add visible checks for duplicate primary IDs and missing or invalid linked IDs. Do not delete a contact that has linked records; explain how to mark it inactive with an optional field instead.

CONTACT SUMMARY Contacts K shows the most recent dated Completed activity for that contact. Planned or Cancelled activities must not affect it. Return blank if no matching completed activity has a date. Give a COUNTIFS guard followed by MAXIFS so unmatched records do not display a zero-date. Flag completed activities missing dates and activity dates in the future.

TASKS Tasks H2 must be: =IF(A2="","",IF(OR(E2="Done",E2="Cancelled"),E2,IF(D2="","No date",IF(D2<TODAY(),"Overdue",IF(D2=TODAY(),"Today","Upcoming"))))) Fill H2:H1000. Keep Completed Date manual; TODAY() is not a permanent completion timestamp. Flag Done tasks with no Completed Date and tasks assigned to contacts marked Do Not Contact for review.

Dashboard

Show total contacts, contacts by relationship and owner, open tasks, overdue tasks, due-today tasks, open tasks without dates, and contacts with no completed interaction. Count contacts by nonblank Contact ID; count tasks by nonblank Task ID. Define open as neither Done nor Cancelled. Include a relationship bar chart and an overdue-task view that includes contact name, owner, due date, and action. Explain how to resolve duplicate contacts without orphaning their activities and tasks.

Tests

Create C-001 with one completed dated activity and one later planned activity. Only the completed date should appear in Contacts K. Add a Done task due yesterday and an open task due yesterday; only the open task is overdue. An activity linked to C-999 with no such contact must be visibly flagged. End with an entry routine and backup checklist.

Source extensions and troubleshooting

Master p1133. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt for my team: Add my owners [names or roles] and an owner filter to the CRM dashboard. Keep all shared summaries consistent and preserve stable IDs.

Add a contact field: Add [field] to Contacts at the far right. Explain its format, dropdown values if needed, and any formulas or views that must change.

Prepare an import: Map these existing headers to the CRM: [paste headers only]. Explain duplicate handling and generate a safe sample import using fictional data.

The last-contact date looks wrong. Check activity status, real date values, and exact Contact ID matches. A planned interaction must not update the last completed date.

Names no longer match records. Sort complete tables, not individual columns. Keep IDs fixed; changing an ID can break every linked record.

Blank tasks appear overdue. Use the ID and missing-date checks in H2. An empty due date should show No date for an open task, not Overdue.

Duplicate contacts exist. Choose the correct surviving Contact ID, repoint its Activities and Tasks, then remove the duplicate. Back up the workbook first.

P005 Qualified restaurant prospect research

Catering, guests and commercial relationships

Source project: Prospecting Project. Master pp 1016, 1135-1136, 2888. Restaurant application below is an editorial adaptation.

Restaurant application

Research restaurants that match a defined offer, record observable need, source/date, qualification and next research step.

Source based lesson

Organize prospective customers, sources, qualification, and the next outreach action before a lead becomes a sales opportunity.

Gather and build

Lead ID, source, fit criteria, contact status, next action date. Define qualified versus unqualified; record evidence of fit; review leads with no next action.

Check before relying on it

Every active lead has an owner and a dated next step. Review ten leads and explain why each belongs in its current status.

Source based practice with fictional inputs

A lead meets service-area and business-type criteria but has no identified decision-maker.

Worked result: Record the two verified criteria and the unresolved buyer field; do not claim full qualification under a rule requiring all three. Apply the written qualification rule and assign a buyer-identification action.

Try a changed condition: The buyer is identified but says the service is not relevant.

Check your answer: Record the new evidence and revise the status; identity alone does not establish need.

Restaurant boundary

Research is not permission to send. Keep contact eligibility and suppression separate; do not invent private contacts or claim observed problems without evidence.

P005 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

Ask for offer and target buyer; whether a lead means a person or a company; locale; sources; qualification criteria; team owners and follow-up timing.

Base Tabs And Columns

Leads: A Lead ID, B Date Added, C Company, D Contact Name, E Email, F Phone, G Source, H Need or Fit Notes, I Status, J Owner, K Last Contact Date, L Next Follow-up Date, M Next Action, N Qualified Date, O Notes, P Follow-up Flag, Q Days Since Contact. Lists: sources, statuses, and owners in separate columns. Dashboard: a first-of-month date in B2 with Month Start in A2; monthly lead additions and current status counts for that month's lead cohort, plus a clearly separate follow-up snapshot as of today.

Use statuses New, Contacted, Replied, Qualified, Unqualified, Do Not Contact. Qualified, Unqualified, and Do Not Contact are closed for this prospecting tracker. Define my chosen record unit in a visible instruction cell.

RECORD RULES One ID represents one lead under the selected person-or-company definition. A new attempt does not create a new lead. Record a short history in Notes if needed; recommend a separate Activity Log only if I request a detailed interaction history. Keep phone numbers as text. Use source dropdowns consistently. Highlight duplicate Lead IDs and review possible duplicate emails while ignoring blank emails.

Dates Added, Last Contact, and Qualified Date are manually entered real dates. Last Contact is the date of an actual attempt, not a scheduled follow-up. Qualified Date is entered when the lead meets my stated criteria. Do Not Contact removes the lead from the outreach queue. No field should imply permission for unsolicited communication.

Formulas

Put this in Leads P2 and fill P2:P1000: =IF(A2="","",IF(OR(I2="Qualified",I2="Unqualified",I2="Do Not Contact"),"Closed",IF(L2="","No date",IF(L2<TODAY(),"Overdue",IF(L2=TODAY(),"Today","Upcoming"))))) Put this in Q2 and fill Q2:Q1000: =IF(A2="","",IF(K2="","",TODAY()-K2)) Flag future Last Contact dates, which create negative days. Blank last-contact dates mean no recorded attempt, not zero days. Use conditional formatting to highlight missing Date Added, missing status, and Qualified rows without Qualified Date.

Dashboard

Show leads added in the selected month; current status counts for those same leads; all-time open leads; overdue follow-ups; follow-ups due today; open leads with no follow-up date; and leads never contacted. Use Date Added for month-cohort metrics. If showing qualifications by month, use Qualified Date and label it separately.

Provide a source summary for leads added during the selected month, with a bar chart. Provide a filterable follow-up view showing lead ID, company or contact, owner, next action, and due date. Closed statuses must be excluded. If using FILTER, use equal-height source and condition ranges and show a friendly no-results message with IFNA.

Tests

An open Contacted lead due yesterday is Overdue. A Qualified lead due yesterday is Closed. A new lead without a follow-up date is No date. A blank row displays no flags. Move a lead's Date Added outside the selected month and confirm it leaves the cohort counts. Duplicate IDs must be flagged rather than silently counted as separate prospects. Finish with a daily outreach routine and a qualification handoff checklist.

Source extensions and troubleshooting

Master p1137. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Define qualification: Adapt the tracker for [offer] sold to [buyer]. Give me three observable qualification criteria and a short handoff checklist. Do not treat interest alone as a purchase commitment.

Add a source: Add [source] to Lists, the lead dropdown, and the selected-month source report. Keep existing records and formulas intact.

Create a weekly queue: Add a view of open leads with follow-ups from today through the next six days. Keep overdue leads in a separate view. Give exact cells and formulas.

Follow-up counts seem inflated. Check for repeat rows created after each attempt. Update the existing Lead ID rather than counting the same prospect again.

Qualified leads still appear. Make sure the status exactly matches Qualified and that the follow-up formula treats it as closed. Update formulas if you rename a status.

Days since contact is negative. The Last Contact Date is in the future. Move scheduled outreach to Next Follow-up Date.

No leads match a FILTER view. A no-match result is normal. Use IFNA around FILTER for a friendly message; keep all filtered ranges the same height.

P006 Restaurant client engagements

Catering, guests and commercial relationships

Source project: Client Management Project. Master pp 1017, 1139-1140, 2900. Restaurant application below is an editorial adaptation.

Restaurant application

Keep one restaurant client separate from its website, menu and merchandise engagements, with scope, approved inputs, milestones and completion evidence.

Source based lesson

Track work after a sale: client engagements, delivery dates, check-ins, owners, and upcoming renewals.

Gather and build

Client ID, agreed scope, kickoff date, deliverables, check-in date. Separate clients from individual projects; track delivery and renewal tasks for each engagement.

Check before relying on it

Closing one project must not mark every client engagement complete. Create a weekly list of commitments due within seven days.

Source based practice with fictional inputs

Client C-01 has P-01 delivered and P-02 awaiting assets.

Worked result: P-01 is complete; P-02 remains blocked; the client has one open engagement. Store client and project states separately.

Try a changed condition: P-02 receives all approved assets but has not been produced.

Check your answer: Move it to ready or in progress under the defined workflow, not complete.

Restaurant boundary

Do not merge a client with an engagement; record changes and actual delivery, not assumed completion.

P006 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

Ask for services; project or recurring work; locale; delivery stages; owners; check-in frequency and renewal warning window. Default the renewal warning window to 30 days as a configurable preference.

Base Tabs And Columns

Engagements: A Engagement ID, B Client ID, C Client Name, D Service, E Start Date, F Target Delivery Date, G Status, H Billing Cadence, I Renewal Date, J Next Check-in, K Owner, L Next Action, M Notes, N Check-in Flag, O Renewal Flag. Lists: statuses, services, billing cadences, and owners in separate columns. Dashboard: B2 is Renewal Warning Days, with its label in A2; default 30. Other metrics show the current snapshot, not historical month-end counts.

Statuses: Onboarding, Active, On Hold, Completed, Cancelled. Billing cadences: One-time, Monthly, Quarterly, Annual, Other. Cadence describes the arrangement; this workbook does not calculate invoices or collect payments.

RECORD RULES One row is one engagement or service agreement. Engagement ID is unique; Client ID may repeat across engagements. Keep the same Client ID and name for the same client. Flag conflicting names under one Client ID, missing Client IDs, and duplicate Engagement IDs. Do not count engagements as distinct clients.

Open engagements are Onboarding, Active, and On Hold. Completed and Cancelled are closed. Include On Hold in check-in and renewal reviews unless I explicitly change that rule. Leave Renewal Date blank for work that does not renew; do not invent a date from Billing Cadence. Renewal dates must come from my actual records. No renewal should be assumed to guarantee income.

Formulas

Engagements N2, filled through N1000: =IF(A2="","",IF(OR(G2="Completed",G2="Cancelled"),"Closed",IF(J2="","No date",IF(J2<TODAY(),"Overdue",IF(J2=TODAY(),"Today","Upcoming"))))) Engagements O2, filled through O1000: =IF(A2="","",IF(OR(G2="Completed",G2="Cancelled"),"Closed",IF(I2="","None",IF(I2<TODAY(),"Past due",IF(I2<=TODAY()+Dashboard!B2,"Due soon","Later"))))) Require Dashboard B2 to be a nonnegative whole number. Today's renewals count as Due soon. Past due is a review flag, not proof of a contractual obligation or late payment.

Dashboard

Show open engagements, distinct clients with at least one open engagement, engagements by status and owner, overdue target deliveries, overdue check-ins, today's check-ins, missing check-in dates, renewals due within the warning window, and past-due renewal dates. Exclude closed engagements from open-work measures. Exclude missing dates from overdue counts and display them separately.

For distinct active clients, filter nonblank Client IDs to open engagement statuses before counting unique values; handle no matches as zero. Provide an upcoming-renewal list and a status bar chart. Include a visible definition for each measure. Do not claim that editing a current status reconstructs past status history.

Tests

Use two open engagements for C-001 and one completed engagement for C-002: open engagements=2 and active clients=1. An open engagement with Renewal Date exactly today plus the warning days is Due soon; one day later is Later. Completed engagements show Closed. A blank renewal date shows None. End with a weekly delivery review and instructions for adding a second engagement for an existing client.

Source extensions and troubleshooting

Master p1141. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Add delivery milestones: Add a separate Milestones tab linked by Engagement ID, with milestone ID, description, owner, due date, status, and completion date. Give all headers, formulas, dropdowns, and sample checks.

Change the review window: Set the renewal warning window to [number] days. Preserve the distinction between Past due, Due soon, Later, None, and Closed. Show the boundary-date tests.

Adapt to my service: Customize services, statuses, and next-action examples for [business]. Map every new status to open or closed and revise all affected formulas.

Client counts are too high. Count unique Client IDs among open engagements. A client with two projects is still one client.

Completed work looks overdue. Make sure Completed and Cancelled are excluded from active delivery and check-in measures.

A renewal date is unknown. Leave it blank and confirm the agreement. A blank renewal date is None; it must not become an overdue date.

A filter has no active clients. Handle no matching rows as zero. Do not count an empty-string placeholder as one distinct client.

P007 Catering and wholesale sales pipeline

Catering, guests and commercial relationships

Source project: Sales Management Project. Master pp 1018, 1143-1144, 2912. Restaurant application below is an editorial adaptation.

Restaurant application

Track qualified inquiries, proposed scope, value, next action and actual won/lost outcome for events, wholesale orders or service packages.

Source based lesson

Build a deal tracker with stages, values, expected close dates, next actions, and clearly labeled pipeline estimates.

Gather and build

Deal ID, stage, amount, expected close date, next step. Write stage entry rules; record stage-change dates; keep won revenue separate from estimated pipeline.

Check before relying on it

Lost deals leave the active forecast without disappearing from history. Review stale deals and update assumptions using evidence.

Source based practice with fictional inputs

Deal A is $1,500 at assumed 40%; deal B is $800 at assumed 25%. No deposits are received.

Worked result: Weighted estimate is $600 + $200 = $800; collected cash is $0. Display forecast and cash receipts separately.

Try a changed condition: Deal A is lost while B's assumed probability rises to 50%.

Check your answer: The active weighted estimate becomes $400; cash remains $0 absent a payment.

Restaurant boundary

Probability-weighted pipeline is an estimate, not booked revenue or cash. Preserve zero-probability and duplicate-deal tests.

P007 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

Ask for business and offer; locale and currency; sales stages; owners; reporting month; stale-deal threshold and probability evidence. Default the stale threshold to 14 days, configurable.

Base Tabs And Columns

Deals: A Deal ID, B Company, C Contact, D Deal Name, E Stage, F Deal Value, G Probability, H Expected Close Date, I Actual Close Date, J Owner, K Last Activity Date, L Next Action, M Next Follow-up, N Weighted Open Value, O Days Idle, P Follow-up Flag, Q Loss Reason. Lists: stages, suggested probabilities, and owners. Default stages are Discovery, Qualified, Proposal, Negotiation, Won, Lost. Dashboard: B2 is selected Month Start, B3 is Stale Days. Give explicit cells for summary metrics, stage tables, and charts.

One row represents a deal, not a contact attempt. Deal IDs are permanent and unique. Multiple deals may have the same company or contact.

STAGES AND VALUES Use Discovery, Qualified, Proposal, and Negotiation as open. Won and Lost are closed. Deal Value is the full value under one consistent definition that I choose; do not mix monthly fees and annual contract values. Values must be nonnegative and use one currency.

Probability is a numeric percentage from 0% to 100%, entered manually. Suggested defaults of 10%, 30%, 50%, and 70% for open stages are placeholders, not measured conversion rates. Won must be 100% and Lost 0%; flag inconsistencies. Do not imply that stage changes automatically update a manually entered probability. Explain a lookup-based alternative only if I request it.

Formulas

Deals N2, filled through N1000:
=IF(A2="","",IF(OR(E2="Won",E2="Lost"),0,IF(OR(F2="",G2=""),"",F2*G2))) Deals O2, filled through O1000: =IF(A2="","",IF(K2="","",TODAY()-K2)) Create P2 with blank-ID handling, Closed for Won/Lost, No date for missing follow-up, and Overdue/Today/Upcoming for open deals. Flag missing stage, value, probability, next action, and close date as applicable. Flag future Last Activity dates and open deals idle longer than Dashboard B3. Separate no recorded activity from zero days idle.

Dashboard

Show open deal count, total open value, weighted open value, open value expected to close in the selected month, overdue follow-ups, stale open deals, Won value by Actual Close Date for the month, Lost count by Actual Close Date for the month, and closed-deal win rate for that month.

Define win rate as Won count divided by Won plus Lost count in the chosen month, using Actual Close Date for both. Display N/A when there are no closed deals. Never divide by all leads or

open deals. Show a stage count and stage value table with a bar chart. Use inclusive month start and exclusive next-month start for all monthly measures.

Expected Close Date is a forecast input; Actual Close Date records an outcome. Won deal value is not cash collected or accounting revenue. Weighted pipeline is a planning estimate, not guaranteed sales. Missing required inputs must be visible rather than silently treated as zero.

Tests

Use a 2000 open deal at 25%, a 1000 Won deal, and a 500 Lost deal. Open value=2000; weighted open value=500. If Won and Lost share the selected actual-close month, win rate=50%. Moving only the expected close date must not alter closed-deal metrics. Finish with a weekly pipeline review checklist.

Source extensions and troubleshooting

Master p1145. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Use my sales stages: Replace the stages with [stages]. Map each one to open, won, or lost. Update dropdowns, closed-stage exclusions, summary formulas, and tests.

Set consistent deal values: My business sells [offer and billing terms]. Help me choose one consistent Deal Value definition and document it. Keep contract value, invoices, and cash collected distinct.

Add a sales review: Create a weekly review view showing owner, stage, value, days idle, expected close date, and next action for open deals. Sort the oldest activity first.

Weighted values are 100 times too large. Enter 25% or 0.25, not 25. Validate probabilities between 0 and 1 and format the column as Percent.

Won deals inflate open pipeline. Exclude Won and Lost from both open deal count and open value. Their Weighted Open Value should be zero.

Win rate changes unexpectedly. Check Actual Close Date and use only Won and Lost deals in that month. With no closed deals, show N/A rather than a misleading 0%.

A forecast looks like revenue. Label it weighted open pipeline. A probability-weighted deal estimate is separate from signed deal value and money received.

P013 Restaurant service onboarding

Catering, guests and commercial relationships

Source project: Client Onboarding Project. Master pp 1024, 1167-1168, 2984. Restaurant application below is an editorial adaptation.

Restaurant application

Collect an approved menu, actual photos, logo rights, scope, contacts, deposit status and required access before a restaurant client project starts.

Source based lesson

Move a new client from accepted scope to a prepared kickoff using a repeatable checklist and ready-to-edit messages.

Gather and build

Accepted scope, client contact, assets, access needs, kickoff agenda. Create a request list; assign owners and dates; confirm prerequisites before kickoff.

Check before relying on it

Do not request passwords by email; use delegated access where supported. Rehearse onboarding with a fictional client and track every unanswered question.

Source based practice with fictional inputs

Scope and payment are confirmed; image-use permission and final text are missing.

Worked result: The intake record shows two unresolved prerequisites and named owners. Track each prerequisite independently; only begin work with sufficient approved inputs.

Try a changed condition: Final text arrives, but image permission remains unresolved.

Check your answer: Close only the text item; keep the image-dependent work blocked.

Restaurant boundary

Do not request passwords in a worksheet. Use approved delegated access; source intake does not itself authorize collection, payment or publication.

P013 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

1. What service and client type is this for?
2. What must be signed or paid before work begins?
3. Which assets and access are needed, and who supplies each?
4. Who approves deliverables and changes?

5. What response windows and kickoff date are agreed?

Structure

Create a welcome packet, intake questionnaire, onboarding checklist, asset/access request, kickoff agenda, handoff note, and three client message templates. Include a simple tracker with ClientID, Step, Owner, DueDate, Status, EvidenceLink, Blocker.

Build Requirements

Write the complete welcome packet: scope recap, included deliverables, exclusions, timeline, client responsibilities, communication channel, approval contact, and next action. Use placeholders for agreement terms, never fabricated commitments. Intake questions must gather project goals, audience, success criteria, existing assets, constraints, accessibility needs, and approval process without asking for passwords.

The checklist must cover accepted scope, agreement confirmation, initial payment confirmation if applicable, intake, assets, secure access invitation, kickoff, and internal handoff. Each row needs owner, dependency, due date, and evidence required. Status values: Not started/Waiting on client/In progress/Complete/Not applicable. Define kickoff-ready as all mandatory gates Complete, with approved exceptions documented separately. A sent request is not completed access.

Supply actual wording for welcome, missing-item reminder, and kickoff confirmation. Each message includes one clear action, due date, support contact, and a respectful escalation route. Create a 30-minute kickoff agenda with time allocations and a decision log. Add a client-facing one-page checklist and an internal version with risk notes. Use access invitations and approved sharing channels; the tracker stores a reference, not credentials.

Walk through a fictional design client: agreement confirmed, deposit pending, assets received, access invitation unaccepted. Show why kickoff is not ready and draft the two next requests. Include a seven-day implementation schedule and a reusable project-folder naming pattern.

Acceptance Check

Every mandatory gate has an owner and evidence; the fictional client remains blocked until payment and access are confirmed. 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 p1169. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Offer-specific: Adapt the packet to [service] with these required assets and gates: [list].

Shorter onboarding: Reduce the intake to the essential questions for [small offer] while keeping scope, approval, and access clear.

Client tone: Rewrite the three messages in [tone], preserving the action, owner, and deadline.

Client is stuck. Identify the single blocking step and send one request with an example of an acceptable response.

Scope expands. Record the request and route it through the agreed change process before promising delivery.

Access is missing. Check invitation acceptance and permissions; do not ask the client to email a password.

Checklist says done too soon. Require the stated evidence, not merely that a request was sent.

P022 Restaurant-industry sales playbook

Catering, guests and commercial relationships

Source project: Sales Strategy Project. Master pp 1033, 1203-1204, 3092. Restaurant application below is an editorial adaptation.

Restaurant application

Define a truthful offer, buyer questions, objections, qualification and next steps for catering, supply or professional services.

Source based lesson

Create a repeatable sales conversation and deal-review process with qualification, evidence, next steps, and handoff.

Gather and build

Buyer role, qualification questions, objections, proof, handoff. Map discovery, proposal, decision, and delivery handoff; write questions that reveal fit and limits.

Check before relying on it

A confident conversation must not substitute for agreement on scope and price. Role-play one qualified prospect and one poor-fit prospect.

Source based practice with fictional inputs

A prospect requests five-day delivery; current verified capacity permits ten days.

Worked result: Offer an agreed narrower scope or later date, or decline. Record scope, timing, price, and approval explicitly.

Try a changed condition: The prospect accepts a smaller deliverable in five days.

Check your answer: Confirm the revised scope and acceptance criteria before handoff.

Restaurant boundary

Use actual prices, capacity and evidence. No pressure tactics, invented results or guaranteed sales.

P022 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

Who buys and who approves? What problem and buying trigger matter? What stages and pricing authority exist? What objections occur? What qualifies or disqualifies an opportunity?

Structure

Customer fit; qualification scorecard; discovery guide; stage exit criteria; demo/proposal plan; objection responses; follow-up templates; handoff checklist.

Build Requirements

Define stages Prospect/Qualified/Discovery/Proposal/Decision/Won/Lost with observable exit criteria and required evidence. Do not advance a deal merely because a follow-up was sent. Build a qualification worksheet for problem, impact, fit, decision process, timing, resources, and next agreed action; unknown is distinct from no.

Write a discovery opening, ten useful questions, an agenda, and a recap. Provide a demonstration structure tied to the buyer's stated problem rather than a list of features. Include a proposal outline with scope, exclusions, assumptions, price supplied by the user, acceptance process, and next step.

Create five objection responses using acknowledge, clarify, evidence, and agreed next action. Avoid fabricated case studies, false urgency, or pressure to ignore a poor fit. Include a disqualification script that remains respectful. Add three follow-up emails: discovery recap, proposal clarification, and close-the-loop message.

Define pricing authority and an exception approval field; do not invent discounts. Handoff to delivery must include accepted scope, decision contact, commitments, payment milestone, risks, and kickoff owner. Record loss reason without rewriting history.

Show a fictional deal where interest is high but the buyer's approval process is unknown. Keep it in Discovery and request the missing decision information. Reporting definitions: win rate=Won/(Won+Lost) for a stated cohort; open opportunities are excluded. Three Won and two Lost gives 60%, regardless of four still-open deals.

Acceptance Check

Scripts and stage criteria are complete; the example stays in Discovery; win rate equals 60%. 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 p1205. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt qualification and scripts to [offer] and [buyer role].

Change the workflow: Add a manager approval process for discounts above [threshold].

Add a useful view: Create a coaching scorecard for discovery quality and evidence-based next steps.

Check the inputs. Unknown budget or authority should remain unknown rather than being inferred.

Check the workflow. Require buyer evidence for stage changes.

Check the result. Use the same cohort and closed-deal denominator when comparing win rates.

P023 Evidence-based restaurant outreach

Catering, guests and commercial relationships

Source project: Outreach Project. Master pp 1034, 1207-1208, 3104. Restaurant application below is an editorial adaptation.

Restaurant application

Draft a relevant message tied to an observed restaurant need such as a broken menu link, and track approved follow-up and suppression.

Source based lesson

Build a small, manually reviewed outreach workflow with relevant prospects, truthful drafts, response handling, and suppression.

Gather and build

Relevant prospect, reason for contact, truthful offer, opt-out process. Research a small list; draft a specific message; review each recipient and maintain suppression records.

Check before relying on it

Do not invent familiarity, results, or permission; review the FTC email guidance. Measure useful replies from a small test before increasing volume.

Source based practice with fictional inputs

A public menu link is broken; the sender has no evidence about the restaurant's revenue.

Worked result: A message can mention the observed link and ask whether the owner wants help. Remove invented impact and keep the observation and proposed task distinct.

Try a changed condition: The owner explains the menu is intentionally unavailable this week.

Check your answer: Acknowledge the context and revise or close the opportunity.

Restaurant boundary

Drafting is not sending; no scraped-list default or invented relationship. Current sending rules and approved recipient eligibility must be checked.

P023 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

Who are the recipients and why is the offer relevant? Where are sender and recipients located? What source and contact permissions apply? What proof may be used? What stop conditions and response process will you use?

Structure

Prospect qualification sheet; source/relevance record; pre-send eligibility gate; initial email plus two optional follow-ups; response triage; suppression log; outcome report.

Build Requirements

Before drafting operational sending instructions, identify jurisdiction and provider requirements. Use current official sources, including the U.S. guide at https://www.ftc.gov/business-guidance/resources/can-spam-act-compliance-guide-business when relevant. Mark unresolved eligibility as Hold. Do not present B2B outreach as automatically exempt, or claim worldwide compliance. Avoid scraped/purchased lists as the default and never invent personalization.

Prospect fields: ProspectID, Organization, Role, Contact, SourceURL, SourceDate, RelevanceEvidence, EligibilityReview, LastContact, NextActionDate, ResponseStatus. Keep a separate suppression list with normalized email, reason, date, and source. Suppression overrides sequence membership and imports.

Draft one concise initial email and two optional follow-ups. Use truthful identity/subject, a specific relevant problem, only verified proof, one low-pressure CTA, and clear sender/address/opt-out placeholders as applicable. No fake prior conversation, deceptive 'Re:', fabricated customer relationship, or guaranteed outcome. Explain where each personalization fact came from.

Create response rules for interested, later, wrong person, not interested, unsubscribe, complaint, and bounce. A reply pauses automated-style follow-up; unsubscribe or complaint stops marketing contact. Set a user-chosen maximum sequence length and a human approval gate. This project drafts a system; it does not send messages or connect a mailbox.

Measure delivered messages, positive replies, booked conversations, and opt-outs for the same cohort. Positive replies/Delivered is not sales conversion. Example: 100 delivered, 5 positive replies, and 2 booked calls yields 5% positive replies and 2% booked-call rate. Test that a suppressed contact is excluded even with a future NextActionDate.

Acceptance Check

The three drafts are complete, unresolved eligibility stays Hold, and the suppressed-contact test prevents follow-up. 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 p1209. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Tailor the drafts for [offer/role] using these real relevance facts and proof assets: [paste].

Change the workflow: Create a manual review queue that excludes Hold, suppressed, bounced, and replied contacts.

Add a useful view: Add reporting by source with identical cohort definitions and separate positive replies from meetings.

Check the inputs. Do not treat a public address as proof of permission or eligibility.

Check the workflow. Suppression must take precedence over every scheduled step.

Check the result. Remove personalized claims that cannot be traced to supplied or verified evidence.

P036 Guest feedback and improvement log

Catering, guests and commercial relationships

Source project: Customer Feedback Project. Master pp 1047, 1259-1260, 3260. Restaurant application below is an editorial adaptation.

Restaurant application

Adapt a short survey to dining, takeout or catering touchpoints and assign evidence-backed improvements.

Source based lesson

Collect structured feedback, group themes, assign improvements, and close the loop with evidence.

Gather and build

Question, response date, customer group, theme, action owner. Ask a small consistent question set; group responses; select one improvement and follow up.

Check before relying on it

Do not present five responses as representative of every customer. Show how a specific response led to a specific change.

Source based practice with fictional inputs

Five customers respond; four criticize the instructions and one praises them.

Worked result: Report the five responses and the recurring issue without claiming all customers agree. Limit the claim to the observed respondents and describe sampling limits.

Try a changed condition: A revision is tested with three different readers.

Check your answer: Treat this as new feedback, not a directly matched before-and-after result for the original five.

Restaurant boundary

Feedback is not public-testimonial permission. Blank scores are unscored, and self-selected small samples are directional.

P036 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

What experience are you evaluating? Who will be asked and when? What scale and questions are useful? Which identifiers are necessary? Who owns follow-up and improvement decisions?

Structure

Survey draft; Feedback table; theme codebook; improvement log; response templates; summary and review routine.

Build Requirements

Write a short survey with one clearly labeled satisfaction scale, two targeted questions, and an optional comment. Avoid leading questions and unnecessary personal data. State that participation is optional and separate feedback from permission to use a public testimonial.

Feedback columns: FeedbackID, Date, Touchpoint, Score, ScaleMax, Comment, Theme, FollowUpPermission, ContactRef. Use a single chosen scale per report. Missing score is unscored; do not convert it to zero. Improvement log: ActionID, Theme, EvidenceCount, Owner, DueDate, Status, Result.

Define themes with examples and allow Other/Unclear. Count responses by theme without treating repeated comments from one submission as extra respondents. Report response count, score distribution, and average only for valid numeric scores on the same scale. Explain selection bias and that small samples are directional.

Test scores 4, 5, 3, and blank on a five-point scale: valid count 3 and mean 4.0; total submitted responses 4. Draft a thank-you, a clarification request, and a response to negative feedback without defensiveness or invented remedies. Create a monthly review that selects one improvement and later records whether it helped.

Acceptance Check

The blank-score example averages 4.0 over three valid scores; each improvement has an owner and follow-up. 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 p1261. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt questions to [touchpoint] without leading respondents.

Change the workflow: Add a follow-up queue restricted to people who agreed to contact.

Add a useful view: Create a score distribution view with sample count and reporting dates.

Check the inputs. Keep blank scores out of the numeric denominator.

Check the workflow. Do not publish a testimonial without the needed permission.

Check the result. Avoid drawing broad conclusions from a small self-selected sample.

P037 Guest and supplier support workflow

Catering, guests and commercial relationships

Source project: Customer Support Project. Master pp 1048, 1263-1264, 3272. Restaurant application below is an editorial adaptation.

Restaurant application

Track delivery issues, guest complaints, product questions or support tickets using approved replies, escalation and resolution evidence.

Source based lesson

Track incoming support requests, ownership, response targets, resolution, and recurring causes.

Gather and build

Ticket ID, customer reference, category, priority, owner, status. Define urgency rules; assign responsibility; record resolution evidence and reopened cases.

Check before relying on it

Closed does not automatically mean the underlying issue is fixed. Review three repeated tickets and identify one preventable cause.

Source based practice with fictional inputs

A customer reports a missing download. An agent sends a clarification question.

Worked result: The ticket is waiting for information, not resolved. Define resolution by the actual customer task and preserve waiting states.

Try a changed condition: The customer receives a replacement but cannot open it.

Check your answer: Continue the issue and investigate access/compatibility; delivery alone did not resolve it.

Restaurant boundary

Do not infer resolution from a status change; food-safety complaints require approved escalation. Source elapsed-hour logic is not automatically business-hour SLA logic.

P037 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

What requests arrive and through which channels? Who owns triage? What severity definitions and response targets are approved? What hours and timezone apply? Which issues need escalation?

Structure

Tickets: TicketID, OpenedAt, CustomerRef, Category, Priority, Owner, Status, FirstResponseAt, ResolvedAt, NextAction, DueAt. Add Updates, Categories, and Queue.

Build Requirements

Use New/Triaged/In progress/Waiting on customer/Resolved/Closed status. Define priority using impact and urgency examples, not the customer's tone. Keep a dated update history and one accountable owner. Opening and resolution timestamps are values; do not use volatile NOW() as a permanent event time.

Calculate elapsed response hours=(FirstResponseAt-OpenedAt)*24 and resolution hours=(ResolvedAt-OpenedAt)*24 only when both timestamps are present and valid. Label them calendar hours. Business-hour targets require a separately designed calendar and must not be implied by simple elapsed-time formulas. DueAt is based on the approved rule, with missing targets shown as unset.

Provide queues for unassigned, due soon, overdue, waiting, and recurring issues. Draft acknowledgement, information request, resolution summary, and escalation messages. Keep private customer details out of public examples and never invent a fix.

Test a ticket opened 09:00, first response 10:30, resolved 15:00: response time 1.5 calendar hours, resolution 6. A reopened ticket retains its history and receives a new resolution event rather than erasing the original. Include close criteria requiring verified resolution or a documented policy-based closure reason.

Acceptance Check

The sample returns 1.5 and 6 hours; unresolved tickets remain open and missing timestamps do not become zeros. 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 p1265. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt categories and severity to [product/service].

Change the workflow: Add a business-hours calendar using these approved working times and holidays: [list].

Add a useful view: Create a recurring-issue report that groups tickets without exposing customer details.

Check the inputs. Do not overwrite timestamps with NOW() formulas.

Check the workflow. Distinguish calendar hours from business hours.

Check the result. Keep reopened-event history and an explicit owner.

P038 Restaurant merchandise returns and refunds

Catering, guests and commercial relationships

Source project: Returns Management Project. Master pp 1049, 1267-1268, 3284. Restaurant application below is an editorial adaptation.

Restaurant application

Record merch returns, damaged supplier orders or approved guest refunds, distinguishing request, approval, actual payment and stock disposition.

Source based lesson

Record return requests, decisions, item receipt, refund actions, and reconciliation against original orders.

Gather and build

Original order, requested amount, decision, returned goods, refund ID. Track request, approval, receipt, refund submission, and settlement as separate events.

Check before relying on it

Prevent a second refund for the same authorized amount. Match one completed refund to both the order and payment records.

Source based practice with fictional inputs

A $60 refund is approved; a $20 partial refund is confirmed paid.

Worked result: Paid is $20 and approved-but-unpaid is $40. Link payment events to the approved amount and calculate the remainder.

Try a changed condition: A duplicate record of the $20 payment is imported.

Check your answer: Keep confirmed paid at $20 after deduplication; do not reduce the remaining amount twice.

Restaurant boundary

No automatic refund/payment. Apply approved policies and current obligations; returned food must not be assumed suitable for resale or restocking.

P038 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

What products and channels are involved? What policy and legal requirements have been verified? Who can approve refunds? Are partial returns allowed? How will receipt and payment evidence be recorded?

Structure

Cases: CaseID, OrderRef, RequestedDate, Reason, RequestedAmount, Decision, ApprovedAmount, ReturnStatus, RefundStatus, RefundRef. Add Items, Events, and Reconciliation.

Build Requirements

Treat this as an operational record; use the user's verified policy and current applicable requirements rather than inventing eligibility rules. Decision values Pending/Approved/Declined/More information. A decision is separate from actual receipt of goods and from payment completion.

Track returned items by line reference and quantity. RefundStatus is Not due/Pending/Processed/Failed, with Processed requiring a payment reference and date. Preserve an event log for approvals, receipt, inspection, and refund confirmation. Draft acknowledgement, request-for-information, approval, and completion messages without promising an unverified timeframe.

Reconciliation fields: OrderRef, OriginalPaid, PriorRefunds, ProposedRefund, RemainingAfter. Formula=OriginalPaid-PriorRefunds-ProposedRefund, guarded for missing inputs. Negative remaining flags an excess refund for review; never silently cap or execute a payment. Keep currency consistent and explain handling shipping or tax from verified policy.

Test an order paid 120 with prior refunds 20 and proposed refund 30: remaining amount 70. A second duplicate refund reference is flagged. A returned item does not automatically mean funds were refunded. Include a weekly queue for approved-but-unprocessed and processed-without-reference cases.

Acceptance Check

The sample balance is 70, duplicate references are flagged, and no case implies money moved without evidence. 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 p1269. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the workflow to [product/channel] using this approved policy: [paste].

Change the workflow: Add partial item returns with line IDs and quantity controls.

Add a useful view: Create a reconciliation queue for missing payment evidence and excess refund amounts.

Check the inputs. Use actual payment references to confirm completion.

Check the workflow. Keep decision, item receipt, and refund status separate.

Check the result. Check duplicate cases and cumulative refunds before approval.

P047 Restaurant partnerships and sponsorships

Catering, guests and commercial relationships

Source project: Partnerships Project. Master pp 1058, 1303-1304, 3392. Restaurant application below is an editorial adaptation.

Restaurant application

Develop supplier, local-event, community or co-marketing proposals with mutual benefit, scope, evidence and permissions.

Source based lesson

Develop a mutually useful partnership proposal with clear deliverables, evidence, boundaries, and a review process.

Gather and build

Partner audience, proposed deliverables, schedule, measurement, price. Define what each party supplies; substantiate audience claims; agree how delivery will be evidenced.

Check before relying on it

Promises in the proposal must match operational capacity and available measurement. Create a mock fulfillment report before pitching the package.

Source based practice with fictional inputs

A proposal promises three placements over one month, with no evidence of resulting sales.

Worked result: Define delivery dates and proof of each placement; do not guarantee purchases. Distinguish controllable deliverables from uncertain business outcomes.

Try a changed condition: Only two placements are delivered by the deadline.

Check your answer: Record the missing deliverable and agree the response; do not report full delivery.

Restaurant boundary

Do not invent audience figures, partner agreement or guaranteed attendance; track commitments only when approved.

P047 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

Who is the partner and why is there a fit? What real audience or distribution evidence exists? What can each side provide? What budget or in-kind contribution is proposed? What dates, approvals, and measurement matter?

Structure

Fit assessment; value exchange; proposal; deliverables matrix; sample package options; measurement plan; outreach draft; negotiation and handoff checklist.

Build Requirements

Separate verified audience data from estimates and goals. Compare the partner's likely objective with what the user can actually deliver. Do not invent reach, endorsements, relationships, sponsor commitments, or guaranteed exposure.

Write a complete proposal with executive summary, audience fit, concept, deliverables, timeline, responsibilities, contribution requested, measurement, assumptions, and next step. Include up to three packages only when their deliverables differ meaningfully. Specify quantities, formats, dates, review rounds, approval owners, and excluded work.

Create a deliverables matrix: DeliverableID, Owner, DueDate, Approval, Evidence, MeasurementWindow. Rights to names, logos, content, exclusivity, cancellation, and commercial terms must use approved agreement language or remain review items; do not imply permission already exists.

Draft a short introductory email and a follow-up that references only real prior contact. Provide a negotiation worksheet distinguishing must-haves, flexible items, and approval limits. Example: a local workshop partner offers venue space while the organizer supplies facilitation and promotion; estimated attendance is labeled an estimate, not guaranteed.

Finish with a launch gate, shared approval checklist, post-activity results report, and a debrief agenda. Compare delivered commitments with the agreed package before evaluating broader outcomes.

Acceptance Check

The proposal and outreach are complete; each commitment has an owner, date, and evidence requirement. 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 p1305. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the proposal to [partner/objective] using these verified assets: [list].

Change the workflow: Create two package options within [delivery capacity].

Add a useful view: Add a commitment-versus-delivery report with evidence links and measurement dates.

Check the inputs. Do not substitute estimated audience for verified attendance or reach.

Check the workflow. Clarify ownership and approval for every promised asset.

Check the result. Keep rights and exclusivity questions open until agreed.

P048 Restaurant membership or professional community

Catering, guests and commercial relationships

Source project: Memberships Project. Master pp 1059, 1307-1308, 3404. Restaurant application below is an editorial adaptation.

Restaurant application

Plan clear member benefits, participation capacity, onboarding, schedule and an optional paid tier.

Source based lesson

Design an ongoing member offering with a clear promise, sustainable delivery, onboarding, and retention learning.

Gather and build

Member promise, recurring activities, capacity, onboarding, cancellation. Plan one delivery cycle; calculate workload per member; define what repeats each month.

Check before relying on it

The service remains deliverable if every member uses the included benefits. Pilot with a small group and review attendance, satisfaction, and workload.

Source based practice with fictional inputs

There are 40 signups: 5 cancelled, 3 paused, 2 unpaid; categories are distinct and unpaid is inactive.

Worked result: Active count is 30 under the stated rule. Apply a documented status definition and verify category overlap.

Try a changed condition: Two paused members return to active status.

Check your answer: Active count becomes 32 with dated changes retained.

Restaurant boundary

Club membership is not billing implementation. Verify actual recurring charges, cancellation and refund terms before launch; avoid promising unavailable benefits.

P048 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

Who joins and what recurring value do they need? What benefits can you reliably deliver? What billing period and price are proposed? What onboarding, support, cancellation, and moderation processes are approved?

Structure

Member promise; benefit schedule; capacity model; onboarding sequence; engagement calendar; member records; renewal/cancellation workflow; metrics and pilot plan.

Build Requirements

Define included benefits, frequency, access, exclusions, and support limits. Avoid adding benefits that cannot be sustained at the chosen member count. Create a month of delivery activities with owners and preparation time. Separate content production from live support effort.

Build a capacity example: 20 members requiring 15 minutes of individual support each consume 5 hours, before group sessions or administration. Model different member counts using explicit assumptions rather than promising passive income.

Write a welcome message, orientation checklist, first-week prompt, inactivity check-in, and cancellation acknowledgement. Use approved billing and cancellation terms; verify relevant platform and jurisdiction rules before implementation. Do not connect payments or enroll anyone.

Member records: MemberID, JoinDate, Plan, Status, CurrentPeriodStart, CurrentPeriodEnd, OnboardingComplete, SupportNotesRef. Keep payment credentials out of the tracker. Status Trial/Active/Paused/Cancelled/Expired must have clear meanings.

Define cohort retention as members from the starting cohort still active at period end divided by the starting cohort. New joiners do not enter that retention denominator. Starting with 20 and retaining 18 gives 90%, even if 5 new members join. Create a pilot with feedback questions, delivery evidence, and a continue/revise decision.

Acceptance Check

The capacity example totals five hours and cohort retention is 90%; benefits and operational terms 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 p1309. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt benefits to [audience] within [weekly hours].

Change the workflow: Create a small pilot for [member count] with explicit feedback and exit criteria.

Add a useful view: Add a cohort retention table that keeps new joins separate.

Check the inputs. Keep promised benefits within capacity.

Check the workflow. Distinguish cancellation intent from the verified access/billing end date.

Check the result. Do not mix new joins into retention of an existing cohort.

P072 Chef research trip, trade-show visit, or off-site catering travel organizer

Catering, guests and commercial relationships

Source project: Travel Project. Master pp 1083, 1403-1404, 3692. Restaurant application below is an editorial adaptation.

Restaurant application

Map travel purpose to restaurant business; preserve confirmed/proposed bookings, local times, transit buffers, costs, and change log.

Source based lesson (restaurant wording adapted)

Plan restaurant-business travel or off-site catering logistics around confirmed bookings, actual local times, realistic transfers, and documented alternatives.

Gather and build

Confirmed bookings, local dates, transit estimates, preferences. Separate confirmed from proposed activities; allow transfers and buffers; retain booking references securely.

Check before relying on it

Check local timezone and the actual travel date for every booking. Verify the first day's route using current official travel information.

Source based practice with fictional inputs

A restaurant-team trip arrives at 14:00; an off-site meeting is proposed for 14:15, but transfer time is unknown.

Worked result: The schedule's feasibility is unresolved. Verify dates, local times, transit, and buffer before committing.

Try a changed condition: Verified transfer is 40 minutes with a chosen 20-minute buffer.

Check your answer: Earliest planned arrival at the activity is 15:00 under these inputs.

Restaurant boundary

Verify live transport/venue facts; no invented bookings or travel rules. Food transport conditions are outside this travel organizer.

P072 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 the restaurant-business purpose of the trip; destinations/dates; participants; confirmed bookings; budget; work schedule; accessibility requirements; transport/timezone changes; and live details still needing official verification.

Structure

Trip brief; booking register; day-by-day itinerary; transport plan; budget; contact references; alternatives; final verification checklist.

Build Requirements

Separate Confirmed/Proposed/Needs verification for every booking and activity. Never invent reservations, confirmation numbers, current opening hours, prices, visa rules, or transit schedules. For live details, consult current official operator, venue, and government sources with checked dates; if browsing is unavailable, create a precise verification task.

Create each day with local date/timezone, location, activity, duration, transit allowance, booking requirement, expected cost, and backup option. Include meals, rest, and realistic buffers based on the user's pace. Do not schedule immediate activities after a long journey without considering arrival uncertainty.

Booking register: BookingID, Provider, Service, LocalDateTime, Timezone, Status, ConfirmationRef, CancellationDeadline, Paid, Balance. Keep sensitive passport and payment data outside broadly shared copies. Budget records currency per item; do not add currencies without explicit conversion inputs and dates.

Test an arrival at 14:00 with 90 minutes allowed for arrival formalities and transfer: the first planned hotel-area activity cannot start before 15:30 under that assumption. Label the buffer an assumption to verify. Supply a shareable daily summary, offline-document checklist, and a change log when a booking shifts.

Acceptance Check

Times include buffers and timezones; bookings have evidence; unknown prices or rules remain verification tasks. 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 p1405. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the itinerary to [pace/interests/access needs].

Change the workflow: Replan after this booking change: [details], preserving confirmed commitments.

Add a useful view: Create a compact offline daily view with contacts and confirmation references.

Check the inputs. Verify current facts with official sources rather than assuming old schedules.

Check the workflow. Label every time with its local timezone when crossing regions.

Check the result. Keep proposed activities distinct from confirmed reservations.

P073 Off-site catering or restaurant-team packing checklist

Catering, guests and commercial relationships

Source project: Trip Preparation Project. Master pp 1084, 1407-1408, 3704. Restaurant application below is an editorial adaptation.

Restaurant application

Map traveler to assigned team member and bag to equipment crate; retain Needed, Owned, ToAcquire, PackedQty, bag assignment, and shared-item ownership.

Source based lesson (restaurant wording adapted)

Build a restaurant-team trip, catering, or pop-up packing list by assigned person, use, bag/crate, required quantity, and packing status.

Gather and build

Confirmed destination/duration, approved activity and equipment list, team assignments, quantities, transport constraints. Create assigned-person lists; group by bag or crate; track required, owned, purchased, packed, and returned quantities distinctly.

Check before relying on it

A checked item must mean packed, not merely purchased. Check essential business documents and the verified carrier or site restrictions. Specialist transport and food controls must come from approved sources.

Source based practice with fictional inputs

A shared catering-team charger is assigned to one person and one bag; it is purchased but not physically packed.

Worked result: Purchased and packed are distinct states. Define the checkbox as physically packed and track shopping separately.

Try a changed condition: The assigned bag changes.

Check your answer: Update the location while keeping one shared-item record.

Restaurant boundary

Packing status does not certify safe food temperatures, transport, or equipment suitability. Keep specialized requirements source-based.

P073 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 off-site catering, pop-up, or restaurant-team trip is in scope; authorized equipment/supply list; team assignments; needed quantities and units; owned items; crate/bag capacity; transport constraints; and return checks. Food-temperature and specialist transport requirements must come from approved sources.

Structure

Team assignments; approved-item checklist; quantity plan; bag/crate allocation; buy/borrow list; departure, arrival, and return checks.

Build Requirements

Create categories for approved service gear, uniforms, electronics, authorized trip documents, and event supplies. Keep food-temperature, hazardous-item, and specialist transport requirements as source-verification tasks; do not invent them.

List fields: ItemID, Traveler, Item, QuantityNeeded, QuantityOwned, ToAcquire, Bag, PackedQty, Status. ToAcquire=MAX(0,Needed-Owned). RemainingToPack=MAX(0,Needed-PackedQty). Packed is not the same as owned. Keep units clear for pairs, pieces, and containers.

Use the confirmed event scope and approved reuse/return plan to justify quantities instead of copying a universal list. Verify current airline/operator and destination restrictions through official sources before giving limits; do not invent baggage weights or liquid rules.

Test needed service trays 5, owned 3, packed 2: acquire 2 and still pack 3 in total. Add a last-minute list for chargers, authorized documents, and other approved items still in use. Include a bag-weight worksheet only with user-entered weights and a return-to-site unload, condition-check, and storage checklist.

Acceptance Check

The quantity example shows two to acquire and three to pack; shared items have one assigned traveler. 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 p1409. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the list to [destination/activities] and [trip length].

Change the workflow: Create a carry-on-only version using verified carrier limits.

Add a useful view: Add traveler-specific and shared-item views to prevent duplicate packing.

Check the inputs. Keep owned and packed quantities separate.

Check the workflow. Use pairs versus individual pieces consistently.

Check the result. Verify baggage and restricted-item rules for the actual operators.

P078 Catering-client gift, staff appreciation, or event-favor plan

Catering, guests and commercial relationships

Source project: Gift Planning Project. Master pp 1089, 1427-1428, 3764. Restaurant application below is an editorial adaptation.

Restaurant application

Map recipient/occasion to explicitly authorized business relationships and events; retain budgets, shipping, purchased/arrived/given states, and privacy views.

Source based lesson (restaurant wording adapted)

Plan authorized restaurant client-appreciation items, staff recognition, or event favors with stated preferences, approved budgets, ordering, and delivery checks.

Gather and build

Recipient, occasion date, preferences, budget, order deadline. Choose a gift within budget; allow delivery time; record purchased and delivered separately.

Check before relying on it

A shipped status does not prove delivery. Review the next month and set one realistic purchasing deadline.

Source based practice with fictional inputs

An authorized restaurant client-appreciation item is ordered, but no delivery confirmation is available.

Worked result: Status is ordered or shipped as supported, not delivered. Track the actual events separately.

Try a changed condition: The tracking record confirms delivery to the wrong address.

Check your answer: Record the exception and follow-up; do not treat the intended recipient as having received it.

Restaurant boundary

Not automatic spending or sending authority; do not disclose private preferences unnecessarily. Gifts and completed deliveries remain distinct.

P078 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 authorized client-appreciation, staff-recognition, or event-favor occasions are included; recipients; known dates; useful stated preferences; approved total/per-item budget; ordering/making/delivery lead times; and privacy boundaries.

Structure

Occasions: OccasionID, Person, Occasion, EventDate, Budget, Idea, DecisionDate, OrderDate, DeliveryDeadline, Status, ActualCost. Add Ideas and BudgetSummary.

Build Requirements

Use exact upcoming event dates rather than assuming a birth year is needed. Store only useful preference notes. Status Idea/Chosen/Ordered/Making/Ready/Given/Skipped. A purchase is not the same as delivery or giving.

Create reminder dates from user-chosen lead times, keeping order date and required arrival distinct. For recurring birthdays, ask how to handle February 29 in non-leap years rather than silently choosing. Provide a simple next-90-days view.

Budget summary separates planned allocation, committed purchase cost, and actual spending. Include shipping and materials if selected and avoid counting deposits twice. Example: a gift 25 plus shipping 5 totals 30 against budget 40, leaving 10.

Add a gift-idea bank linked to person and occasion, a handmade-gift task checklist, and an optional private note draft. Do not send messages or place orders. Include a discreet shared-view option that hides gift details from recipients and an after-event review to record what was given.

Acceptance Check

The sample total is 30 with 10 remaining; upcoming dates and gift privacy are handled explicitly. 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 p1429. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the planner to [occasions] and [annual budget].

Change the workflow: Add handmade-gift tasks with materials and realistic completion dates.

Add a useful view: Create a shared reminder view that hides surprise gift details.

Check the inputs. Separate ordered, arrived, and given states.

Check the workflow. Include shipping or materials consistently.

Check the result. Choose an explicit policy for recurring leap-day dates.

P079 Restaurant-hosted wedding and wedding-catering coordination workspace

Catering, guests and commercial relationships

Source project: Wedding Planning Project. Master pp 1090, 1431-1432, 3776. Restaurant application below is an editorial adaptation.

Restaurant application

Map couple priorities to confirmed client requirements and restaurant delivery scope; retain RSVPs, party/meal counts, vendor scopes, deposits, balance, run sheet, and fallback owners.

Source based lesson (restaurant wording adapted)

Coordinate a restaurant-hosted wedding or wedding-catering assignment with confirmed client priorities, party/guest counts, vendors, costs, and an event-day responsibility plan.

Gather and build

Priorities, guest count, budget, venue constraints, vendor commitments. Set priorities before shopping; track deposits and balances; build an event-day responsibility list.

Check before relying on it

Do not count a deposit as both an additional cost and part of the original total. Review the largest commitments and confirm what remains undecided.

Source based practice with fictional inputs

A restaurant wedding-event vendor total is $1,200; a $300 deposit has been paid.

Worked result: Remaining balance is $900, assuming no changes. Treat the deposit as part-payment of the agreed total.

Try a changed condition: A separately approved $150 addition is made.

Check your answer: Revised total is $1,350 and remaining balance is $1,050.

Restaurant boundary

No booking, contract acceptance, or invented cancellation rights. Dietary/access details need consent, minimal sharing, and approved handling outside the generic planner.

P079 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-hosted wedding or wedding-catering service is in scope; the client couple's confirmed priorities; date/venue/guest count; restaurant scope and exclusions; approved budget; decision owners; and supplied dietary, accessibility, cultural, and practical requirements.

Structure

Priorities; planning timeline; guest/RSVP tracker; vendor comparison; budget/payment log; decision register; ceremony/reception run sheet; contingency and handoff.

Build Requirements

Begin with the couple's priorities and distinguish confirmed choices from options. Build a timeline around actual venue/vendor requirements, not universal deadlines. Assign one owner and decision deadline to each major item.

Guests fields: GuestID, PartyID, Name, InvitationStatus, RSVP, AttendanceCount, MealChoice, AccessNeedsRef, Table. Count invited people separately from invitation parties and avoid storing unnecessary sensitive details in shared copies. Vendor fields include scope, quote, contract reference, deposit, balance, due date, and contact.

Budget uses category, planned amount, agreed total, paid, remaining, and contingency. A deposit belongs within the agreed total. Example: vendor total 2,000, paid deposit 500 gives remaining 1,500, not a 2,500 total. Keep gifts or expected contributions unconfirmed until actually committed.

Create an event-day run sheet with time, action, location, owner, supplier contact, setup dependency, and fallback. Include meals/breaks for key helpers, transport, setup/teardown, and a designated issue contact. Draft vendor confirmation questions and a final handoff checklist. Do not book services or invent contract cancellation rights; use supplied terms.

Acceptance Check

The budget example leaves 1,500, every major decision has an owner, and the run sheet includes dependencies and fallback. 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 p1433. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the plan to [guest count/style/cultural needs] and [budget].

Change the workflow: Create a smaller-event version focused on the couple's three priorities.

Add a useful view: Add a final-week handoff view for vendors and designated helpers.

Check the inputs. Count people and invitation parties separately.

Check the workflow. Avoid double counting deposits and unconfirmed contributions.

Check the result. Verify access, setup, cancellation, and payment terms from actual agreements.

P080 Restaurant launch, tasting, private dining, or catering event plan

Catering, guests and commercial relationships

Source project: Event Planning Project. Master pp 1091, 1435-1436, 3788. Restaurant application below is an editorial adaptation.

Restaurant application

Use event purpose, guest/party counts, unit-based supplies, owned tasks, budget, run of show, cleanup, returns, and contingency decisions.

Source based lesson (restaurant wording adapted)

Plan a restaurant launch, tasting, fundraiser, private-dining event, or catered gathering with confirmed guests, tasks, supplies, budget, and an event-day checklist.

Gather and build

Purpose, guest count, location, supplies, tasks, budget. Create a simple schedule; assign setup and cleanup; track confirmations and contingency choices.

Check before relying on it

Invited, confirmed, and attended are different totals. Walk through the event from arrival to cleanup and resolve the first gap.

Source based practice with fictional inputs

Thirty people are invited to a restaurant event, eighteen confirm, and sixteen attend.

Worked result: Report 30 invited, 18 confirmed, and 16 attended. Use separate event counts with clear definitions.

Try a changed condition: Two confirmed guests cancel before the event.

Check your answer: Update the confirmation state; retain the response history.

Restaurant boundary

Supply quantities are labeled assumptions, not safety/capacity approvals. Venue and food-service requirements must be verified.

P080 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 launch, tasting, private-dining, fundraiser, or catering event is planned; desired outcome; confirmed date/venue; guest/party counts; supplied needs; budget; team capacity; and approved contingencies.

Structure

Event brief; guest list; task timeline; menu/supply plan; budget; run-of-show; contingency plan; cleanup and closeout.

Build Requirements

Define the essential experience and three priorities. Build a task list with owner, deadline, dependency, and status. Separate confirmed venue arrangements from proposed options. Use actual venue rules and current local requirements where relevant rather than inventing restrictions.

Guest fields: GuestID, Name, RSVP, PartySize, DietaryOrAccessNotesRef. Supply list: Item, Unit, Needed, OnHand, ToBuy, Owner. ToBuy=MAX(0,Needed-OnHand), preserving units. Do not assume dietary requirements from identity; use information guests provide.

Create a budget with planned, committed, paid, and remaining amounts. Write a run-of-show covering setup, arrivals, food/activity timing, breaks, cleanup, and returns. Assign a fallback for weather or key-supplier delay and state the decision time and owner.

Test 20 expected guests and 12 suitable cups already available: buy 8 if one per guest is the chosen assumption. Label that assumption and adjust for reusable service or multiple drinks as requested. Draft an invitation and reminder with actual details or visible placeholders; do not send them. Include a post-event expense and borrowed-item reconciliation.

Acceptance Check

The supply example gives eight to acquire; tasks, contingencies, cleanup, and returns have owners. 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 p1437. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the plan to [occasion/guest count] within [budget].

Change the workflow: Create an indoor fallback for [weather concern] with a decision deadline.

Add a useful view: Add a borrowed/rented item return checklist with owners and due dates.

Check the inputs. Label quantity assumptions and keep units consistent.

Check the workflow. Confirm venue access and supplier arrangements before treating them as settled.

Check the result. Assign cleanup and returns before the event begins.

P093 Restaurant community-kitchen, fundraiser, or volunteer-session record

Catering, guests and commercial relationships

Source project: Community Involvement Project. Master pp 1104, 1487-1488, 3944. Restaurant application below is an editorial adaptation.

Restaurant application

Use actual host-organization commitments, approved roles, service hours, breaks, evidence, verification state, and permitted public descriptions.

Source based lesson (restaurant wording adapted)

Plan restaurant community-kitchen, fundraiser, or other actual volunteer commitments and retain accurate activity, eligible-hour, and verification records.

Gather and build

Organization, date, activity, start and end, breaks, verifier. Log actual service; retain verification where required; distinguish travel from accepted service hours.

Check before relying on it

Follow the receiving organization's definition of eligible hours. Request verification while the activity is still easy to confirm.

Source based practice with fictional inputs

A restaurant team's community-kitchen activity lasts three hours on site, including a 30-minute excluded break under the host organization's stated rule.

Worked result: Eligible time is 2.5 hours. Apply the receiving organization's actual definition and retain verification.

Try a changed condition: The organization confirms that one hour of travel is eligible for this activity.

Check your answer: Eligible total becomes 3.5 hours under that verified rule.

Restaurant boundary

Do not invent accepted volunteer hours, certifications, eligibility, food safety training, or beneficiary stories. Travel eligibility depends on the organization.

P093 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 community-kitchen, fundraiser, or volunteer host organization and roles are involved; confirmed commitments; availability/access needs; actual training/permission requirements; eligible-hour definitions; and verification evidence.

Structure

Opportunities: OpportunityID, Organization, Role, ContactRef, Requirements, Status. Sessions: SessionID, OpportunityID, Date, Start, End, BreakMinutes, Hours, Activity, VerificationStatus, VerificationRef, Verifier. Add Commitments and Reflection.

Build Requirements

Separate opportunities from accepted commitments. Record requirements and training from the actual organization, not assumptions. Do not claim eligibility, clearance, or completed hours without evidence.

Use full date-times if sessions can cross midnight. Hours=(End-Start)*24-BreakMinutes/60, only when inputs are complete and valid. Planned and completed hours must be reported separately. VerificationStatus Pending/Confirmed/Not required uses the program's supplied rules.

Create an availability plan, a contact question list, and a session reflection on contribution and learning. Keep beneficiary identities and sensitive stories out of the tracker. Any public account of work must respect actual permissions and confidentiality requirements.

Test a 09:00-12:00 session with 15 minutes break gives 2.75 hours. If verification is pending, the hours are recorded but not included in a Confirmed-only report. Provide a monthly report with organization, dates, activities, hours, and verification reference; avoid generating a false certificate or supervisor signature.

Acceptance Check

The sample is 2.75 hours and pending verification is excluded from confirmed-only totals. 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.

E07: Append VerificationRef/Verifier if confirmed hours are reported, using actual organization rules rather than a generated certificate.

Source extensions and troubleshooting

Master p1489. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the report to these supplied program requirements: [paste].

Change the workflow: Create a commitment calendar within [available hours].

Add a useful view: Show planned, completed, and verified hours as separate totals.

Check the inputs. Do not count planned sessions as completed service.

Check the workflow. Keep confidential beneficiary details out of notes.

Check the result. Never invent a verifier, signature, or confirmation.

Restaurant delivery and catering records

Adapted from master pp 1925, 1930, 1940, 1949-1960, 1964, 1987-1989. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

What transfers from the driving material

Use the source recordkeeping methods for a restaurant delivery service, catering runner or industry delivery business. The adaptation is administrative: complete time, payment timing, costs, order references, waiting and exceptions. It does not assert current platform eligibility, insurance coverage, employment status, tax treatment or delivery operating rules.

Compare the full work period

Source p1925: choose one reporting week, define included activities, record receipts and refunds, and separate estimates. Count preparation, travel, waiting, loading, service and administration. Compare like definitions; do not compare one service's active-time metric with another's full elapsed work time.

Keep operational and official records separate

Source p1930 distinguishes operational records from the source documents needed for bookkeeping or tax preparation. An operational mile is not automatically deductible. Use order references rather than customer identities in learning copies; do not retain customer addresses, access codes or payment details in the workbook.

A useful delivery record

Adapted from pp1949, 1953 and 1989: Date; order or job reference; agreed service scope; start/end/break times; travel; waiting/loading; service time; administration time; completed jobs; operational miles; receipts; reimbursement; cash costs; chosen reserve; unresolved adjustment; handoff issue; next improvement. Count overlapping sessions only once.

Keep money definitions explicit

Source p1940: cash operating surplus equals work receipts minus included cash operating costs. Planning surplus subtracts the chosen wear/replacement reserve as well. Divide by total work hours, including waiting and repositioning. A reserve is a planning allocation, not the same as an expense or tax deduction. Do not subtract fees twice when the starting amount is already net of them.

Original worked example

Source p1964 uses five total hours, $120 earnings after platform fees, $5 reimbursement, 90 operational miles, $16 fuel, $5 tolls, $4 other allocated cash costs and an $18 reserve. Receipts are $125; cash costs $25; cash surplus $100; planning surplus $82; planning surplus per full work hour $16.40; receipts per operational mile about $1.39. These are fictional inputs, not expected restaurant delivery earnings.

Payout reconciliation

Source p1952: expected closing unpaid balance = opening unpaid balance + statement earnings + reimbursements + adjustments - separately charged cash-out fees - deposits received. Include an amount only once, match statement periods to deposit timing, explain any difference and record the support reference and next action.

Supporting records

Sources pp1956-1960: maintain service/concern records using manufacturer or qualified advice; record incidents factually without unnecessary guest data; verify renewal/requirement sources and dates; compare required versus optional equipment with a spending cap; test one improvement and retain costs, time and comparable conditions.

Compare a delivery opportunity

Source p1987: compare confirmed local availability, official source/date, time windows, vehicle/equipment, screening/training, upfront/ongoing costs, payment timing, cancellation terms and total work hours. The original named-platform cards are dated examples, not proof of current openings. Reverify any service before applying or committing.

Restaurant handoff boundary

Use the restaurant's current approved order-handoff, packaging, incident and delivery procedures. The retained principles are matching the order reference, recording discrepancies, confirming completion and escalating exceptions. This edition supplies no food-transport temperature, alcohol handoff, substitution, refund or driving-safety procedure.