Booking.com Extranet Listing Automation: Workflow, API Schema, and Validation — Prepared by Macari Company

Executive Summary

This proposal focuses on the system workflow, API payload schema, and constraints/validation required to automate hotel property listing creation on the Booking.com Extranet. It outlines a practical, auditable flow from Excel-based data ingestion to field mapping, validation, and browser automation, with optional AI-assisted enrichment gated by human review. A detailed, cloud-agnostic technology stack is proposed to ensure reliability, traceability, and scalability while meeting the RFP's goals to reduce manual entry time, increase field accuracy, and enable batch processing.

- Scope centers on workflow design, payload specification, and validation/controls.

- Emphasis on accuracy, auditability, and safe automation for the Booking.com Extranet.

- Detailed, vendor-neutral technology stack aligned with reliability and scale.

Outcome	Objective
Browser automation with validated payloads and audit trails	Automate listings end-to-end
Structured mapping, strict validation, and retry/fallback logic	Reduce time and errors
Confidence thresholds with human-in-the-loop review	Enable safe AI use

Company Introduction

Macari Company prepares this focused response on workflow, payload schema, and validation controls for Booking.com Extranet listing automation. The team functions bilingually (Arabic and English) and emphasizes clear process orchestration, robust data handling, and controllable automation aligned to operational goals.

- Brand: Macari Company

- Languages: Arabic, English

- Focus in this response: workflow, payload, validation

Details	Attribute
Macari Company	Brand Name
Arabic, English	Operating Languages
System workflow, API payload, constraints/validation	Proposal Focus

Understanding of the RFP and Objectives

The RFP seeks a robust system workflow for automating Booking.com Extranet listings, a clear API-style payload definition for property data, and concrete validation/constraint rules to control risk, accuracy, and authorization. We align our approach to: (1) define a staged, auditable workflow; (2) formalize property payload structures that map internal form fields to extranet inputs; and (3) enforce validations, human review thresholds, and access controls to minimize errors while accelerating time-to-listing.

- System Workflow: user input → parse/validate → mapping → automation → audit

- Payload: structured property JSON aligned to extranet field needs

- Constraints: authorization, mandatory fields, confidence thresholds, fallbacks

Our Alignment	RFP Focus Area
Modular, auditable stages with retries and human review	Workflow
Typed JSON with rooms, amenities, policies, images	API Payload
Schema + business rules, RBAC, and error routing	Validation

Technical Approach and Methodology

We propose a modular service that ingests Excel data, normalizes and validates it, generates a compliant property payload, and drives browser automation to the Booking.com Extranet. The system includes a mapping layer, strict validation with business rules, robust error reporting, and audit logging. Optional AI assists with enrichment only when confidence thresholds are met; otherwise records are routed for human review.

- Separation of concerns: ingest, validate, map, automate, audit

- Deterministic validation before any automation step

- Human-in-the-loop for low-confidence enrichment

Outcome	Key Function	Stage
Structured internal model	Read Excel/uploader and normalize	Ingestion
Clean, consistent payload or actionable errors	Schema + business rules + referential checks	Validation
Accurate extranet listing creation	Browser automation with session control	Automation

Framework Overview

The framework integrates a typed data layer, a mapping engine, and an automation driver. A rules engine enforces constraints, and an audit subsystem records every submission with diffs, timestamps, and operator IDs. A queue decouples ingestion from automation, enabling retries and batch operations.

- Typed models for predictable mappings

- Rules engine for constraints and defaults

- Queue-driven automation with idempotency

Interaction	Purpose	Module
Validates and serializes payload	Define property/room/policy/images	Data Models
Produces field-level instructions	Map internal keys to extranet fields	Mapping Engine
Executes mapped actions safely	Login, navigate, input, upload	Automation Driver

Phased Methodology

Delivery is phased to reduce risk: define the workflow and payload first, then implement validation and automation, and finally harden reliability with retries, monitoring, and UAT. This mirrors the RFP timeline while concentrating on sections covering workflow, payload, and constraints.

- Discovery: confirm fields, mappings, and rules

- Build: payload schema, validation, automation

- UAT: accuracy checks and operator feedback

Primary Deliverable	Focus	Phase
Workflow spec and mapping dictionary	Field catalog, mappings, workflow	Discovery
Working prototype with audit logs	Schema, validation, automation driver	Implementation
UAT sign-off and runbooks	Edge-cases, retries, observability	UAT & Hardening

Methodological Pillars

Our methodology prioritizes determinism, transparency, and safety. Deterministic validations prevent malformed submissions. Transparency is ensured via full audit logging. Safety comes from role-based controls, confidence thresholds, and structured fallbacks.

- Deterministic validation before automation

- Full auditability and traceability

- Guardrails for AI and permissions

Practices	Pillar
Schema validation, retries, idempotency	Reliability
Audit logs, mapping references, diffs	Traceability
RBAC, confidence thresholds, human review	Safety

Project Architecture

A service-oriented architecture decouples ingestion/validation from automation. Data enters via an uploader or Excel import, passes through schema and business-rule validators, is transformed to a property payload, and is executed by an automation driver against the extranet. Observability and audit services provide end-to-end visibility.

- Service boundaries for resilience and scale

- Clear data contracts between components

- Observability first for rapid diagnosis

Responsibility	Components	Tier
Intake and operator control	Uploader/UI, API gateway	Interface
Correctness and business logic	Mapper, validators, rules engine	Core
Extranet interactions	Browser automation worker	Automation

System Components

Core components include an ingestion service for Excel/uploads, a mapping engine correlating internal form keys to extranet fields, a validation layer combining schema and business rules, an automation worker handling login, navigation, inputs, and image uploads, and an audit/logging subsystem. Optional AI enrichment is isolated and gated by confidence thresholds and human review.

- Ingestion: Excel/uploader to normalized models

- Mapping: dictionary-driven field alignment

- Automation: session management, inputs, uploads

Component	Key Functions	Notes
Ingestion Service	Parse Excel, normalize, dedupe	Supports batch processing
Validation Layer	Schema + business rules + referential checks	Stops on mandatory field failures
Automation Worker	Login, form fill, image upload, submit	Retry on transient errors
Audit & Monitoring	Event logs, metrics, alerts	Compliance and support
AI Enrichment (Optional)	Fill missing data under guardrails	Human-in-the-loop if low confidence

Data Flow & Integration

The workflow proceeds as follows: (1) User uploads or selects an internal Excel content form; (2) The system parses and normalizes data; (3) Validation enforces mandatory fields (e.g., name, location), types, and business rules; (4) A mapping engine translates internal keys to Booking.com Extranet fields; (5) A property payload is generated; (6) The automation worker logs in, navigates the extranet, inputs values, and uploads images; (7) Audit logs capture all actions and outcomes. Optional AI-assisted enrichment occurs before validation and requires human approval if below confidence thresholds. Example property payload structure is shown below.

- Strict validation before automation

- Mapping dictionary governs field alignment

- Full audit, with per-field status and timestamps

Output	Description	Step
Raw property dataset	Excel upload or form-based entry	User Input
Internal typed model	Normalize and type-cast	Parsing
Validated model or errors	Mandatory fields, formats, rules	Validation
Submission-ready payload	Form keys → extranet fields	Mapping
Created/updated listing	Login, fill, upload, submit	Automation

Technology Stack

The proposed stack is cloud-agnostic and emphasizes reliability, observability, and maintainability. It supports Excel ingestion, strict validation, resilient automation, and auditable operations. Alternatives are provided to fit the client's environment and preferences.

- Cloud-agnostic components to support hybrid or cloud deployment

- Typed models and validation-first design

- Resilient, observable automation with safe fallbacks

Purpose	Alternatives	Primary Option	Layer
REST APIs for ingestion, validation, orchestration	Node.js (NestJS), Java (Spring Boot)	Python (FastAPI)	Backend Framework
Robust Excel ingestion and normalization	Apache Arrow, openpyxl	Pandas (Excel/CSV)	Data Parsing
Schema and type validation; business rules layering	Marshmallow, Cerberus	Pydantic (typed models)	Validation
Business constraints, defaults, and gating	Drools (Java), json-rules-engine	Custom rule layer (Python)	Rules Engine
Booking.com Extranet login, navigation, inputs, uploads	Selenium, Puppeteer	Playwright (headless, Python)	Automation
Async jobs, retries, rate control, idempotency	RabbitMQ, Kafka	Celery + Redis	Queue/Workers
Mappings, job states, audit references	MySQL	PostgreSQL	Storage (Relational)

Image files, logs, artifacts	MinIO (on-prem)	S3-compatible storage	Object Storage
Centralized, searchable logs and audit trails	ELK Stack	Structured JSON + OpenSearch	Logging
Metrics, dashboards, alerting	OpenTelemetry, Cloud vendor tools	Prometheus + Grafana	Monitoring
Exception visibility and triage	Open-source equivalents	Sentry	Error Tracking
Build, test, security scans, deploy	GitLab CI, Jenkins	GitHub Actions	CI/CD
Packaging services and workers	Podman	Docker	Containerization
Scalable deployment of services	Docker Swarm	Kubernetes	Orchestration
Manage credentials and session tokens	Doppler	Vault or cloud secrets manager	Secrets
Operator access control and audit linkage	OIDC/OAuth2 (SSO) if available	RBAC within backend	AuthN/AuthZ
Suggest values; gated by confidence thresholds	Local LLM with retrieval	Configurable provider via API	AI Enrichment (Optional)

Relevant Experience and Case Evidence

This response is scoped to workflow, payload, and validation. While specific case details are not provided in the digest, Macari Company applies industry-standard engineering patterns and bilingual delivery to ensure clarity, correctness, and operator usability for internal automation projects.

- Bilingual delivery (Arabic and English)

- Focus on deterministic, testable automation

- Emphasis on operator-centric workflows

Relevance	Area
Decoupled workers and idempotent jobs	Automation Patterns
Typed models, schema-first validation	Data Discipline
Human-in-the-loop for safety and accuracy	Usability

Project Team and Roles

A lean team structure ensures rapid delivery of the workflow, payload schema, and validation rules. Roles are defined to maintain separation of concerns across data, automation, and QA.

- Technical Lead: overall architecture, quality gates

- Backend Engineer: APIs, validation, rules engine

- Automation Engineer: Playwright flows, resilience

Responsibilities	Role
Architecture, security, review, delivery oversight	Technical Lead
Ingestion, schema, validation, mapping	Backend Engineer
Extranet automation scripts and hardening	Automation Engineer
Test plans, regression, UAT support	QA Engineer
Schedules, stakeholder communication	Project Coordinator

Work Plan, Timeline, and Milestones

Aligned to the RFP's milestones, with emphasis on delivering workflow, payload, and constraint artifacts early for validation and sign-off. Timelines can be refined during discovery.

- Front-load workflow/payload definitions for clarity

- Parallelize validation and automation after mapping sign-off

- UAT-driven refinements before pilot

Duration	Description	Milestone
1 week	Finalize workflow, payload schema, constraints	Requirements Sign-off
2 weeks	Core ingestion, mapping, validation	Prototype Development
3 weeks	Extranet automation and session management	Integration & Automation Engine
2 weeks	End-to-end tests, operator feedback	Testing & UAT
1 week	Live run with 5 properties	Pilot Launch
1 week	Rollout to all properties	Full Deployment

Quality Assurance and Risk Management

Quality is enforced via schema validation, business-rule checks, and automated tests. Key risks include Booking.com UI changes and data quality issues. Mitigations include resilient selectors, health checks, retries, and human-in-the-loop routing when validation confidence is low.

- Unit/integration tests for mappings and validators

- Mocked automation flows for stability

- Monitoring with alerts on failures and drifts

Mitigation	Risk
Selector abstraction, monitoring, quick-fix playbooks	Booking.com UI changes
Pre-submit validations, enrichment gates, operator review	Data inconsistencies
Robust login flows, token refresh, backoff/retry	Session/auth failures

KPIs and Service Levels

KPIs measure time-to-listing, error rates, and throughput. SLAs cover job success rates, maximum retries, and response times for incident triage.

- Reduce manual time by 80% per listing

- Achieve >95% field accuracy for critical attributes

- Track listings per week per operator

Target	Baseline	KPI
≤ 0.5 hours	2.5 hours	Average time per listing
< 5%	12% correction requests	Listing error rate
Significant increase (to be confirmed in pilot)	Current baseline	Listings per operator/week

Data Privacy, Security, and IP

Access is role-based; only authorized users can trigger listing creation. Secrets are stored securely, and all actions are audited. Data flows are minimized to the least necessary scope, with sensitive

data masked in logs. IP for configurations, mappings, and code is to be defined in the final agreement.

- RBAC and least-privilege access

- Encrypted secrets and transport (TLS)

- Redacted logs and structured audits

Description	Control
RBAC for workflow triggering and approvals	Authorization
Managed via secure vaulting	Secrets
Per-field, per-action logging with timestamps	Audit

Compliance with RFP Requirements

Our solution delivers a concrete workflow, a typed property payload schema, and robust constraints aligned with the RFP's stated needs. It also supports human review when AI confidence is low and prevents unauthorized or incomplete submissions.

- Covers System Workflow with auditable stages

- Provides example property JSON payload

- Implements constraints and validation rules

Coverage	RFP Area
User input → parse → validate → map → automate → audit	System Workflow
Property, rooms, amenities, policies, images	API Payload
Mandatory checks, defaults, confidence thresholds, RBAC	Validation Rules

Deliverables Summary

Deliverables emphasize clarity and operational readiness for Sections 6–8: workflows, payload schema, and validation rules, along with a prototype automation that demonstrates end-to-end feasibility.

- Workflow specification and swimlane diagram (textual/visual)

- Property payload JSON schema and field mapping dictionary

- Validation/constraints matrix and error-handling strategy

Description	Deliverable
End-to-end stages, retries, and operator actions	Workflow & Runbooks
Typed property JSON with examples	Payload Schema
Mandatory/optional, thresholds, fallbacks	Validation Rules
Automation against staging with audit logs	Prototype

Assumptions

Assumptions clarify boundaries and responsibilities while focusing on Sections 6–8 outcomes.

- Access to Booking.com staging/sandbox for safe testing

- Internal Excel format is stable or versioned

- Image assets are available via URLs or uploader

Impact	Assumption
Enables safe automation testing	Staging access
Reliable mappings and validations	Stable form schema
Ensures end-to-end listing creation	Image availability

Pricing Approach (Summary)

Pricing for this scope is typically structured as a fixed fee for discovery and definition (workflow, payload, validation), followed by time-and-materials for implementation and UAT. Final pricing depends on confirmed requirements and environment constraints.

- Fixed-fee discovery for clarity and alignment

- T&M for build with capped sprints

- Pilot-based adjustment before full rollout

Pricing Model	Phase
Fixed Price	Discovery & Design
Time & Materials	Implementation
Time & Materials (with cap)	UAT & Pilot

Why Macari Company

Macari Company aligns the automation workflow, payload design, and safeguards to your operational goals. Our bilingual capability supports broader operator adoption, and our validation-first approach enhances accuracy while reducing manual workload.

- Workflow clarity and validation-first engineering

- Auditable, deterministic automation with guardrails

- Bilingual (Arabic/English) collaboration

Benefit	Strength
Higher accuracy and fewer corrections	Validation-first design
Traceable submissions and faster support	Auditability
Flexible deployment and scaling	Cloud-agnostic stack