BHM — Detailed Feature Documentation

BHM (Banquet & Hotel Management) is a full-stack Django web application for end-to-end hotel and restaurant property management. Below is an exhaustive, detailed breakdown of every feature, page, operation, data field, and workflow in the system.

Table of Contents

1. Authentication & Access Control

1.1 Login / Logout

- Login Page (`/accounts/login/`): Username + password authentication using Django's `auth.authenticate()`.
- Login Redirect Logic: After login, users are redirected to different dashboards based on their Django Group:

Group Name	Redirect Destination
`business-admin`	Hotel Dashboard
`hotel-admin`	Hotel Dashboard
`hotel-staff`	Hotel Dashboard
`hotel-front-desk`	Hotel Dashboard
`house-keeping`	Room List
`restro-admin`	Restaurant Dashboard
`restro-front-desk`	Restaurant Dashboard
`restro-waiter`	Table View (POS)
`restro-kitchen`	Table View (POS)

- Logout (`/accounts/logout/`): Ends session, redirects to login.
- Sign-in / Sign-out: Alternative endpoints (`/accounts/sign_in/`, `/accounts/sign_out/`) with session-based user ID tracking.

1.2 Middleware

- LoginRequiredMiddleware: Intercepts every request. If the user is not authenticated and the URL is not in the exempt list (`login`, `logout`, `admin`, `static`), redirects to login.
- PermissionDeniedRedirectMiddleware: Catches `PermissionDenied` exceptions globally; shows an error message ("You do not have permission…") and redirects to the previous page or dashboard.
- GroupRequiredMiddleware: Defines all 10 user roles. Currently configured for group-based routing logic.

1.3 Permission System

Every URL endpoint is wrapped with `permission_required_with_message()` — a custom decorator that checks Django model-level permissions (e.g., `hospitality.view_category`, `restaurant.change_order`). If permission is denied, it raises `PermissionDenied` which the middleware catches.

1.4 Role-Based Navigation

12 separate navigation templates exist for different roles:

Template	Role
---	---
`admin_nav.html`	Business Admin — full access
`hotel_admin.html`	Hotel Admin
`hotel_front_desk.html`	Hotel Front Desk
`hospitality_staff_nav.html`	Hotel Staff
`restro_admin_nav.html`	Restaurant Admin
`restro_front_desk.html`	Restaurant Front Desk
`waiter_navs.html`	Restaurant Waiter
`kitchen.html`	Kitchen Staff
`room_service.html`	Room Service

2. Dashboards

2.1 Hotel Dashboard (`/accounts/dashboard/`)

Displays real-time hotel operations data:

	Widget	Data	
	---	---	
	Total Rooms	Count of all active rooms (excluding OOO/OOS)	
	Occupancy %	`(occupied rooms / total rooms) × 100`, rounded to 2 decimal places	
	House Count	Sum of `no_of_adult` for all currently in-house reservations	
	Today's Arrivals	Count of reservations with `date_from = today`	
	Rooms by Category	Grouped count: Standard, Deluxe, Twin, Suite, Pent House, Presidential Suite	
	Rooms by Status	Grouped count across all 12 room statuses	
	Reservations by Status	Grouped count of in-house reservations by status	
	Today's Reservations	List of currently in-house reservations	

2.2 Restaurant Dashboard (`/accounts/dashboard_restro/`)

Displays real-time restaurant operations data:

	Widget	Data	
	---	---	
	Table Categories	Count of active table categories	
	Tables	Count of active tables	
	Dishes	Count of Kitchen-category items	
	Bar Items	Count of Bar-category items	
	Tables by Category	Grouped table count by category	
	Tables by Status	Grouped count: Available, Occupied, Reserved, Cleaning	
	Orders by Status	Grouped order item count by status (only active orders where `left_time` is null)	
	Top 10 Dishes	Top 10 most ordered kitchen items by quantity, with total revenue	
	Total Customers Visited	Sum of `no_of_customers` across all orders	
	Average Order Value	Average `total_amount` from restaurant bills	
	Total Sales	Sum of `total_amount` from all restaurant bills	

2.3 Staff Dashboard (`/accounts/staff_dashboard/`)

A dedicated lightweight dashboard for staff-role users.

3. Hospitality — Room Setup

3.1 Room Categories

Pages: List (`/hospitality/category/`), Add Form, Edit Form
CRUD: Create, Read, Update, Delete
Data Fields:

	Field	Type	Details	
	---	---	---	
	`title`	CharField(100)	Unique. e.g., Standard, Deluxe, Suite	
	`description`	TextField	Optional	
	`image`	ImageField	Uploaded to `hospitality/category/` with timestamped filename	

- List page shows room count per category via annotation.

3.2 Rooms

Pages: List (`/hospitality/room/`), Add Form, Edit Form
CRUD: Create, Read, Update, Delete
Data Fields:

	Field	Type	Details	
	---	---	---	
	`category`	FK → Category	Room type	
	`number`	CharField(100)	Unique room number (e.g., "101", "205")	
	`price`	Float	Nightly rate	
	`description`	TextField	Optional	
	`image`	ImageField	Optional	
	`floor`	Integer	Floor number	
	`capacity`	Integer	Max guest capacity	
	`accessible`	Boolean	Wheelchair accessible	
	`child_access`	Boolean	Children allowed	
	`smoking`	Boolean	Smoking allowed	
	`cleaning`	Boolean	`True` = Clean, `False` = Dirty	
	`occupancy`	Boolean	`True` = Available for clicking/posting, `False` = Not available	
	`room_status`	CharField	One of 12 statuses (see below)	

Room Status Values (12 states):
1. `**Vacant-Clean**` — Empty and cleaned (default)
2. `**Vacant-Dirty**` — Empty but needs cleaning
3. `**Arrival**` — Room assigned, guest expected
4. `**Occupied**` — Guest is checked in
5. `**Dirty**` — Needs cleaning
6. `**OOO**` — Out of Order (not sellable, needs repair)
7. `**OOS**` — Out of Service (temporary maintenance)
8. `**Due-Out**` — Guest scheduled to depart today
9. `**Departed**` — Guest has left
10. `**Overstay**` — Guest stayed past checkout date
11. `**Under-stay**` — Guest left before scheduled date
12. `**Stay-over**` — Guest extending stay

Room Status Toggle: The `update_cleaning` action flips the cleaning status and changes room_status between `Vacant-Clean` ↔ `Vacant-Dirty`.

Room Display (`/hospitality/display_rooms_status/`): Visual floor-wise room grid showing each room's status, current guest (if occupied), check-in/out dates, and nights remaining. Rooms are colored by status.

3.3 Meal Plans

Pages: List (`/hospitality/plan/`), Add Form, Edit Form
CRUD: Create, Read, Update, Delete
Data Fields:

	Field	Type	Details	
	---	---	---	
	`category`	FK → PlanCategory	e.g., Meal, Drinks	
	`title`	CharField(100)	Plan name	
	`description`	TextField	Optional	
	`price`	Float	Daily plan price	
	`image`	ImageField	Optional	

- Plans are linked to reservations. Price is multiplied by number of days during billing.
- A plan titled "None" means no meal plan is attached.

3.4 Packages

Pages: List (`/hospitality/package/`), Add Form, Edit Form
CRUD: Create, Read, Update, Delete
Data Fields:

	Field	Type	Details	
	---	---	---	
	`rooms`	ManyToMany → Room	Rooms included in package	
	`title`	CharField(100)	Package name	
	`description`	TextField	Optional	
	`price`	Float	Package price per day	
	`image`	ImageField	Optional	
	`category`	FK → Category	Optional room category	

- Packages bundle rooms together at a fixed price.
- A package titled "None" means no package — room price is used instead.
- Rooms by Category API (`/hospitality/rooms_by_category/<id>/`): JSON endpoint returning rooms for a given category, used by the package form's dynamic room selector.

4. Hospitality — Reservation Management

4.1 Reservation List

Page: `/hospitality/reservations/`
- Shows all reservations, ordered newest first.
- Date Filter: Filter by `date_from` using a date picker.
- Each reservation shows: confirmation no, customer name, category, room (if assigned), dates, status, actions.

4.2 Add Reservation

Page: `/hospitality/add_reservation/`
A multi-step reservation form that collects:

Step 1 — Guest Information:

	Field	Details	
	---	---	
	`first_name`, `middle_name`, `last_name`	Guest name	
	`mobile`	Phone number	
	`email`	Email address	
	`gender`	M/F	
	`dob`	Date of birth	
	`nationality`	Country	
	`address`	Street address	
	`id_card`	ID type (Passport, Driving License, etc.)	
	`id_number`	ID number	
	`company`	Company name	
	`customer_pan`	Tax PAN number	

Step 2 — Reservation Details:

	Field	Details	
	---	---	
	`date_from`	Check-in date (min: today)	
	`no_of_days`	Duration of stay	
	`no_of_rooms`	Number of rooms needed	
	`no_of_adult`	Number of adults	
	`no_of_child`	Number of children	
	`meal-plan`	Selected meal plan (dropdown of active plans)	
	`package`	Selected package (dropdown of active packages)	

Step 3 — Room Availability Check:
`/hospitality/check_room_availability/<date>/<days>/<rooms>/<persons>/<children>/<reserves>/<package_id>`

- Returns JSON with available room counts per category after accounting for existing reservations and date overlaps.
- Logic: For each category, counts total rooms, then subtracts rooms reserved in overlapping date ranges.
- If a package is selected, filters rooms to those in the package.

Step 4 — Additional Details:

	Field	Details	
	---	---	
	`reservation_class`	Class of reservation	
	`billing_instruction`	Billing notes	
	`payment_mode`	Cash, Card, Credit, etc.	
	`payment_mode_data`	Card/reference number	
	`business_source`	Walk-in, Online, Agent, etc.	
	`market_segment`	Corporate, Leisure, etc.	
	`referred_by`	Referral source	
	`reservation_mode`	Phone, Email, In-person, etc.	
	`reservation_type`	Residential / Residential & Program	
	`purpose`	Purpose of visit	
	`agent`	Travel agent name	
	`res_status`	Confirmed, Guaranteed, Non-Guaranteed, Tentative, WaitListed	

Step 5 — Requests (added during reservation):

	Field	Details	
	---	---	
	`request_type`	Request, Service, Alert, Suggestion	
	`request_title`	Title	
	`request_details`	Description	
	`service_type`	Chargeable, Non-Chargeable, Complementary	
	`price`	Charge amount	
	`remarks`	Additional notes	

Confirmation Number: Auto-generated 8-character alphanumeric code, checked for uniqueness.

4.3 Edit/Update Reservation

Page: `/hospitality/reservation/<resid>`
- Pre-fills all guest and reservation data.
- Updates customer information in-place (same customer record).

- Room availability re-checked via `/hospitality/check_room_availability_for_update/<resid>/<category>/<date>/<days>/<persons>/<children>`.

4.4 Cancel / Delete Reservation

- Cancel (`/hospitality/cancel_reservation/<resid>`): Sets `status = 'Cancelled'` — record persists.
- Delete (`/hospitality/delete_reservation/<resid>`): Permanently removes the reservation record.

4.5 Reservation Status Flow

Requested → Booked → Checked-In → In House → Checked-Out
↗
Cancelled ←── (from any state)
Extended → Continued → Shifted/Transferred

5. Hospitality — Check-in Process

5.1 Arrivals / Check-in Page

Page: `/hospitality/checkin/`
- Shows reservations arriving today (or filtered by date).
- Filters: `date_from >= today`, `check_in IS NULL`, excludes Cancelled.
- Each reservation row shows: confirmation no, guest name, category, dates, room (if assigned), status.

5.2 Room Assignment

Page: `/hospitality/assign_room/<resid>`
- Shows available rooms in the reservation's category that don't overlap existing bookings.
- Clicking a room triggers `/hospitality/room_assignment/<resid>/<rid>`:
  1. If a previous room was assigned → previous room is set to `Vacant-Clean`, `occupancy = True`.
  2. New room is assigned → `occupancy = False`.
  3. Room status set to `Arrival` (new assignment) or `Occupied` (transfer).
  4. Reservation price set from package price or room price.
  5. `allocated_by` set to the logged-in staff.
  6. Reservation status set to `Booked`.

5.3 Check-in Action

POST to `/hospitality/check_in`
Performs:
1. Sets `check_in = datetime.now()`.
2. Sets `checked_in_by` to current staff.
3. Sets `status = 'Checked-In'`.
4. Creates an Advance Payment record with the advance amount, payment method, and method data.
5. Processes any requests submitted with the check-in.
6. Updates room: `cleaning = False`, `room_status = 'Occupied'`.

5.4 Room Change

Page: `/hospitality/change_room/<resid>`
- Shows available rooms in same category.
- Reassigns room using the same `room_assignment` logic.

6. Hospitality — In-House Operations

6.1 In-House Reservations

Page: `/hospitality/inhouse/`
- Shows all currently checked-in guests (`check_in` set, `check_out` null).
- Date filter: Shows guests who were in-house on a specific date.

6.2 Booked Rooms / Posting

Page: `/hospitality/booked_rooms/`
- Shows active in-house reservations with options to post charges.
- Can filter by specific room ID.
- Lists available services for booking.

Posting Rooms (`/hospitality/posting_rooms/`): Shows in-house reservations with their room details for posting charges.

6.3 Options Hub (POST `/hospitality/options`)

Central handler that routes to different actions based on the `option` field:

Option	Action
`Request`	Add guest request(s)
`Service`	Book a service
`Payment`	Record a payment

6.4 Add Guest Request

Supports batch creation — multiple requests submitted simultaneously via array fields:

Field	Details
`request_type[]`	Array: Request, Service, Alert, Suggestion, Minibar, Laundry
`request_title[]`	Title for each request
`request_details[]`	Description
`price[]`	Charge per unit
`qty[]`	Quantity
`service_type[]`	Chargeable / Non-Chargeable / Complementary
`remarks[]`	Notes

Request Status Workflow: `Requested → Waiting → Approved → Accomplished → Cancelled`

6.5 Book a Service

Books a hotel service (spa, gym, etc.) for the guest:
- Links to the guest's reservation.
- Records service ID, frequency, date, and time.
- Initial status: `Requested`.

6.6 Add Payment

Records a payment against the reservation:
- Fields: confirmation_no, reservation, room, particular (description), rate, qty, remarks.

6.7 View Requests / Services / Payments

- Requests (`/hospitality/reservation_requests/<resid>`): Lists all requests for a reservation.
- Services (`/hospitality/reservation_services/<resid>`): Lists all booked services.
- Payments (`/hospitality/reservation_payments/<resid>`): Lists all payments.

6.8 Cancel / Delete Requests & Services

- Cancel Request: Sets `status = 'Cancelled'` (record kept, excluded from billing).
- Delete Request: Permanently removes the request.
- Cancel Service: Sets booking `status = 'Cancelled'`.
- Delete Service: Permanently removes the booking.

6.9 Add Days (Stay Extension)

URL: `/hospitality/add_days/<resid>/<nod>`
1. Checks room availability for the extension period.
2. If available, extends `date_to` and increments `no_of_days`.
3. If the specific room is already booked by another guest for the new dates, shows error.

6.10 Add Guests

URL: `/hospitality/add_guests/<resid>/<nog>`
1. Increments `no_of_adult` by `nog`.
2. Creates a `Request` record with type "Request", title "Guest-Add".
3. Price is pulled from `PrimaryData` (key: `guest-add`).

---

7. Hospitality — Folio, Billing & Checkout

7.1 Guest Folio

Page: `/hospitality/folio/<resid>`
A comprehensive financial summary page showing:

	Section	Details	
	---	---	
	Room/Package Charges	Room price × nights or Package price × nights	
	Meal Plan Charges	Plan price × nights	
	Service Bookings	Service price × frequency for each booking	
	Restaurant Bar Orders	All bar items ordered by the guest (from linked restaurant bills)	
	Restaurant Food Orders	All food items ordered by the guest	
	Payments Made	List of all payments: advance, installments	
	Bill Status	Whether a final bill has been generated	

7.2 Prepare Bill

Page: `/hospitality/bill/<resid>`
Calculates the final bill by summing:

1. Room or Package Charges: `(room.price or package.price) × no_of_days`
2. Meal Plan Charges: `plan.price × no_of_days`
3. Guest Requests (non-cancelled): `price × qty` for each. Guest-add uses the configurable `PrimaryData` rate.
4. Service Bookings (non-cancelled): `price × frequency`
5. Restaurant Bills: Sum of `final_amount` from linked restaurant bills.
6. Less: Payments: Sum of all payment rates.

Tax Calculation:
- VAT Rate: Read from `PrimaryData` (key: `vat`), applied as % of taxable amount.
- Service Charge Rate: Read from `PrimaryData` (key: `service-charge`).
- Fiscal Year: Read from `PrimaryData` (key: `year`), e.g., `2082`.
- Formula: `final_amount = taxable_amount + VAT + service_charge`
- Taxable Amount: `total_amount − discount`

7.3 Save Bill

POST to `/hospitality/save_bill/`

Bill Data Fields:

	Field	Type	Details	
	---	---	---	
	`invoice`	CharField	Auto-generated: `{sequence:04d}/{fiscal_year}` (e.g., `0001/2082`)	
	`confirmation_no`	CharField	From reservation	
	`customer`	FK → Customer	Guest	
	`room`	FK → Room	Room	
	`fiscal_year`	Integer	From PrimaryData	
	`date` / `time`	Date/Time	Current date & time	
	`name`, `company`, `pan`, `address`, `phone`, `email`	CharFields	Billing details	
	`total_amount`	Float	Auto-calculated sum of all charges	
	`discount`	Float	User-entered discount	
	`taxable_amount`	Float	`total_amount − discount`	
	`tax`	Float	`taxable_amount × (VAT% / 100)`	
	`service_charge`	Float	Configurable charge	
	`final_amount`	Float	`taxable_amount + tax + service_charge`	
	`payment_method`	CharField	Cash, Card, Credit, etc.	
	`payment_method_data`	CharField	Card/QR reference	
	`paid`	Float	Amount paid	
	`due`	Float	`final_amount − paid`	
	`paid_status`	CharField	`Clear` (fully paid), `Partial`, `Credit`	

Credit Payment Handling: If payment method is "Credit":
- Records `paid` and `due` amounts separately.
- Creates a Payment record with `particular = 'Payment Installment'`.
- Sets `paid_status = 'Credit'`.

7.4 Bills List

Page: `/hospitality/bills/`
- Lists all generated bills ordered by date (newest first).
- View Bill redirects to the folio page.

7.5 Checkout

Page: `/hospitality/checkout/`
- Lists all guests who are checked-in but not checked-out.
- Shows whether the bill has been settled.

Checkout Action (`/hospitality/do_checkout/<resid>`):
1. Validates that a bill exists. If not → error "Please complete the bill payment before checking out."

2. Sets `check_out = datetime.now()`.
3. Sets `status = 'Checked-Out'`.
4. Sets `checked_out_by` to current staff.
5. Updates room: `cleaning = False`, `room_status = 'Vacant-Dirty'`, `occupancy = True`.

---

8. Hospitality — Night Audit

8.1 Night Audit — Detailed View

Page: `/hospitality/night_audit?date=YYYY-MM-DD`
Shows all financial transactions for a given date for verification:

	Category	Source	
	---	---	
	Reservations	Reservations billed on that date (where package = None)	
	Meal Plans	Reservations billed on that date (where plan ≠ None)	
	Packages	Reservations billed on that date (where package ≠ None)	
	Requests	Requests created on that date	
	Bookings	Service bookings on that date	
	Orders	Restaurant order items from orders billed on that date	
	Payments	Payments created on that date	

|

Each item shows its audit status (✓ audited / ✗ not audited). Staff can select unaudited items and mark them as audited.

8.2 Night Audit — Category Summary

Page: `/hospitality/night_audit_category?date=YYYY-MM-DD`
Shows a summary table with counts per category:

	Category	Total Count	Audited Count	
	---	---	---	
	Reservation	n	m	
	Plan	n	m	
	Package	n	m	
	Request	n	m	
	Booking	n	m	
	Order	n	m	
	Payment	n	m	
	Exchange	n	m	
	MiniBar	n	m	
	Laundry	n	m	

8.3 Night Audit — Category-Wise Detail

Page: `/hospitality/night_audit_category_wise/<type>/<date>`
Drill-down into each category (type 0–9) showing individual transactions with:
- ID, confirmation code, topic/description, price, qty, amount, audit status.

8.4 Perform Audit

- Detailed Audit (POST `/hospitality/do_night_audit`): Marks selected transactions as audited, creates `NightAudit` records.
- Category-Wise Audit (POST `/hospitality/do_night_audit_category_wise`): Same but for category-wise view.

NightAudit Record Fields: `transaction_type`, `transaction_id`, `audited` (bool), `date`, `audited_datetime`, `staff`.

---

9. Hospitality — Foreign Exchange

9.1 Exchange List

Page: `/hospitality/foreign_exchange/`
- Lists all foreign exchange transactions, newest first.

9.2 Exchange Form

Page: `/hospitality/foreign_exchange_form/`
- Shows in-house reservations dropdown.
- Fields: reservation, currency, rate, amount.
- `exchanged_amount = rate × amount` (auto-calculated).
- Links exchange to both reservation and customer.
- Records performing staff.

ForeignExchange Fields: `reservation`, `customer`, `currency`, `rate`, `amount`, `exchanged_amount`, `datetime`, `staff`.

---

10. Restaurant — Menu Setup

10.1 Categories

Pages: List, Add, Edit, Delete
Fields: `title` (unique), `caption`, `description`, `image`.
Examples: "Kitchen" (food), "Bar" (drinks).

10.2 Classifications

Pages: List, Add, Edit, Delete
Fields: `category` (FK), `genre`, `species`.
Multi-dimensional classification system. Examples:

	Category	Genre	Species	
	Bar	Alcoholic	Beer, Vodka, Whiskey, Rum	
	Bar	Non-Alcoholic	Cold-drink, Energy-drink	
	Kitchen	Cuisine	Nepalese, Italian, Japanese, Fast-Food	
	Kitchen	Menu	Starter, Main Course, Dessert, Beverages	
	Kitchen	Veg/Non-veg	Paneer, Chicken, Mutton, Buff, Pork	
	Kitchen	Format	Single, Combo, Platter, Meal	

10.3 Menu Items

Pages: List, Add, Edit, Delete
Fields:

	Field	Details	
	`tracking_no`	Auto-generated: `BAR-{hex}` or `Dish-{hex}` (UUID-based, 10 chars)	
	`category`	FK → Category	
	`classification`	ManyToMany → Classification	
	`name`	Item name	
	`price`	Price	
	`ingradient`	Text description of ingredients	
	`description`	Full description	
	`image`	Item image	
	`remarks`	Notes	

10.4 Courses (Sub-items)

Fields: `classification` (M2M), `item` (FK → Item), `title`, `name`, `price`, `image`, `ingradient`, `description`.

Used for combo menus where one item contains multiple courses/components.

Related Delete Operations: Delete Course, Delete Ingredient, Delete Course Ingredient — all available on the item edit form.

---

11. Restaurant — Table Management

11.1 Table Categories

Pages: List, Add, Edit, Delete
Fields: `title`, `caption`, `description`, `image`.
Examples: Indoor, Outdoor, VIP, Bar Counter.

11.2 Tables

Pages: List, Add, Edit, Delete
Fields:

	Field	Details	
	`number`	Unique table number	
	`floor`	Floor location	
	`category`	FK → TableCategory	
	`capacity`	Max seating	
	`description`	Optional	
	`image`	Optional	
	`occupancy_status`	Available, Occupied, Reserved, Cleaning	

11.3 Table View (POS)

Page: `/restaurant/table_tmp/`
Visual layout of all tables showing their status. Click a table to view details.

11.4 Table Details

Page: `/restaurant/tabledetails_tmp/<id>`
Shows the selected table's:
- Current order and all order items.
- Option to place new orders, modify existing, cancel items.
- Billing and checkout controls.

11.5 Table Reservation

Reserve Table (`/restaurant/reserve_table/`): Creates a table reservation.
Cancel Reservation (`/restaurant/cancel_reservation/<id>`): Cancels the reservation.
Reservation Create (`/restaurant/reservation_create/`): Form submission for table reservation.

11.6 Table Merge

URL: `/restaurant/merge_table/`

Merges multiple tables into a single order. The merged table IDs are stored in `Order.merged_table`.

11.7 Update Table Status

URL: `/restaurant/update_available_status/<id>`
Toggles table status between Available/Occupied/etc.

---

12. Restaurant — Order Management

12.1 Order Model

Fields:

	Field	Details	
	---	---	
	`order_no`	Auto-generated: `ORDER-{hex}` (UUID-based)	
	`customer`	FK → Customer (optional)	
	`occupied_table`	FK → Table	
	`occupancy_date`	Date	
	`left_time`	Time the party left (null while active)	
	`no_of_customers`	Head count	
	`merged_table`	Comma-separated merged table IDs	

12.2 Order Items

Fields:

	Field	Details	
	---	---	
	`order`	FK → Order	
	`table`	FK → Table	
	`item`	FK → Item	
	`qty`	Quantity	
	`price`	Price at time of order	
	`cover`	Course cover details	
	`modification`	Special modifications	
	`remarks`	Notes	
	`status`	See workflow below	
	`staff`	FK → Staff (waiter/server)	
	`datetime`	Order time (auto)	
	`courses_details`	Course details text	
	`bill_done`	Boolean — set to `True` when included in a bill	
	`inventory_deducted`	Boolean — tracks whether inventory ingredients were deducted	

12.3 Order Item Status Workflow

```
Requested → Pending → Preparing → Ready → Served
                                ↘ Cancelled
```

12.4 Order Item Status Log

Every status change is recorded in `OrderItemStatusLog`:
- `order_item` (FK), `old_status`, `new_status`, `changed_by` (FK → Staff), `changed_at` (auto timestamp).

12.5 Update Order Status

URL: `/restaurant/update_order_status/<id>`
Updates an order item's status. Triggers the status log creation.

12.6 Table Order

URL: `/restaurant/table_order/<id>`
Place or manage orders for a specific table.

12.7 Cancel Order

URL: `/restaurant/cancel_order/<id>`
Cancels an entire order.

---

13. Restaurant — Kitchen Dashboard

Page: `/restaurant/orders_kitchen/`
- Shows all active order items grouped by order.
- Kitchen staff can update item status: Pending → Preparing → Ready.
- Displays order time for tracking preparation duration.

---

14. Restaurant — Billing & Checkout

14.1 Checkout

URL: `/restaurant/checkout/<id>/<bill_no>`
Initiates the checkout/billing process for a table order.

14.2 Checkout Payment

URL: `/restaurant/checkout_payment/<id>`
Processes the payment for a table order.

14.3 Bill Generation

Restaurant Bill Fields:

	Field	Details	
	---	---	
	`invoice`	Auto-generated: `{sequence:04d}/{fiscal_year}`	
	`customer`	FK → Customer	
	`order`	FK → Order	
	`order_items`	TextField storing list of item IDs included in this bill	
	`fiscal_year`	From PrimaryData	
	`date` / `time`	Bill date/time	
	`name`, `company`, `pan`, `address`, `phone`	Billing details	
	`total_amount`	Sum of `(price × qty)` for included order items	
	`discount`	User-entered	
	`taxable_amount`	`total_amount − discount`	
	`tax`	`taxable_amount × (VAT% / 100)`	
	`service_charge`	Configurable (currently 0%)	
	`final_amount`	`taxable_amount + tax + service_charge`	
	`payment_method`	Cash, Card, etc.	
	`paid` / `due` / `paid_status`	Payment tracking (Clear/Partial/Credit)	

- On bill save, the included order items are marked `bill_done = True`.

14.4 Split Bill

URL: `/restaurant/split_bill/<id>`
Splits a single order into multiple bills (selecting which items go into which bill).

14.5 Bills List

Page: `/restaurant/bills/`
Lists all restaurant bills.

15. Restaurant — Staff & Shifts

15.1 Restaurant Staff

Pages: List, Add, Edit, Delete
Dedicated staff records for restaurant operations (waiters, chefs, etc.).

15.2 Staff Shifts

Pages: List, Add, Edit, Delete
Define shift schedules for restaurant staff.

---

16. Housekeeping — Task Assignment

Pages: List (`/housekeeping/tasks`), Create, Update, Delete
Fields:

	Field	Details	
	---	---	
	`room`	FK → Room (optional)	
	`area`	CharField — for non-room areas (lobby, corridor, etc.)	
	`date`	Task date	
	`shift`	Which shift (Morning, Evening, Night)	
	`cleaning_type`	Type of cleaning	
	`cleaning_task`	Detailed task description	
	`staff`	FK → Staff — assigned housekeeper	
	`supervisor`	FK → Staff — supervising staff	
	`task_status`	Assigned → On-Progress → Completed → Cancelled	
	`remarks`	Notes	

---

17. Housekeeping — Room Inspection

Pages: List (`/housekeeping/inspections`), Create, Update, Delete
Fields:

	Field	Details	
	---	---	
	`area`	Area inspected (if not a room)	
	`room`	FK → Room (optional)	
	`staff`	FK → Staff — inspector	
	`items_checked`	TextField — checklist of inspected items	
	`cleanliness_rating`	Integer (1–5 or 1–10 scale)	
	`datetime`	Inspection date/time	
	`remarks`	Notes	

---

18. Housekeeping — Lost & Found

Pages: List (`/housekeeping/lostnfounds`), Create, Update, Delete
Fields:

	Field	Details	
	---	---	
	`room`	FK → Room — where item was found	
	`datetime`	When found	
	`item`	Item description	
	`description`	Detailed description	
	`found_by`	Who found the item	
	`found_location`	Specific location	
	`stored_in`	Where the item is stored	
	`owner_identified`	Boolean — owner known?	
	`claimed_by`	Who claimed the item	
	`staff`	FK → Staff — staff who handed over	

---

19. Housekeeping — Maintenance Requests

Pages: List (`/housekeeping/maintenance`), Create, Update, Delete
Fields:

	Field	Details	
	---	---	
	`room`	FK → Room	
	`datetime`	When issue was reported	
	`issue`	Description of the problem	
	`reported_by`	FK → Staff	
	`assigned_to`	FK → Staff — maintenance person	
	`action_taken`	What was done	
	`completion_time`	How long it took	
	`verified_by`	FK → Staff — who verified the fix	

---

20. Housekeeping — Deep Cleaning

Pages: List (`/housekeeping/dc`), Create, Update, Delete
Fields:

	Field	Details	
	`room`	FK → Room	
	`scheduled_datetime`	When scheduled	
	`staff`	FK → Staff — assigned cleaner	
	`description`	Cleaning details	
	`completion_time`	When completed	
	`verified_by`	FK → Staff — who verified	

---

21. Housekeeping — MiniBar

21.1 MiniBar Items (Master List)

Pages: List (`/housekeeping/minibar_item/`), Create, Update, Delete
Fields: `type` (Alcoholic/Non-alcoholic/Juice), `item` (name), `description`, `unit`, `rate`.

21.2 MiniBar Setup (Per Room)

Pages: List (`/housekeeping/minibar`), Refill (`/housekeeping/refill_minibar/<rid>`), Update, Delete
Fields:

	Field	Details	
	`room`	FK → Room	
	`datetime`	Refill date/time	
	`item_refilled`	FK → MiniBarItem	
	`qty`	Quantity refilled	
	`charged`	Boolean — billable?	
	`staff`	FK → Staff	

Constraint: Unique combination of `room` + `item_refilled` (one entry per item per room).

21.3 MiniBar Consumption

Pages: List (`/housekeeping/minibar_consumptions`), Create (two forms: standard and by reservation), Update, Delete
Fields:

	Field	Details	
	`reservation`	FK → Reservation	
	`room`	FK → Room	
	`guest`	FK → Customer	
	`date`	Consumption date	
	`total`	Total charge	
	`staff`	FK → Staff	

21.4 Consumed Items (Detail per Consumption)

Fields: `minibar_consumption` (FK), `minibar` (FK → MiniBar), `qty_consumed`, `charge`, `datetime`, `remarks`.

Consumed Items List (`/housekeeping/consumed-items/`): View all consumed item records.

---

22. Housekeeping — Laundry

22.1 Laundry Services (Master List)

Pages: List (`/housekeeping/laundry_service/`), Create, Update, Delete
Fields: `service` (Wash/Iron/Dry-clean), `item` (garment type), `rate`.

22.2 Guest Laundry

Pages: List (`/housekeeping/laundry`), Create, Update, Delete
Fields:

	Field	Details	
	`reservation`	FK → Reservation	
	`room`	FK → Room	
	`guest`	FK → Customer	
	`date`	Service date (default: today)	
	`item`	FK → LaundryService	
	`description`	Notes	
	`qty`	Number of items	
	`price`	Price per item	
	`total`	`price × qty`	
	`remarks`	Additional notes	
	`staff`	FK → Staff	

---

23. Inventory Management

23.1 Inventory Categories (Top-Level)

Pages: List (`/inventory/categories/`), Add, Edit, Delete
Fields: `title` (unique), `caption`, `description`.
Examples: Room, Housekeeping Supplies, F&B, Equipment, Maintenance Items.

23.2 Sub-Categories

Pages: List (`/inventory/subcategories/`), Add, Edit, Delete
Fields: `category` (FK → InventoryCategory), `title` (unique), `caption`, `description`.

23.3 Units

Pages: List (`/inventory/units/`), Add, Edit, Delete
Fields: `title` — e.g., kg, litre, piece, pack, dozen.

23.4 Inventory Items

Pages: List (`/inventory/items/`), Add, Edit, Delete
Fields:

	Field	Details	
	`name`	Item name	
	`category`	FK → Category (Sub-Category)	
	`description`	Optional	
	`unit`	FK → Unit	
	`current_stock`	Decimal(10,2) — running balance	
	`par_level`	Optimal stock level	
	`reorder_level`	Minimum before restocking needed	

Restocking Alert: `needs_restocking()` returns `True` when `current_stock ≤ reorder_level`.

23.5 Stock Entries

Pages: List (`/inventory/stock_entries/`), Add, Edit, Delete
Fields: `inventory_item` (FK), `quantity`, `rate`, `date`, `supplier`, `notes`.

On Save: `current_stock += quantity` (auto-updates the inventory item).

23.6 Stock Usage (Auto-tracked)

Fields: `inventory_item` (FK), `quantity`, `date`, `used_by`, `department`.
On Save: `current_stock -= quantity`.

23.7 Item Ingredients (Recipe Link)

Model: `ItemIngradient`
Fields: `item` (FK → restaurant.Item), `ingradient` (FK → InventoryItem), `qty`, `unit` (FK → Unit).
Constraint: Unique `(item, ingradient)`.

This model links restaurant menu items to their raw ingredient requirements. When an order item is marked as prepared/served and `inventory_deducted = False`, the system can deduct the required ingredient quantities from inventory stock.

23.8 Inventory Logs

Fields: `inventory_item`, `action` (IN/OUT), `quantity`, `timestamp`, `notes`.
Full audit trail of all stock movements.

23.9 Suppliers

Fields: `name`, `contact_info`.

---

24. Services (Spa, Activities)

24.1 Service Categories

Pages: List (`/services/category/`), Add, Edit, Delete
Fields: `title`, `description`, `image`.

24.2 Services

Pages: List (`/services/service/`), Add, Edit, Delete
Fields: `category` (FK), `title`, `price`, `description`, `image`.

24.3 Service Bookings

Created via the in-house operations (Options Hub → Service) when a guest books a hotel service. Linked to a reservation.

Fields: `reservation` (FK), `customer` (FK), `service` (FK), `price`, `frequency`, `booked_date` (auto), `date`, `time`, `status`, `booked_by` (FK → Staff).

Status Values: Requested → Waiting → Approved → Accomplished → Cancelled.

---

25. Reports

25.1 Guest In-House Report

Page: `/reports/guest_in_house/?date=YYYY-MM-DD`
Shows:

	Data Point	Details	
	---	---	
	In-house guests	All reservations where `(check_in ≤ date OR date_from ≤ date) AND (check_out ≥ date OR date_to ≥ date)`, excluding cancelled	
	Today's arrivals	Count of `date_from = date`	
	Today's departures	Count of `date_to = date`	
	Next day arrivals	Count of `date_from = tomorrow`	
	Occupied rooms (today)	Count of checked-in reservations where `date_from ≤ date`	
	Occupied rooms (tomorrow)	Same for next day	
	Complementary rooms	Rooms with `price = 0`	

25.2 Flash Report

Page: `/reports/flash_report/?date=YYYY-MM-DD`
Daily operational snapshot with two sections:

Yesterday's Summary:
- House count (sum of adults)
- Rooms occupied
- Complementary rooms (price = 0)
- Charged rooms (price > 0)
- Occupancy percentage

Today's Summary:
- Expected occupancy (in-house count)
- Daily arrivals / departures
- House count
- OOO / OOS room counts
- Occupancy %: `(expected_occupancy / (total_rooms − OOO − OOS)) × 100`

Also lists all in-house reservations, expected arrivals, and expected departures.

25.3 Forecast Report

Page: `/reports/forecast_report/?date_from=&date_to=`

Date-range report showing day-by-day:

Column	Details
Allotment	Reservations with rooms assigned
Vacant	Available rooms − reserved rooms
Waitlisted	(placeholder, currently 0)
Total Rooms	All rooms
Actual Occupancy	Currently occupied count
Expected Arrival	`date_from = date`
Expected Occupancy	arrivals + carry-over occupancy
Expected Departure	`date_to = date`
Block OOO/MGM/GST	(placeholder)
Occupancy Forecast %	`(occupancy / total_rooms) × 100`

25.4 Occupancy Report

Page: `/reports/occupancy_report/?date_from=&date_to=`
Room-by-room occupancy grid. For each room × each date in the range, shows whether the room has a reservation (with guest details).

Permissions: All reports require `reports.view_reports` permission.

---

26. User & Staff Management

26.1 User Management

Pages: List (`/accounts/users/`), Add Form, Edit, Delete
Permission: Requires `auth.can_manage_users`.

User Fields: `username`, `password` (hashed), `email`, `mobile`, `first_name`, `last_name`, `is_active`, `is_staff`, `status`.

26.2 Staff Profiles

Page: `/accounts/profile/`
Shows the logged-in user's linked staff record and available shifts.

26.3 Change Password

Page: `/accounts/change_password/`
Fields: old_password, new_password, confirm_new_password.

26.4 Staff Model

Fields: `name`, `post`, `role`, `address`, `email`, `mobile`, `shift` (FK → Shift), `id_type`, `id_no`.

26.5 Shifts

Fields: `title`, `start_time`, `end_time`.

26.6 Linking Models

- StaffUser: Links Django `User` to `Staff` (unique constraint).
- CustomerUser: Links Django `User` to `Customer` (unique constraint).
- CustomerBill: Consolidates bills from all segments (hotel, restaurant, service, laundry) per customer.

26.7 Primary Data (System Config)

Model: `PrimaryData`
Key-value pairs for system-wide settings:

	Key	Example Value	Used For	
	---	---	---	
	`year`	`2082`	Fiscal year in invoices	
	`vat`	`13`	VAT percentage for billing	
	`service-charge`	`10`	Service charge percentage	
	`guest-add`	`500`	Price for adding an extra guest	

---

27. REST API

27.1 API v1 (DRF Router)

Mounted at `/api/v1/` using DRF's DefaultRouter with ViewSets.

27.2 API v1 (Manual Endpoints)

Mounted at `/api1/v1/`:

	Endpoint	Method	Details	
	---	---	---	
	`/api1/v1/login/`	POST	API authentication, returns token/session	
	`/api1/v1/restaurant/tables/`	GET	List all tables with status	
	`/api1/v1/restaurant/tables_category/`	GET	List table categories	
	`/api1/v1/restaurant/items/`	GET	List menu items with courses	

27.3 Planned API Endpoints (from notes)

	Resource	Methods	Filters	
	---	---	---	
	Category	GET	—	
	Classification	GET	filter by category	
	Foods & Courses	GET	filter by category & classification	
	Orders & Order Items	GET, POST, PUT, DELETE	filter by left_time	

Development Notes (from `note.txt`)

Completed ✅
- Ingredient calculation after preparing foods
- Users role middleware
- Table merge and take-away
- Room status change on different actions

In Progress 🔄
- Actions (Edit, Delete, Print) in View Requests, Bookings and Payments in Folio Page
- Reports finalization

- Housekeeping modules (final stage)

Remaining 📋
- Meal plan quantity add/edit
- Ingredient stock problem (update flow)
- Bill print (Hospitality)
- Synchronization for Minibar and Laundry service if cancelled from requests in hospitality

Overview

#	Module	Key Capabilities
1	**Hospitality**	Reservations, check-in/out, room status (12 states), folio, billing, night audit, foreign exchange
2	**Restaurant**	Full POS — menu management, classifications, table merge/take-away, kitchen dashboard, order status tracking, billing with VAT
3	**Housekeeping**	Task assignments, room inspections, minibar management & consumption, laundry, maintenance requests, lost & found, deep cleaning
4	**Inventory**	Stock entries/usage with auto-deduction, suppliers, ingredient linking to menu items, reorder alerts
5	**Services**	Bookable amenities (spa, activities) linked to guest reservations
6	**Reports**	Flash report, forecast, occupancy, guest in-house
7	**Accounts**	Customer database, user auth, consolidated billing, system config (VAT, fiscal year)
8	**Staff**	Staff profiles, shift management, role-based routing

Plus a REST API for restaurant operations (categories, menu items, orders) and role-based middleware for access control. See the full artifact for detailed breakdowns of each module.