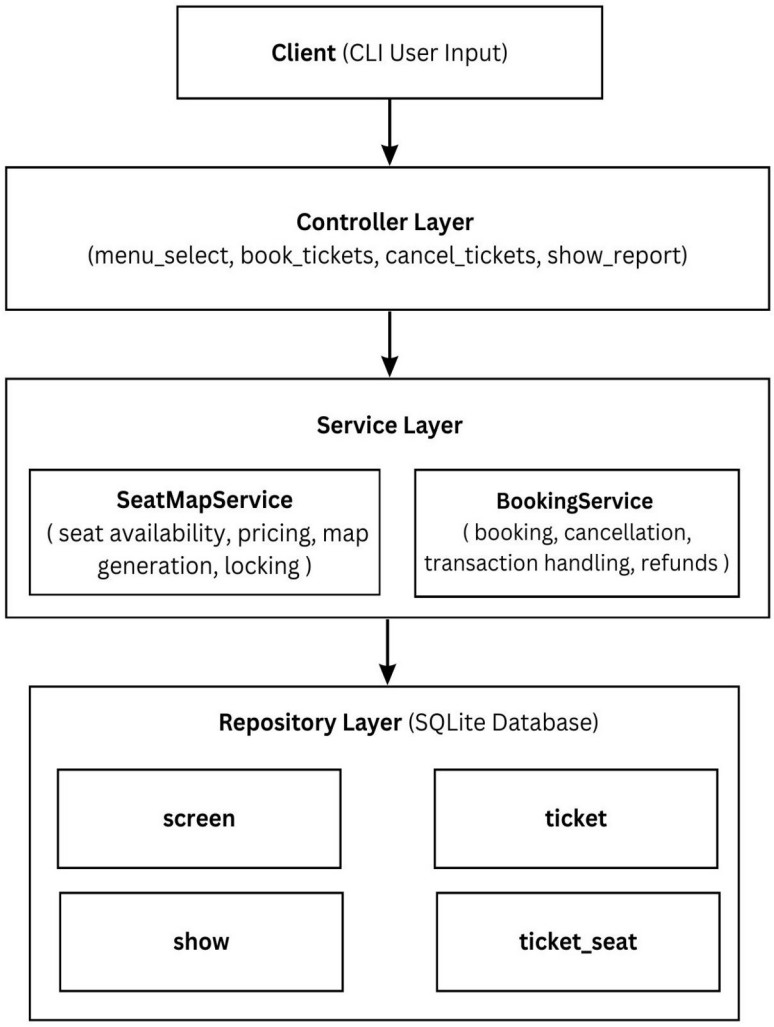
1) System Architecture

The system is a CLI-based ticket booking application built with **Python** and **SQLite**.

1.1 Flow



External Integrations (future-ready): Payment Gateway, User Authentication, Cloud Storage for reports.

2) Database Design (ERD & Schema)

The system is designed so that a single movie screen can host multiple shows. For each show, many tickets can be sold, and a single ticket can cover multiple seats. The system also directly links each show to every individual seat, which helps prevent double-booking.

2.1 Table Schema

Screen

Field	Type	Constraints	Notes
id	INTEGER	PK	Auto-increment
name	TEXT	NOT NULL	Movie/Screen name
rows	INTEGER	NOT NULL	Seat rows
cols	INTEGER	NOT NULL	Seat columns

Show

Field	Type	Constraints	Notes
id	INTEGER	PK	Auto-increment
screen_id	INTEGER	FK → screen.id	Movie-screen mapping
show_ts	TEXT	NOT NULL	Show date-time
base_price	REAL	NOT NULL	Base ticket price

Ticket

Field	Type	Constraints	Notes
id	INTEGER	PK	Auto-increment
show_id	INTEGER	FK → show.id	Show reference
user	TEXT	NOT NULL	Customer name
mobile	TEXT	10-digit	Customer mobile
status	TEXT	NOT NULL	BOOKED / CANCELLED
total	REAL	NOT NULL	Total price

TicketSeat

Field	Type	Constraints	Notes
id	INTEGER	PK	Auto-increment
ticket_id	INTEGER	FK → ticket.id	Ticket reference
show_id	INTEGER	FK → show.id	Show reference
row	INTEGER	NOT NULL	Seat row
col	INTEGER	NOT NULL	Seat column
price	REAL	NOT NULL	Seat price
UNIQUE(show_id,row, col)	—	Prevents double booking	

3) Functional Flow

3.1 Booking a Ticket

1. User chooses **Book Tickets**.
2. System lists available Movies.
3. User selects a movie → system shows Dates (**7-day window**).
4. User selects a date → system shows Show Times (**10 AM, 2 PM, 6 PM**).
5. System displays Seat Map (rows A–G, cols 1–7):
    - □ Available
    - ■ Booked
    - **X** User's selection
6. User selects seats (e.g., A1,B2).
7. User enters Name and Mobile Number.
8. System generates a **Booking Summary** with seat details and total price.
9. On confirmation, ticket is created and stored in the database.

3.2 Cancelling Tickets

1. User chooses **Cancel Tickets**.
2. Selects Movie → Date.
3. Enters Booking ID (e.g., B12).
4. System validates ticket details.
5. Shows booked seats.
6. User can cancel:
    - A single seat → refund for that seat.
    - Multiple seats → partial refund.
    - All seats → ticket marked CANCELLED.

3.3 Reports

1. User chooses **Show Report**.
2. Selects Movie → Date.
3. System shows each show with:
    - Occupancy (X/Y seats (Z%)).
    - Revenue collected.
    - Seat Type Breakdown:
        - Standard: 10 seats (₹1000)
        - Premium: 5 seats (₹750)

4) Controllers and Functions

BookingController

Function	Input	Output	Validation
get_movies	—	List of movies	DB not empty
get_dates	movie	List of dates	Movie exists
get_shows	movie, date	List of shows	Valid date
book	show_id, seats[], user, mobile	Ticket	Seat availability
cancel	ticket_id, seats[]	Refund	Ticket must be BOOKED
report	movie, date	Report	Valid movie/date

5) REST API Design (Future Web/Mobile Extension)

Method	URL	Purpose	Body	Response	Status
GET	/api/movies	List movies	—	[Movie]	200
GET	/api/movies/{name}/dates	Dates for movie	—	[Date]	200
GET	/api/movies/{name}/shows?date=	Shows	—	[Show]	200
POST	/api/tickets	Book ticket	{user, mobile, show_id, seats[]}	TicketDTO	201/400
PUT	/api/tickets/{id}/cancel	Cancel seats	{seats[]}	RefundDTO	200/404
GET	/api/reports/{movie}/{date}	Report	—	ReportDTO	200

6) UI Wireframes (CLI → Possible Web UI)

Movie List

Movies:
1. Coolie
2. Thug Life
3. Love Marriage
B. Back

Seat Selection

Seat Layout (□ Available, ■ Booked, X Yours)
1 2 3 4 5 6 7
A □ □ ■ □ □ □ □
B □ □ □ □ ■ □ □
...
Select seats: A1,B2

Booking Summary

Booking Confirmed!
Movie: Coolie
Date: 2025-09-04
Show: 6 PM
Seats:
A1 - Standard ₹100
B2 - Standard ₹100
Total: ₹200
Booking ID: B15

Report

Show Reports for Coolie on 2025-09-04

Show Time	Occupancy	Revenue	Seat Breakdown
10 AM	14/49 (28%)	₹1800	Standard: 10 (₹1000), Premium: 4 (₹800)
2 PM	5/49 (10%)	₹600	Standard: 3 (₹300), Premium: 2 (₹300)