

Book My Show Application:

App Idea: Booking Tickets for Movies

App Features:

- Customer can Buy Tickets for Shows
- Customer must login to buy tickets.
- Customer can select theatres based on location
- Customer can select date and check availability of seats
- Customer can use personal profile and can update the details
- Customer can check personal booking history
- Customer can search for his required movie
- Customer can filter his requirements (Date, Categories, Age, Price)
- Customer can use various payment portals (namesake)

App Tech:

- Angular (SASS and Bootstrap for Styling)
- .Net Core
- SQL Server
- PetaPoco(Micro-ORM) –
 - Provides faster data access.
 - MicroORM's provide raw speed compared to ORM's.
- Autofac(Dependency Injector) –
 - Clear documentation, feature rich and larger community.
- Code Repository – Git
- Deployment – Azure (Not Sure)
- NLB – Network Load Balancer (Not Necessary for now)

App Sketch:

- NavBar, Sub-NavBar, Recent releases
- Home Page (Best of services ...)
- Movies Page (Based on location(mandatory) ...)
- Search section (search for shows etc...)
- Location Picker Section (Pick or auto detect location)
- Booking Page (Contains about, booking proceeds, cast details etc ...)
- Booking Selection Page (various availability and provisions to book the shows)
- Theatre Seat Selection Page
- Payment page (confirm and block the booking)
- Confirmation Page (Display confirm message and ticket)

App Specifications:

- Customer can either be logged in or not
- Customer searches for his movie choice
- Customer selects the movie
- Customer selects his preferred date
- Customer selects the theatre
- Customer selects the screen and show
- Customer is requested to login if he/she's not
- Customer is directed to seat selection (selected seats are blocked)
- Customer is redirected to Payment Gateway and makes the payment
- Customer Bookings are confirmed and redirect to confirmation page
- Customer ticket is now displayed in Booking History
- Customer will now be notified ahead of time for the show

App System Design:

There will be two main components in the system

1. Book My Show App
2. Theatre Web-App (Serves to provide theatre list and availability)

BMS app can connect to Theatre API from any device

Theatre API needs its own server and database to provide its details (could try web scraping BMS or any external API)

The App contacts the Theatre API to obtain the details and sync based on the customer selections and bookings.

We'll request the available seats from Theatres API and book using the API's.

We will block the tickets based on the IO Request and blocking.

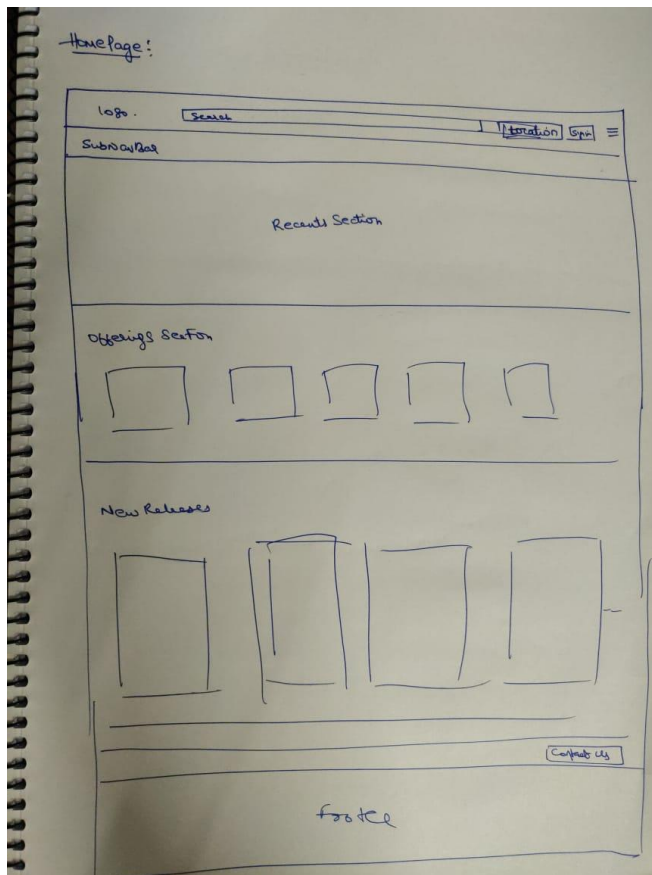
- We can follow OS Scheduling Algorithms for the blocking and unblocking requests

Theatre API Requirements:

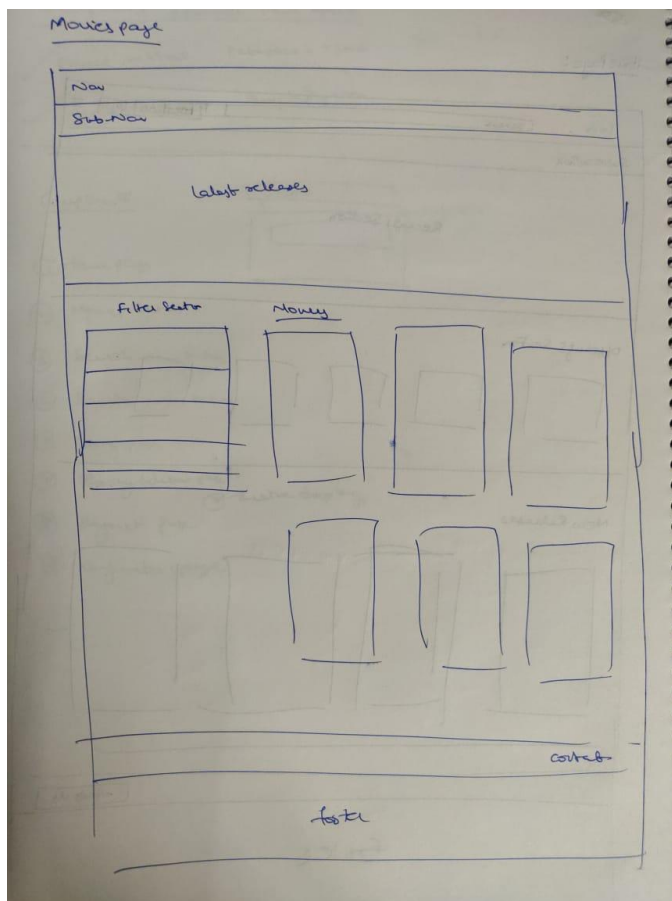
- List of cities
- Get Available Seats
- Get Show Timings
- Block Seats
- Unblock Seats
- Book Seats
- Etc.

App WireFraming:

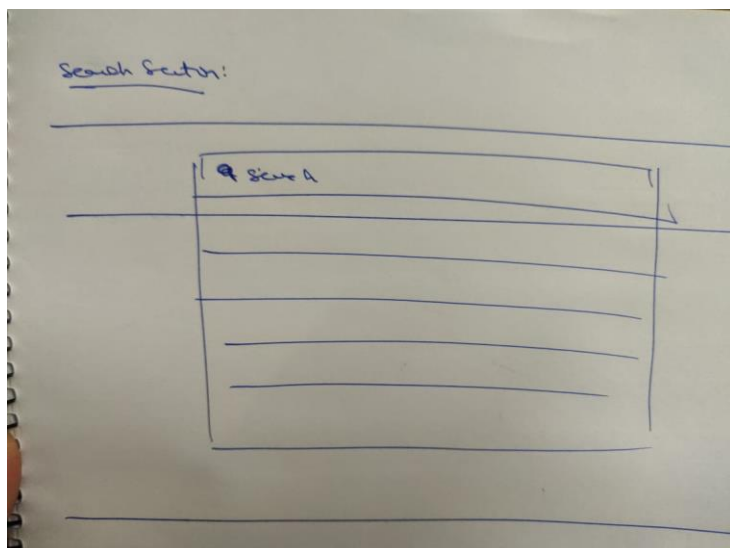
- Home Page:



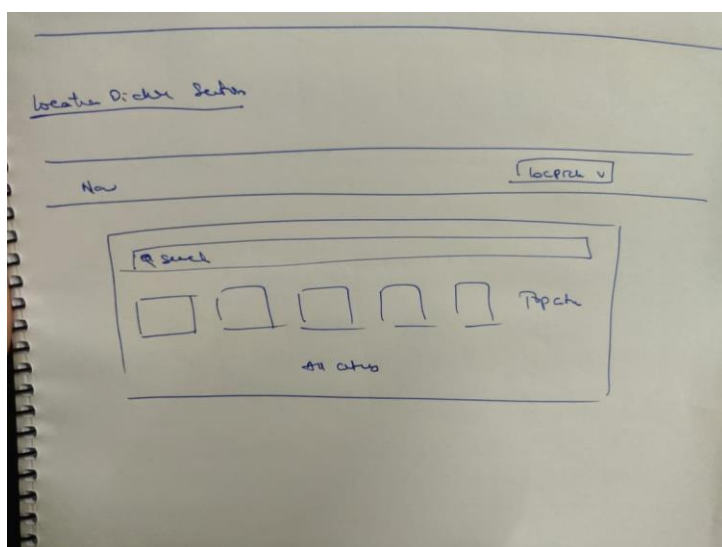
- Movies Page:



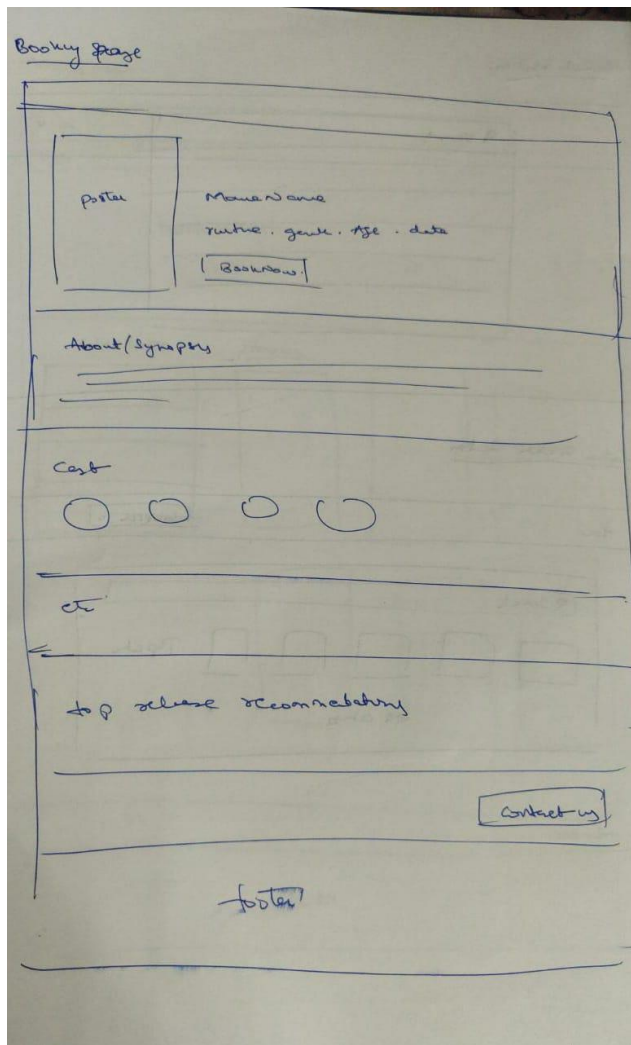
- Search section:



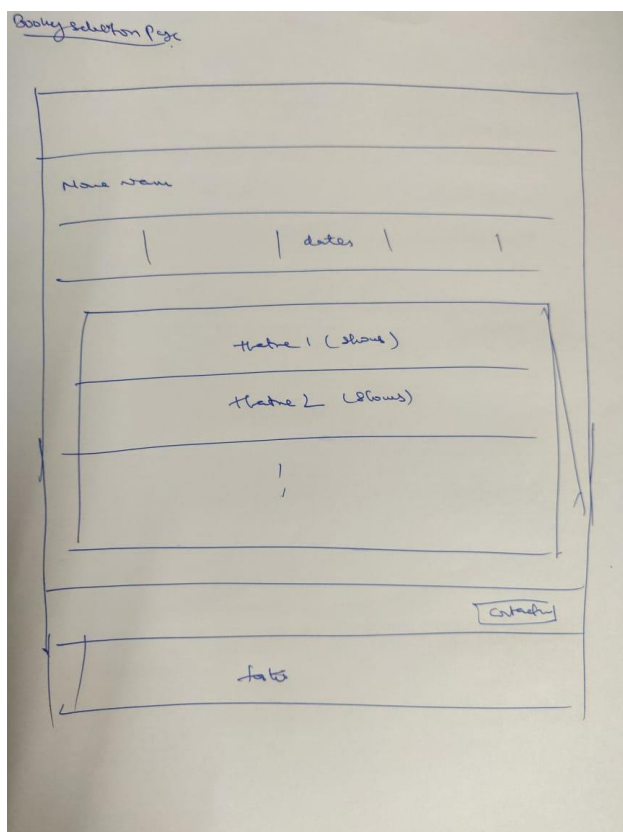
- Location Picker Section:



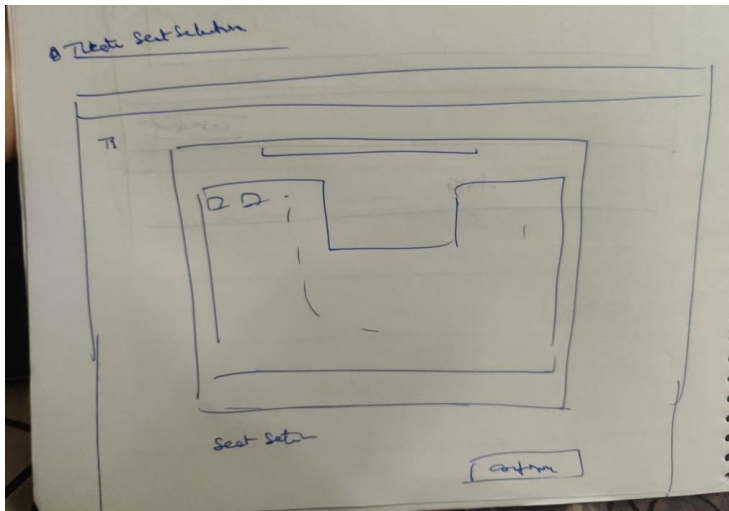
- Booking Page:



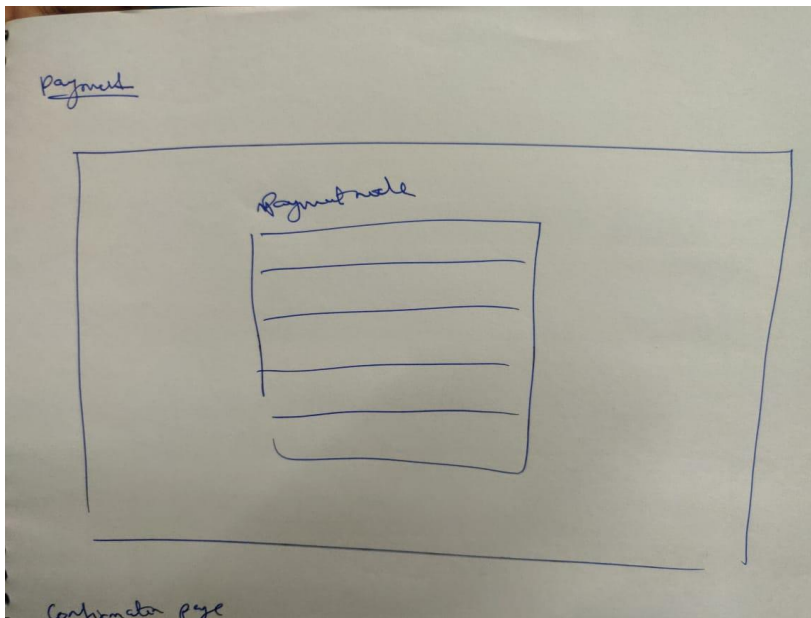
- Booking Selection Page:



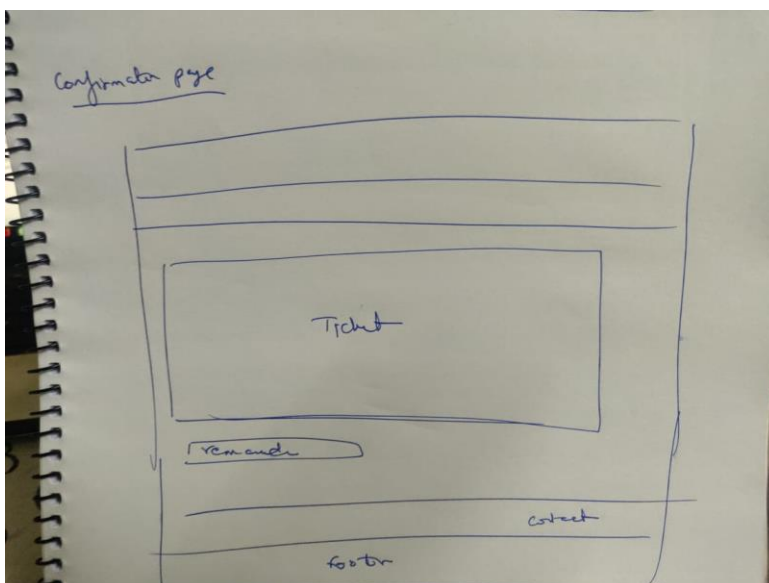
- Theatre Seat Selection Page:



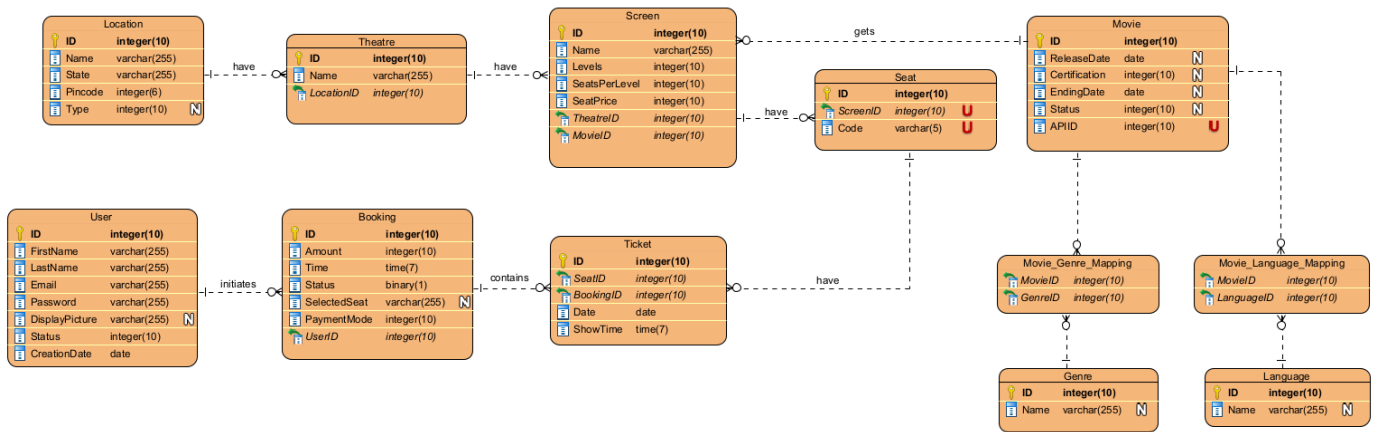
- Payment page:



- Confirmation Page:



Database Structure:



Predicate:

- Location is independent
 - Location is identified by ID
 - Location have 0-to-n theatres
- Movie is independent
 - Movie is identified by ID
 - Movie have genres and allows languages
 - Each movie gets 0-to-n screens
- User is independent
 - User is identified by ID
 - User can initiate 0-to-n number of bookings
- Theatre is dependent on, Location
 - Theatre is identified by ID
 - Each Theatre will have 0-to-n Screens
- Screen is dependent on, Theatre and Movie
 - Screen is identified by ID
 - Each screen gets 1 movie
 - Each screen have 0-to-n seats
- Seat is dependent on, Screen
 - Seat is identified by ID and ScreenID, SeatCode
 - Seat have 0-to-n tickets for x number of days
- Ticket is dependent on, Seat
 - Ticket is identified by ID and SeatID, BookingID
 - Each Ticket is assigned to 1 Seat
- Booking is dependent on, User
 - Booking is identified by ID
 - Each booking has 0-to-5 tickets
- Movie_Genre_Mapping and Movie_Language_Mapping maps movies to genres and languages respectively

Theatre WebApi

BMS FrontEnd

BMS BackEnd

[Testing, Beta, Production]

Deploy

[Define, Design, Develop, Test, Deploy] - Repeat Cycle