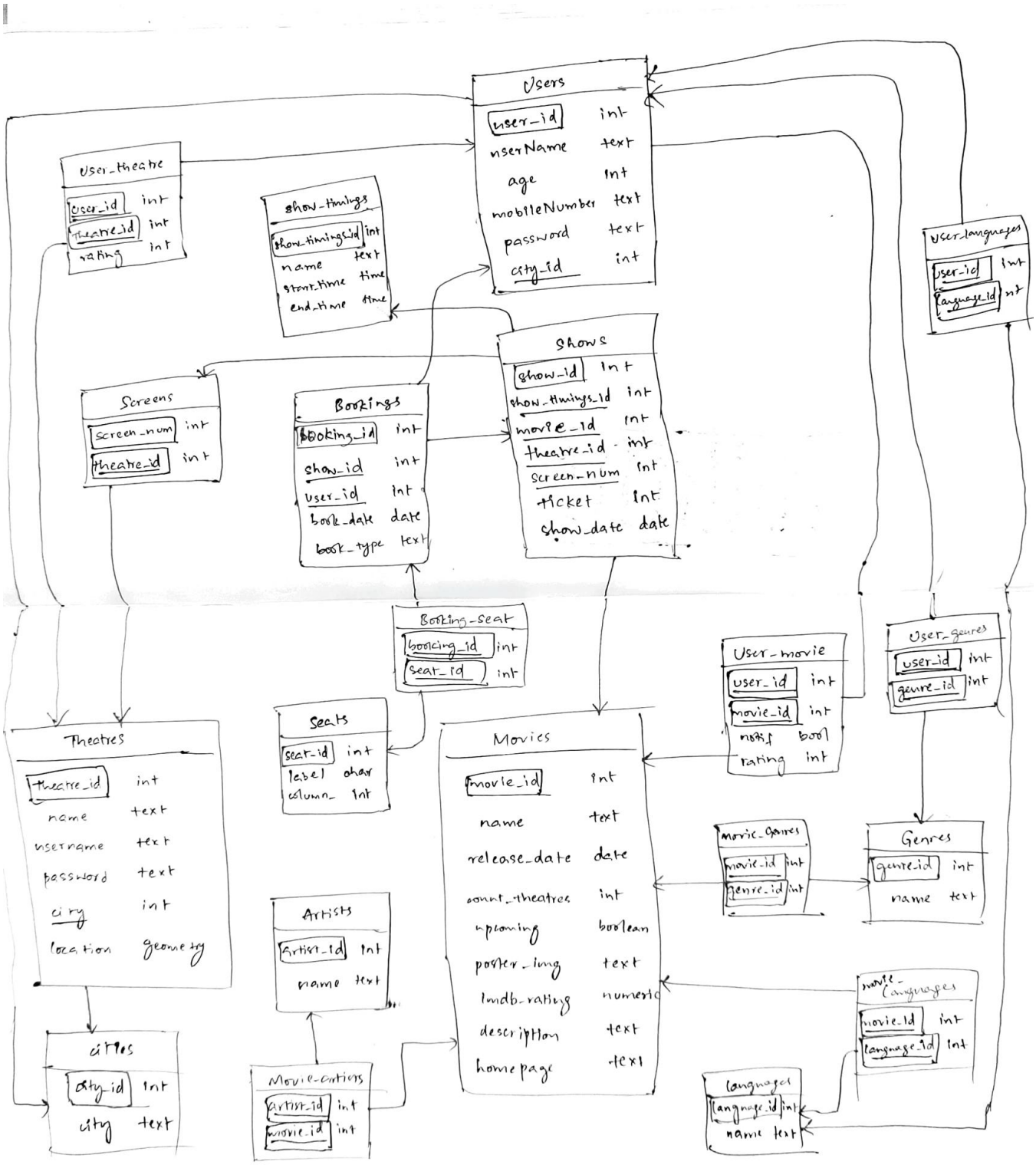
190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

Project Design Document

a. Logical Schema:



190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

a. Integrity Constraints:

1. Movies:
    a. Movie_id - primary key
    b. name not NULL
    c. count_theatres>=0: Number of theatres in which the movie is being aired can only be greater than or equal to 0
    d. imdb_rating>=0 and imdb_rating<=10: Imdb rating of movie must lie between 0 and 10
    e. Release_date can be NULL - NULL value of release_date means we do not have its value available
    f. Poster_img can be NULL - NULL means Poster image of the movie is not available
    g. Imdb_rating can be NULL - NULL means IMDB rating of movie isn't available
2. Cities:
    a. city_id - primary key
    b. city not NULL
3. Theatres:
    a. Theatre_id - primary key
    b. name not NULL
    c. username unique not NULL
    d. Password not NULL
    e. City can be NULL - NULL means city info isn't available
    f. City - Foreign key references cities
4. Users:
    a. User_id - primary key
    b. userName not NULL
    c. age>0
    d. mobileNumber unique not NULL
    e. mobileNumber not like '%[^0-9]%' and LENGTH(mobileNumber)=10 - mobile number must be of length 10 and must contain digits only
    f. Password not NULL
    g. City_id can be NULL - NULL means city info of user isn't available
5. Show_timings:
    a. Show_timings_id - primary key
    b. name in('Morning','Afternoon','First Show','Second Show') - name can be one of these only
    c. end_time>start_time
6. Artists:
    a. Artist_id - primary key
    b. Name not NULL
7. Movie_Artists:
    a. Artist_id, Movie_id - primary key
    b. Artist_id foreign key references artist
    c. Movie_id foreign key references movie

190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

8. Genres:
    a. Genre_id - primary key
    b. Name not NULL
9. Languages:
    a. Language_id -primary key
    b. Name not NULL
10. Movie_genres:
    a. Movie_id, genre_id - primary key
    b. Movie_id foreign key references movie
    c. Genre_id foreign key references genre
11. Movie_languages:
    a. Movie_id, language_id - primary key
    b. Movie_id foreign key references movie
    c. Language_id foreign key references language
12. Screens:
    a. Screen_num, theatre_id - primary key
    b. Screen_num not NULL
    c. Theatre_id foreign key references theatre
13. Seats:
    a. Seat_id - primary key
    b. label like '%[A-M]%' and LENGTH(label)=1 - label must be a letter between A and M
    c. column_ in (1,2,3,4,5,6,7,8,9,10)
    d. UNIQUE (label,column_) - {label,column_} is a candidate key
14. Shows:
    a. Show_id - primary key
    b. ticket>0
    c. Show_date not NULL
    d. Show_timings_id foreign key references show_timings
    e. Movie_id foreign key references movie
    f. screen_num, theatre_id foreign key references screen
    g. UNIQUE (show_timings_id,movie_id,screen_num,theatre_id,show_date)
15. Bookings:
    a. (booking_id, seat_id) - primary key
    b. user_id foreign key references users
    c. show_id foreign key references shows
    d. set (booking_id, show_id, user_id) has to be unique
    e. book_type can either be 'online' or 'offline'
16. Booking_seat:
    a. Primary key - booking_id, seat_id
    b. booking_id foreign key references bookings
    c. show_id foreign key references seats
17. User_languages:
    a. primary key - user_id,language_id
    b. language_id foreign key references languages

c. user_id foreign key references users

18. User_genres:
    a. primary key - (user_id,genre_id)
    b. genre_id foreign key references genres
    c. user_id foreign key references users
19. User_movie:
    a. (rating>=1 and rating<=5) or rating is null(when movie is upcoming)
    b. primary key - (user_id,movie_id)
    c. notif - not null when upcoming movie and subscribed for notification by user - either 0 or 1
    d. user_id foreign key references users
    e. movie_id foreign key references movies

20. User_theatre:
    a. rating>=1 and rating<=5
    b. user_id,theatre_id - primary key
    c. user_id foreign key references users
    d. theatre_id foreign key references theatres

c. Views

```
CREATE view all_bookings AS
SELECT show_id, user_id, bookings.booking_id, book_date, book_type, COUNT(*)
seats_booked
FROM bookings, booking_seat
WHERE bookings.booking_id = booking_seat.booking_id
GROUP BY show_id, user_id, bookings.booking_id, book_date, book_type
```

Created a view for “Number of seats for each booking id” - this is used because it is used multiple times in “viewing Analytics” use cases

d and f. Transactions

SET 1

1 Sign up for application
transaction -
checks if entry of user with entered phone number is present in users
if(success){
  insert entry into users
  Get id of user created
  Get id of language_entry
  Get id of genre entry
  Get id of city entry

190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

insert entry into user_language if language entry not empty
insert entry into user_genre if genre entry not empty
}

queries -
SELECT COUNT(*) FROM users WHERE mobileNumber= mobile_entry
INSERT INTO users (userName, age, mobileNumber, password, city_id)
VALUES(username_entry, age_entry, mobile_entry, pswd_entry, city_id_entry)
INSERT INTO user_languages (user_id, language_id) VALUES(user_id, language_id)
INSERT INTO user_genres (user_id, genre_id) VALUES(user_id, genre_id)

4 Edit Profile
Get city_id_entry from city_entry
Get genre_id_entry from genre_entry
Get language_id_entry from language_entry
Update values in users, user_language, user_genre

Queries-
UPDATE users SET city_id = city_id_entry WHERE user_id = user_id_entry
UPDATE users SET age = age_entry WHERE user_id = user_id_entry
UPDATE users SET userName = userName_entry WHERE user_id = user_id_entry
UPDATE user_languages SET language_id =language_id_entry WHERE user_id = user_id_entry
UPDATE user_genres SET genre_id = genre_id_entry WHERE user_id = user_id_entry

10 Booking available seats for a show
Insert entries into bookings, seat_booking(Multiple inserts)

Queries -
INSERT INTO bookings (show_id, user_id, book_date, book_type) VALUES(show_id_entry, user_id_emtry, book_Date_entry, ‘online’)
INSERT INTO booking_seat VALUES(booking_id, seat_id_entry)

SET2

4. Register a Theatre
INSERT INTO theatres(name, city, location) values(name_entry, city_entry, location_entry)
INSERT INTO screens values (screen_num_entry, theatre_id)

5 Add a movie
Get movie_id_entry from movie_entry
Get genre id entries from genre_entries

190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

Get language id entries from language_entries
Insert entries into movie, movie_genre, movie_language, cast_

Queries -
INSERT INTO movies (name, release_date, count_theatres, upcoming, poster_img, imdb_rating, description, homepage) VALUES(movie_name_entry, release_date_entry, 0, 1, img_entry, NULL, description_entry, homepage_entry)

13 Change availability of seats
Insert entries into bookings, seat_booking

Queries-
INSERT INTO bookings (show_id, user_id, book_date, book_type) VALUES(show_id_entry, user_id_emtry, book_Date_entry, ‘offline’)
INSERT INTO booking_seat VALUES(booking_id, seat_id_entry)

14 Add movie shows
Insert multiple entries of movie show timings in shows table

Queries-
SELECT * from shows where show_date BETWEEN today_date AND today_date+7days AND theatre_id = theatre_id_entry AND movie_id = movie_id_entry AND screen_num = screen_num_entry
INSERT INTO shows (show_timings_id, movie_id, theatre_id, screen_num, ticket, show_date) values(show_timings_id_entry, movie_id_entry, theatre_id_entry, screen_num_entry, ticket_entry, show_date_entry)

15 Extend an already airing movie of a screen to 7 more days
Insert multiple entries of movie show timings in show table

Queries -
SELECT * from shows where show_date BETWEEN today_date AND today_date+7days AND theatre_id = theatre_id_entry AND movie_id = movie_id_entry AND screen_num = screen_num_entry
INSERT INTO shows (show_timings_id, movie_id, theatre_id, screen_num, ticket, show_date) values(show_timings_id_entry, movie_id_entry, theatre_id_entry, screen_num_entry, ticket_entry, show_date_entry)

190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

Non - Transaction Use cases:

SET 1

2. Select city
NO sql query

3. View Profile (get user_id from logged in user)
SELECT userName, age, mobileNumber, city_id from users where user_id = user_id_entry
SELECT name from genres where genre_id in (SELECT genre_id from user_genre where user_id = user_id_entry)
SELECT name from languages where language_id in (SELECT language _id from user_language where user_id = user_id_entry)
SELECT city from cities where city_id = city_id_entry

5. View movies being aired in a city

SELECT name from movies where movie_id in (SELECT distinct movie_id from shows where theatre_id in (SELECT theatre_id from theatres where city = city_id_entry))

6. Click a movie and view its info
SELECT * from movies where movie_id = movie_id_entry
SELECT distinct name from artist WHERE artist_id in (SELECT artist_id from movie_artists where movie_id = movie_id_entry)

7. View artist profile in the artist page
SELECT * FROM movie_artists WHERE artist_id = artist_id_entry

8. View movies being aired in a theater
SELECT * from shows where theatre_id in (SELECT theatre_id from theatre where city_id = city_id_entry) AND movie_id = movie_id_entry

9. View available seats for a show
SELECT seat_id FROM booking_seat WHERE booking_id IN
(SELECT booking_id FROM bookings WHERE show_id IN
(SELECT show_id FROM shows WHERE movie_id = movie_id_entry AND theatre_id = theatre_id_entry AND screen_num = screen_num_entry AND show_date = show_date_entry AND show_timings_id = show_timings_id_entry))

11 Rate a movie
Get movie_id_entry for movie_entry
Insert rating entry in user_movie table
(notif and rating cannot be non null values simultaneously. When upcoming movie is subscribed by user, entry is created in user_movie with notif as 1 and rating as NULL. When upcoming

movie is released in theatres, entrie for the user_id is removed from user_movie. When movi is atched by user, rating can be gien only once and can be given only when movie is watched by user. Entry is created with notif as null and rating as rating entry by user)

Queries -
INSERT INTO user_movie VALUES(user_id_entry, movie_id_entry, NULL, rating_entry)

12. View upcoming movies
SELECT name from movies where upcoming = TRUE

13 Info of Upcoming movie (get upcoming_movie_id_entry)
SELECT * FROM movies WHERE movie_id = upcoming_movie_id_entry

14 Set a notification preference for movie tickets release update
INSERT INTO user_movie VALUES(user_id_entry, movie_id_entry, 1, NULL)

15. Showing Recommendations

SELECT name from movies where movie_id in (SELECT movie_id from movie_genres NATURAL FULL OUTER JOIN movie_languages where genre_id in (SELECT genre_id from user_genres where user_id = user_id_entry) and language_id in (SELECT lanuguage_id from user_languages where user_id = user_id_entry)) SORT BY imdb_rating DESC LIMIT 5

16 View available theatres
SELECT city_id FROM users WHERE user_id = user_id_entry
SELECT name FROM theatres WHERE city = city_id_entry

17 View shows available in a theater
SELECT * FROM shows WHERE theatre_id = theatre_id_entry

18 Rate a theatre
Get theatre_id_entry for theatre_entry
Insert rating entry in user_theatre table

Queries -
INSERT INTO user_theatre VALUES(user_id_entry, theatre_id_entry, rating_entry)

19 Filter out movies based on genre, language (get genre_id_entry and language_id_entry)
SELECT name FROM movies WHERE movie_id IN (SELECT movie_id FROM movie_genres WHERE genre_id = genre_id_entry)
SELECT name FROM movies WHERE movie_id IN (SELECT movie_id FROM movie_languages WHERE language_id = language_id_entry)

190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

20 View theatres available within a selected range of distance from a selected location
SELECT name FROM theatres WHERE ST_Distance(location, selected_location) < threshold
(all theatres within given threshold would be shown to user)

21 View history of bookings

SELECT name, booking_id, book_date, book_type, label, column_ FROM movies, shows, booking_seat, seats
(SELECT * FROM bookings WHERE user_id = user_id_entry) user_bookings
WHERE user_bookings.booking_id = booking_seat.booking_id AND user_bookings.show_id = shows.show_id AND shows.movie_id = movies.movie_id AND booking_seat.seat_id = seats.seat_id

SET 2:

1.Logging on to the system
SELECT count(*) from users WHERE mobileNumber = mobile_number_entry AND password = password_entry

2.Reset Password
UPDATE users SET password = password_entry WHERE user_id = user_id_entry

3. Logging out of System
No sql queries

6 Add an artist
INSERT INTO artists (name) VALUES(artist_name_entry)

7 Admin Analytics

SELECT theatre_id, theatres.name, show_online_offline.book_type,
SUM(show_online_offline.seats_booked) sum_seats
FROM theatres, shows,
(SELECT book_type, show_id, SUM(seats_booked) seats_booked_ FROM all_bookings
GROUP BY book_type, show_id) show_online_offline
WHERE theatres.theatre_id = shows.theatre_id AND shows.show_id = show_online_offline.show_id
GROUP BY theatre_id, theatres.name, show_online_offline.book_type

190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

8 View movies being aired in a theater
SELECT movie_id, name from movie WHERE movie_id in (SELECT movie_id FROM shows WHERE theatre_id = theatre_id_entry)

9 Theatre Analytics - Live vs Online Bookings
SELECT count(seat_id) from booking_seat where booking_id in (SELECT booking_id FROM bookings WHERE book_type = ‘online’ and show_id in (SELECT show_id from shows WHERE theatre_id = theatre_id_entry))
SELECT count(seat_id) from booking_seat where booking_id in (SELECT booking_id FROM bookings WHERE book_type = ‘offline’ and show_id in (SELECT show_id from shows WHERE theatre_id = theatre_id_entry))

10 Theatre Analytics - Percentage of Audience (get movie_id_entry from movie_entry by user)
SELECT seats_for_show.theatre_id, theatres.name, SUM(seats_for_show.seats_booked) audience FROM theatres, shows,
(SELECT show_id, SUM(seats_booked) seats_booked_ FROM all_bookings GROUP BY show_id) seats_for_show
WHERE theatres.theatre_id = seats_for_show.theatre_id AND shows.movie_id = movie_id_entry AND shows.show_id = seats_for_show.show_id
GROUP BY seats_for_show.theatre_id, theatres.name

11 Theatre Analytics - Choices of Audience
Genre -
SELECT genres.genre_id, genres.name, SUM(seats_for_show.seats_booked)
FROM movies, movie_genres, shows, genres,
(SELECT show_id, SUM(seats_booked) seats_booked_ FROM all_bookings GROUP BY show_id) seats_for_show
WHERE movies.movie_id = movie_genres.movie_id AND shows.movie_id = movies.movie_id AND shows.show_id = seats_for_show.show_id AND genres.genre_id = movie_genres.genre_id
GROUP BY genres.genre_id, genres.name

Language -
SELECT languages.language_id, languages.name, SUM(seats_for_show.seats_booked)
FROM movies, movie_languages, shows, languages,
(SELECT show_id, SUM(seats_booked) seats_booked_ FROM all_bookings GROUP BY show_id) seats_for_show
WHERE movies.movie_id = movie_languages.movie_id AND shows.movie_id = movies.movie_id AND shows.show_id = seats_for_show.show_id AND languages.language_id = movie_languages.language_id
GROUP BY languages.language_id, languages.name

190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna

12 Theatre Analytics - View Ratings
Movies aired-
SELECT movies.movie_id, movies.name, AVG(rating)
FROM user_movie, movies
GROUP BY movies.movie_id, movies.name

Theatres -
SELECT theatres.theatre_id, theatres.name, AVG(rating)
FROM user_theatre, theatres
GROUP BY theatres.theatre_id, theatres.name

e. DDL.sql and InsertData.sql

DDL.sql , InsertData.sql

g. Indices

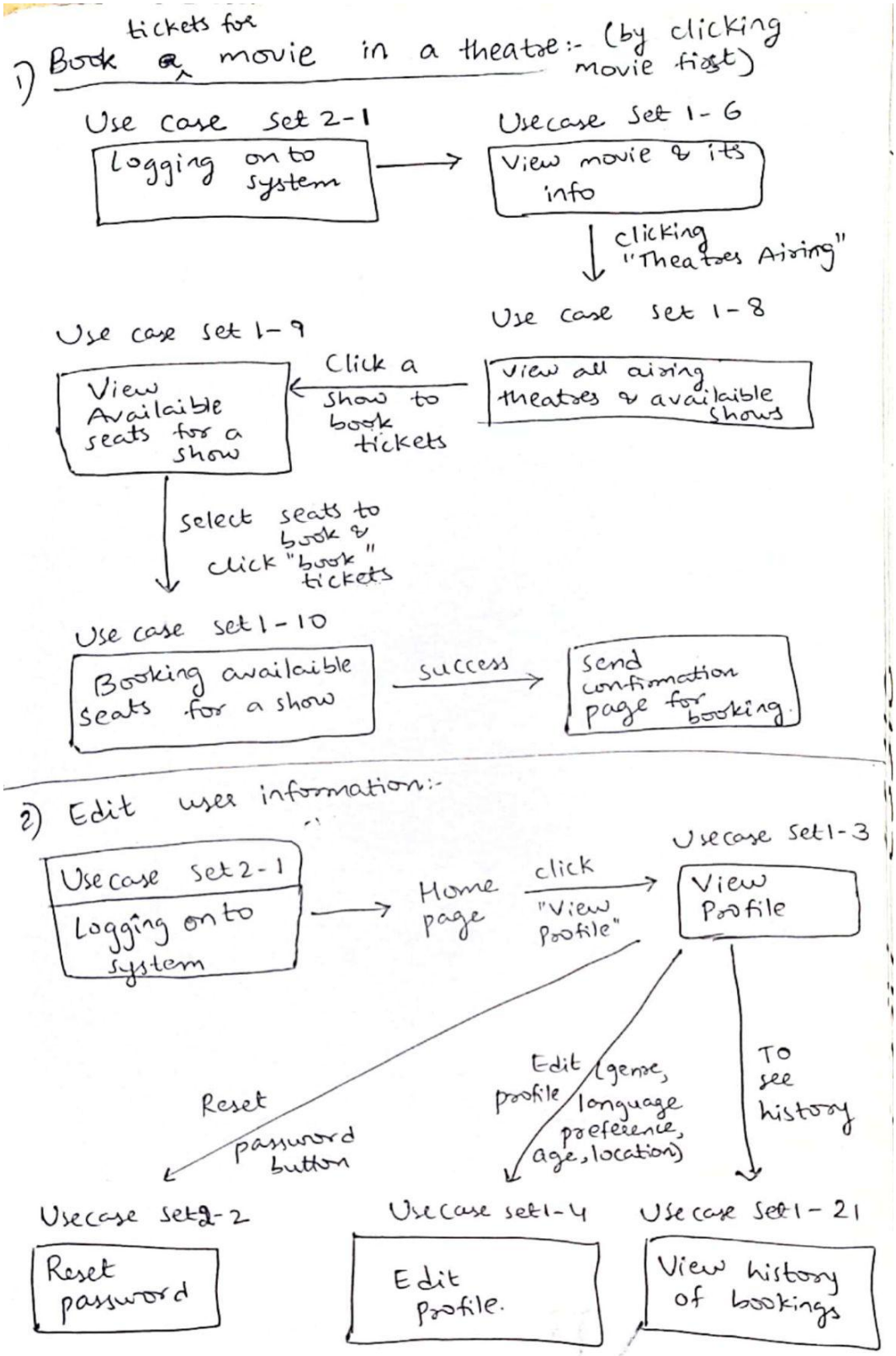
Index on show_date in shows table helps in increase in the performance
Most of the other queries use ids of the entities as predicates, which are already indices, so no additional indices needed.

h. Forms

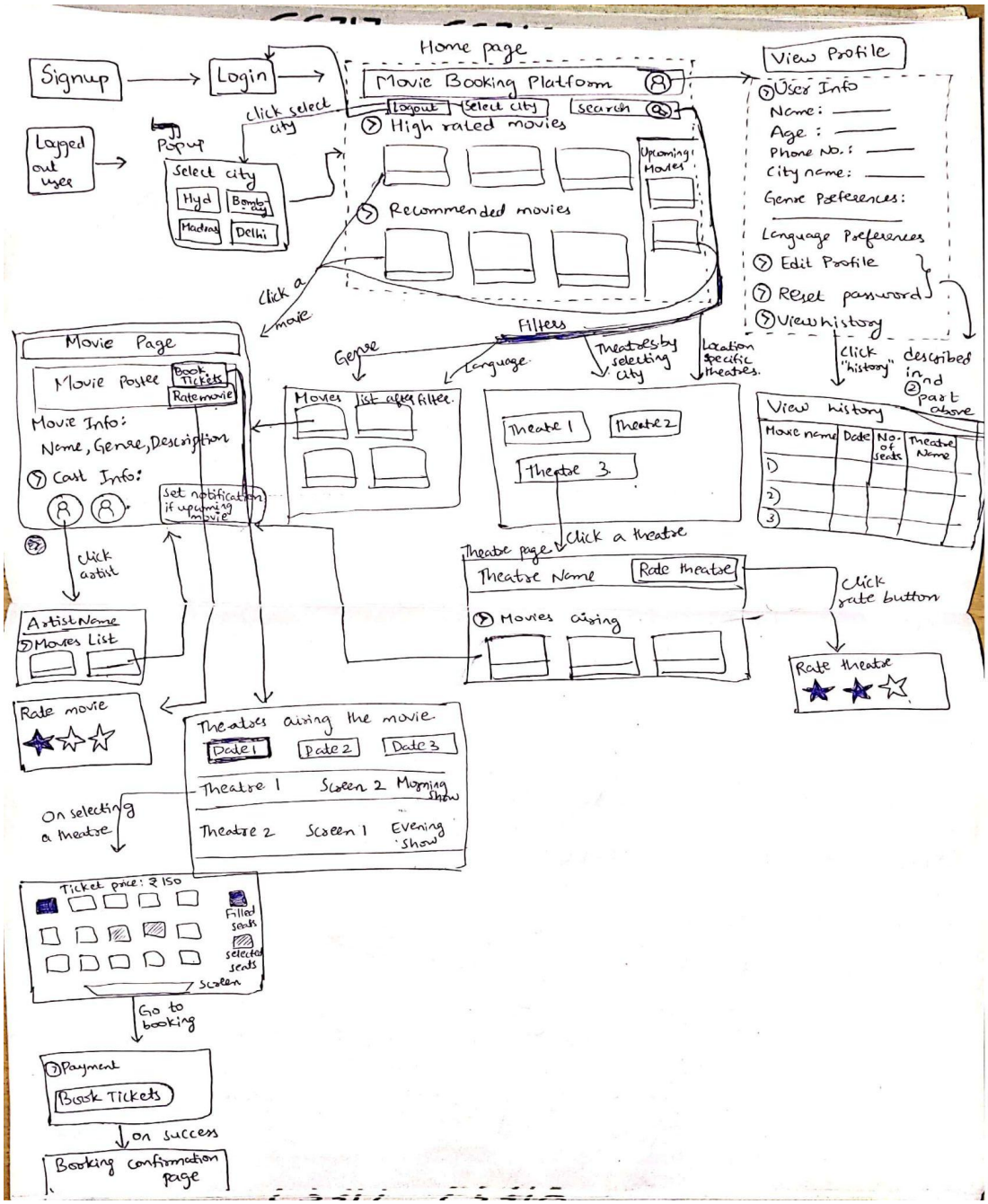
Link: https://drive.google.com/file/d/1S13lysIETUOYacBkeiNZ2xVBGgfqJbWj/view?usp=sharing

i. Business Logic Controller

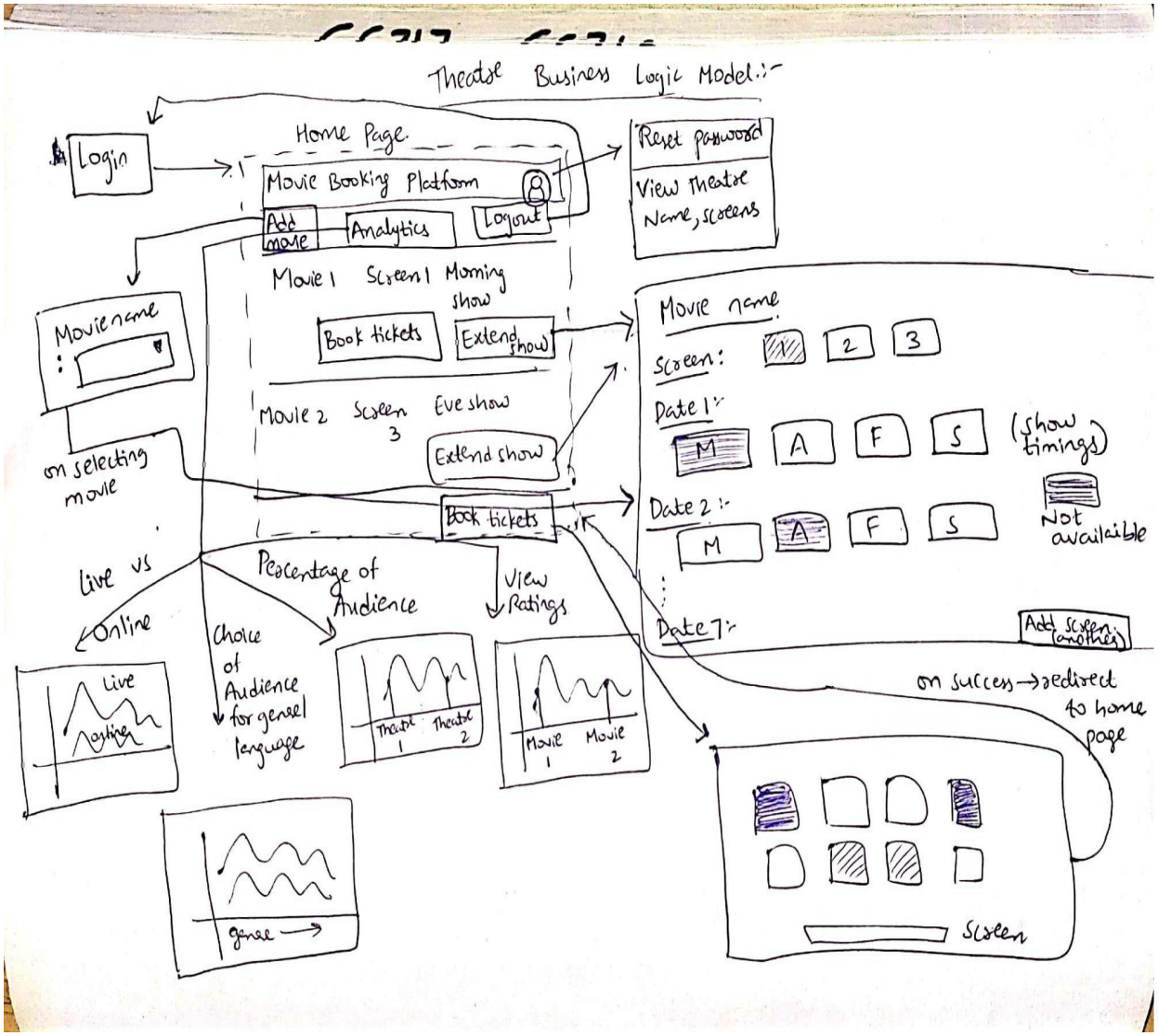
190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna



190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna



190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna



190050050 - Ilindra Sai Lakshmi Shreya, 190050061 - Konda Renu
, 190050078 - Palti Ramyasri, 190050082 - Penta Rahul Krishna