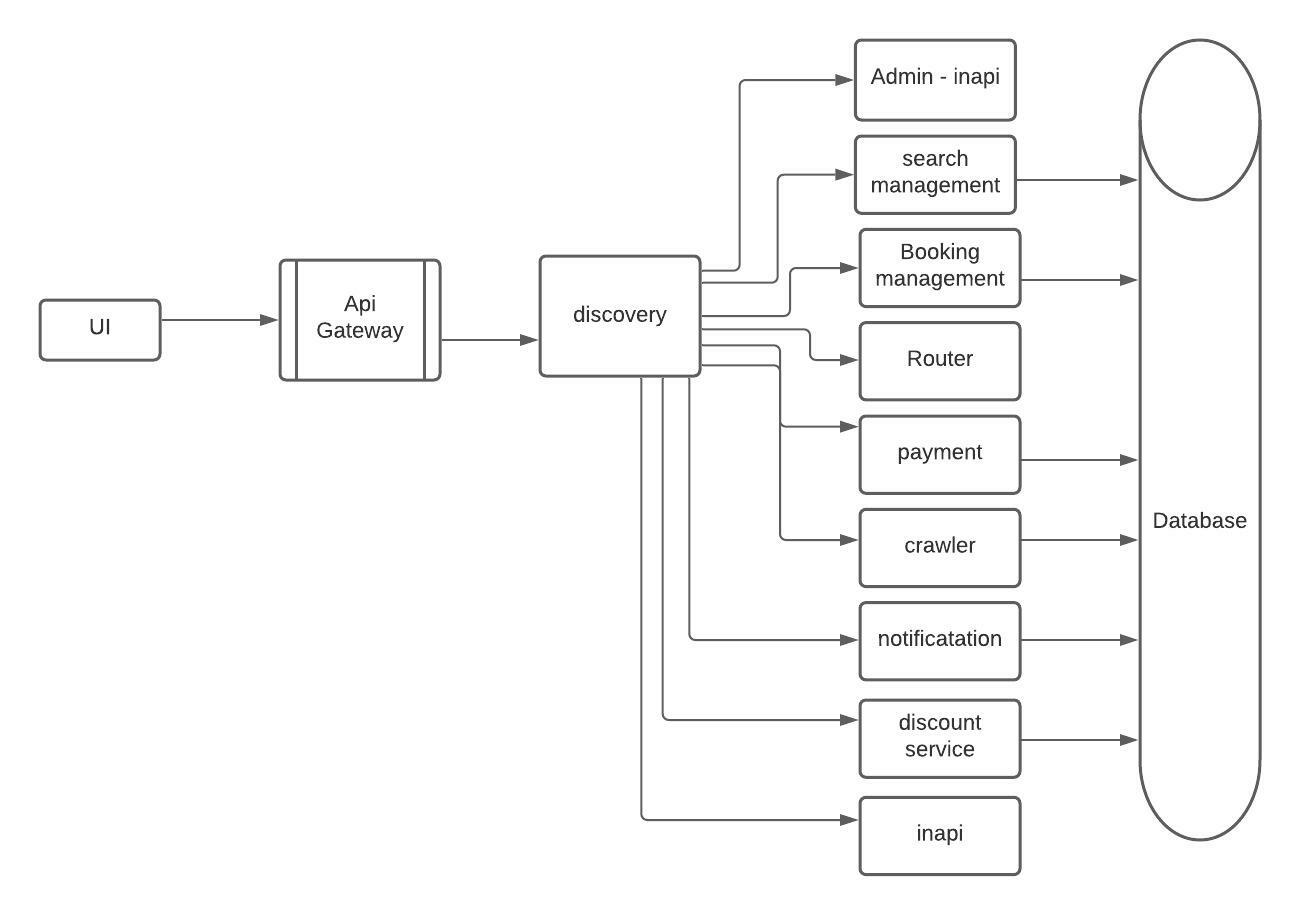
Fun Time

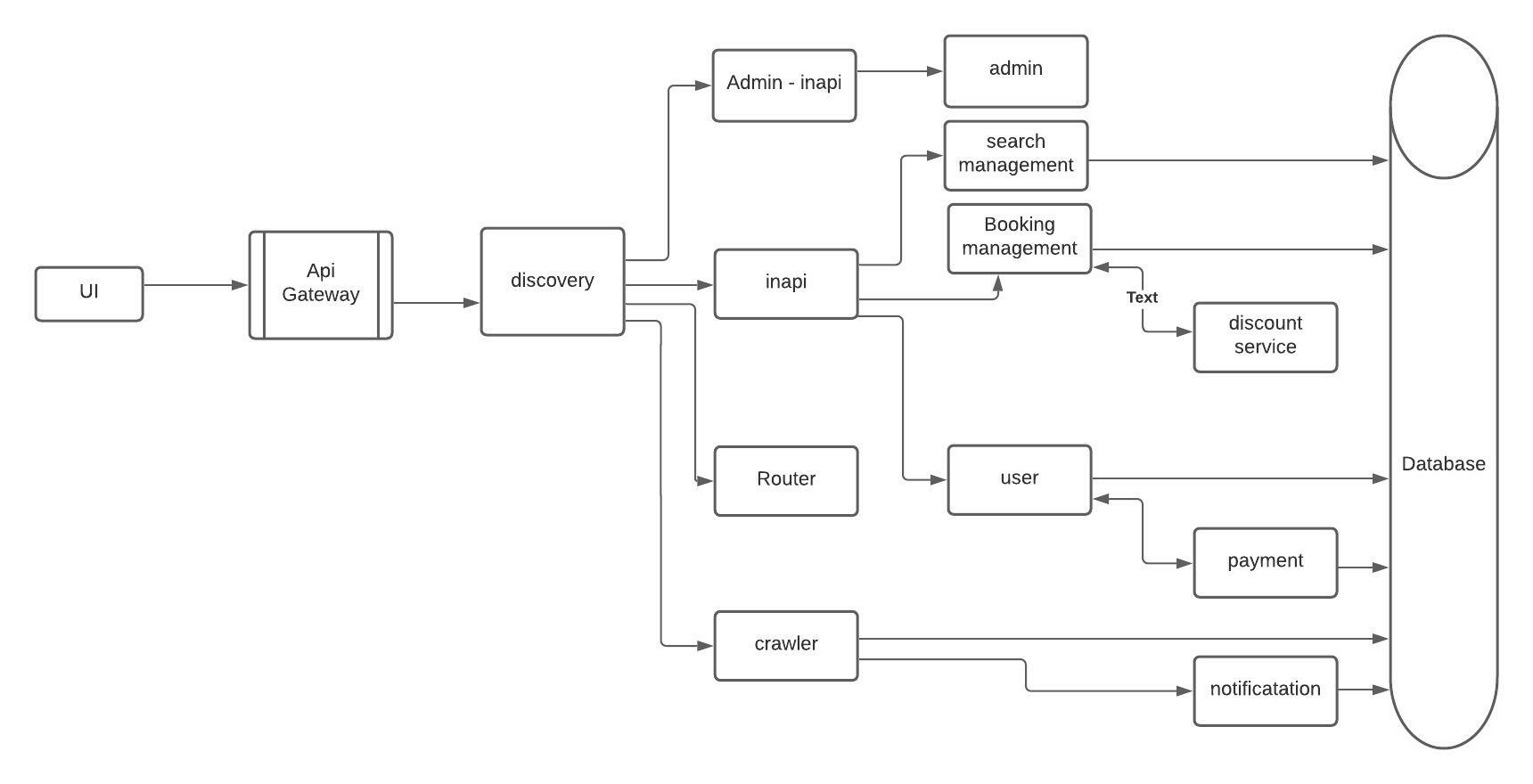
**Overview**

Create a movie ticket booking platform, which will cater both B2C and B2B

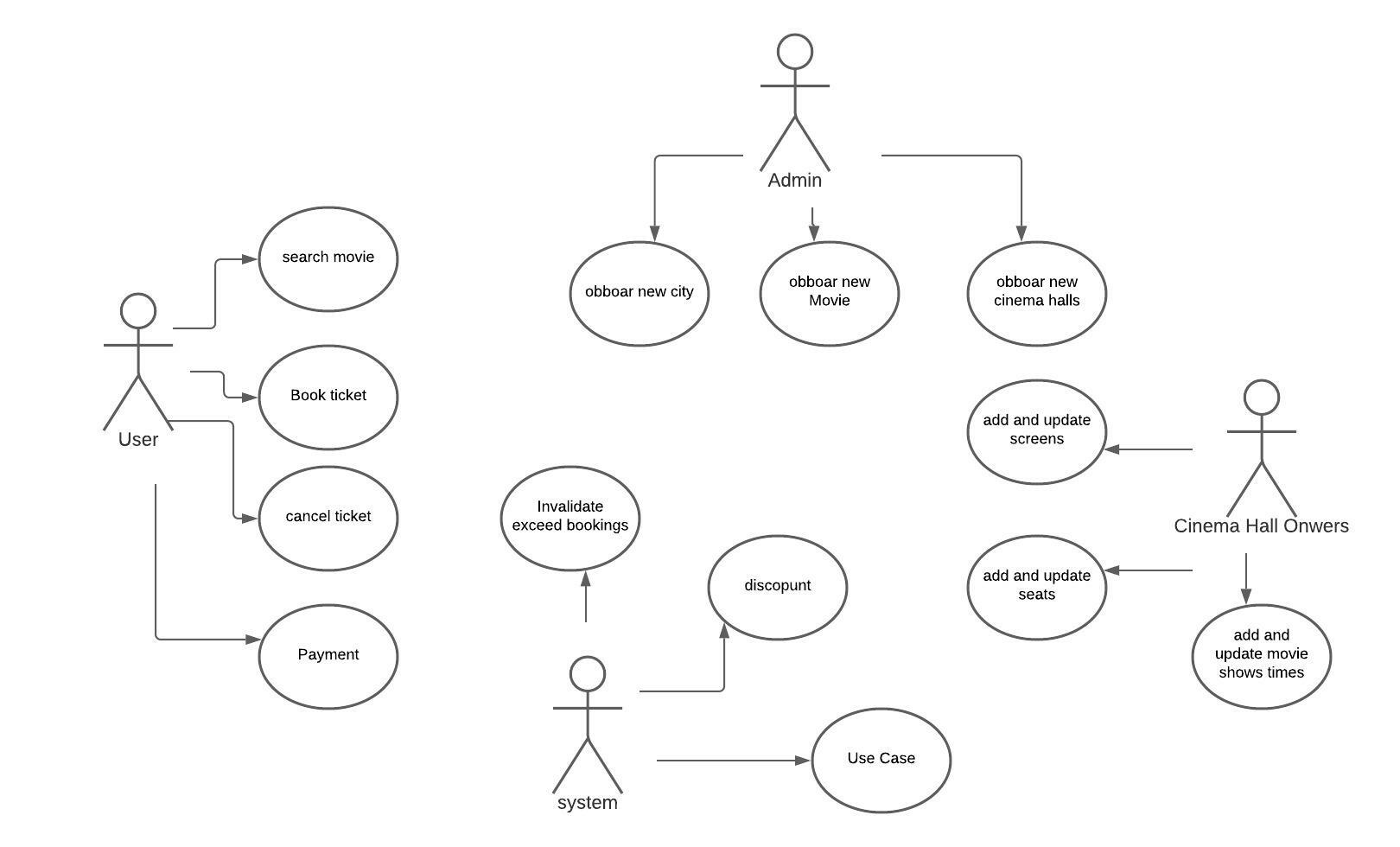
**Project Eye view**

****

**Project Layer view**

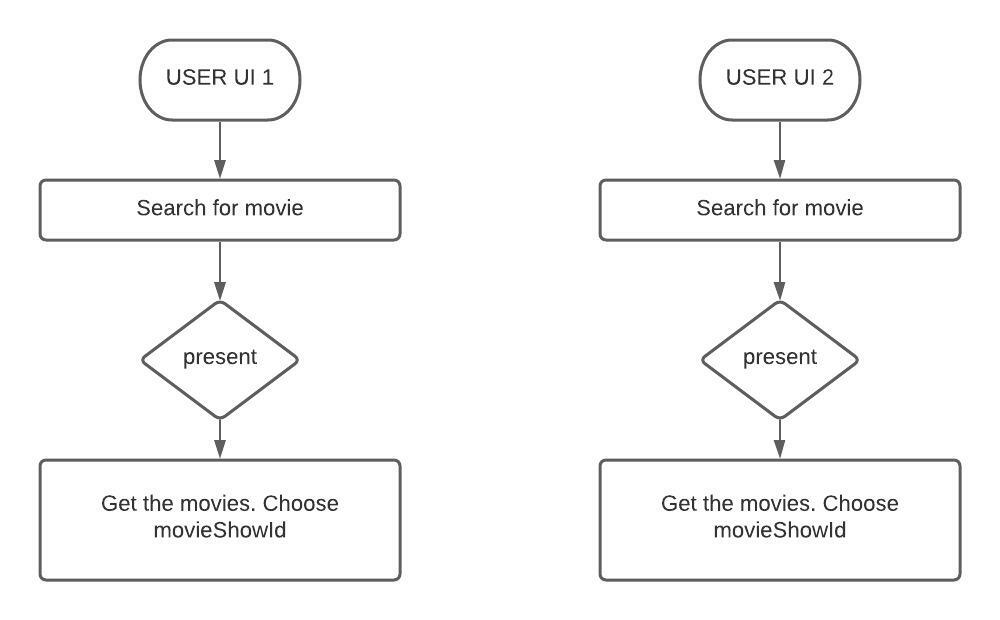
****

**User cases**

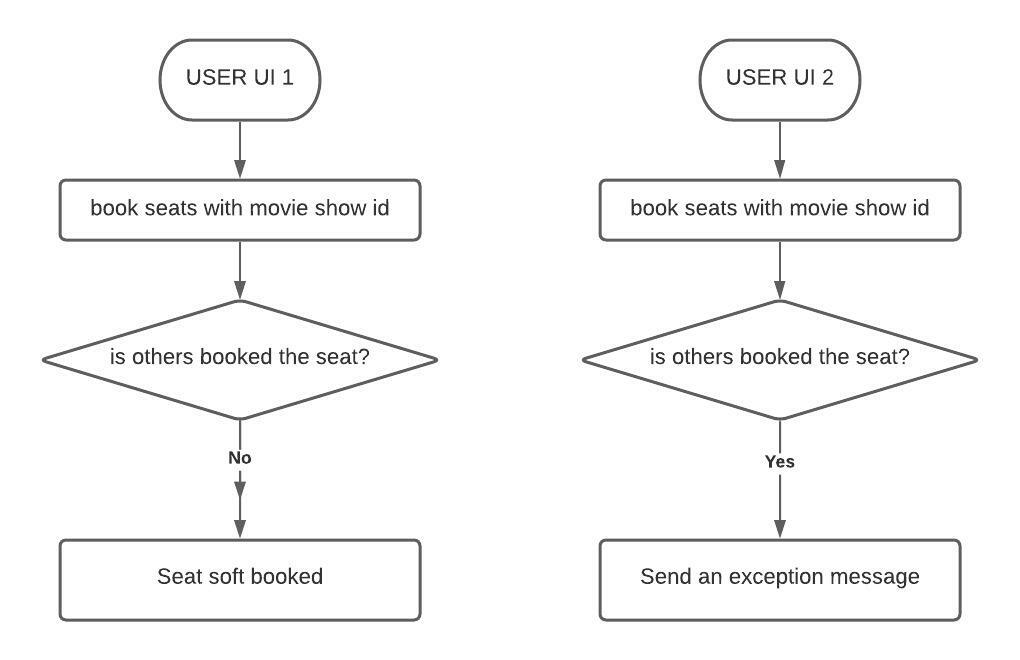
****

**Flowchart**

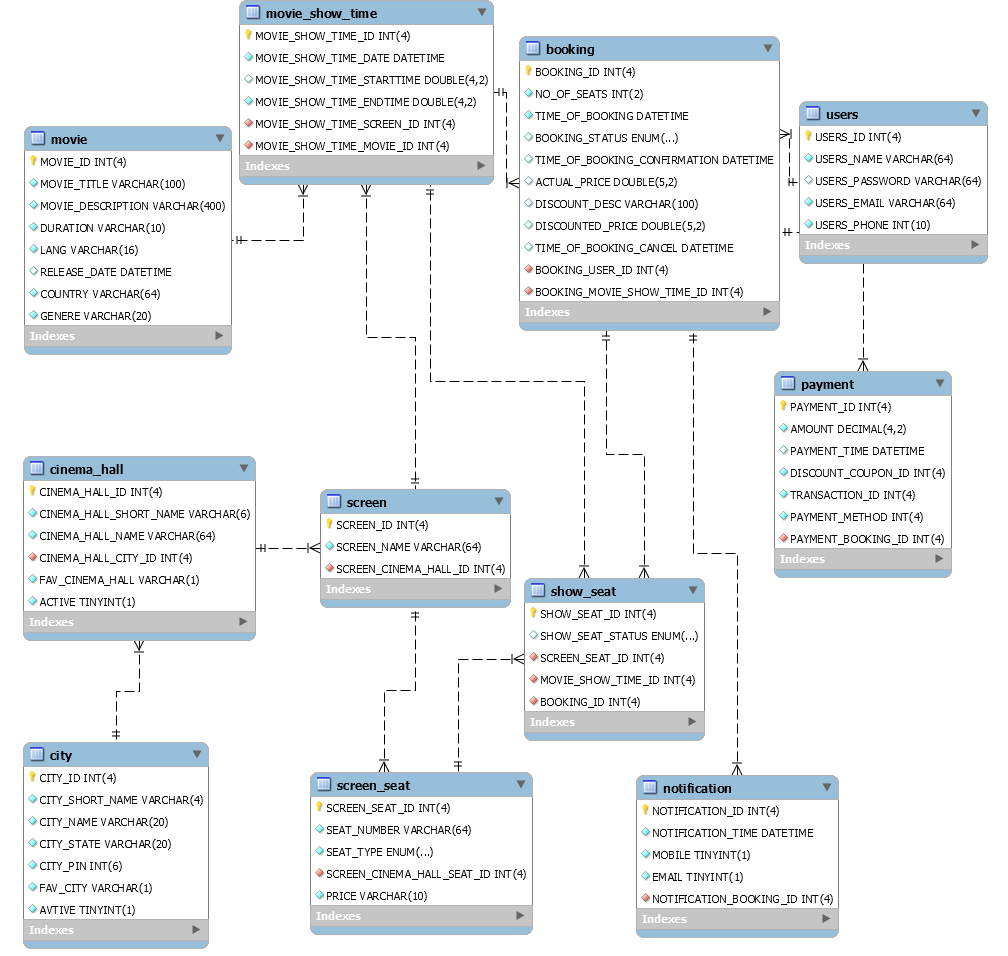
1. when two user searches a same movie in same city, screen and show time

****

2. when user try to book a seat



**Data Base modeling**

I have chosen RDBMS as there are structured visualization of data. In this case I have used mySql.

**Database transactions scenarios**

One transaction scenario I would like to say is "Booking of a seat". Scenarios can be

1. Me and Others are try to book a same seat -

* My seat will be reserved for me for 5minutes. Other will receive and error message. I did it with simple database search operations. Others will not be able to see the seats when searched. When a user checkout a booking a status(Pending) is maintained in both Booking and ShowSeat table with timestamp.

2. I am booking seat no say 1,2,3 and some other one has already taken 2,3,4. So, in two bookings seat no 2 & 3 is common.

* Same as point no 1

3. I have check out the bookings but got busy in some other works more than 5 minutes.

* A crawler service is there which checks of the exceeded timeframe ( i.e. 5 minute) every minute and make those booking invalidated. In booking table the status will be changed for the particular booking\_id.

I did it with db interactions. I want to reduce some db network calls, I would like to introduce **caching** to store the key of **date, screen\_id, seat\_id**.

For better isolation, we can use mysql "**for update**" and lock the rows pessimistically. Programmatically, we have ensured the isolation, this will give additional protection if anything goes wrong. Crawler then will have additional responsibly to release those locking.

**How Scale to platform availability?**

We need to setup some fault tolerant data centers like city or country wise and need to keep replica for each components.

**Monetizing platform**

We can open source our product. We will provide a platform where any other party who wants to consume our apis will register there and get a access\_token. This token, he will use as header when he is consuming our apis. We will keep a track of how many hits he has made and charge him after a limit.

**OWSAP**

1. Apis will be well managed and restricted based on roles.

**user** role will only access search apis, booking, payment

**owner** role will be accessing the theatre management api like add movieshow time etc.

**admin** will be accessing add movie, add city etc

2. Apis will be using authentication tokens with expiry like JWT, etc.

3. All modules will be responsible for their respective tasks, like booking service will provide only booking info not seaching and vice versa.

4. Crawler service is there to check the data validity

5. there are two entry api, admin-inapi and inapi. I have a plan to add another which is exapi. admin-inapi will access admin service, inapi will access booking, discount, payment, search. Exapi in future will manage user service. via user service we will access the search, some given api from payment service.

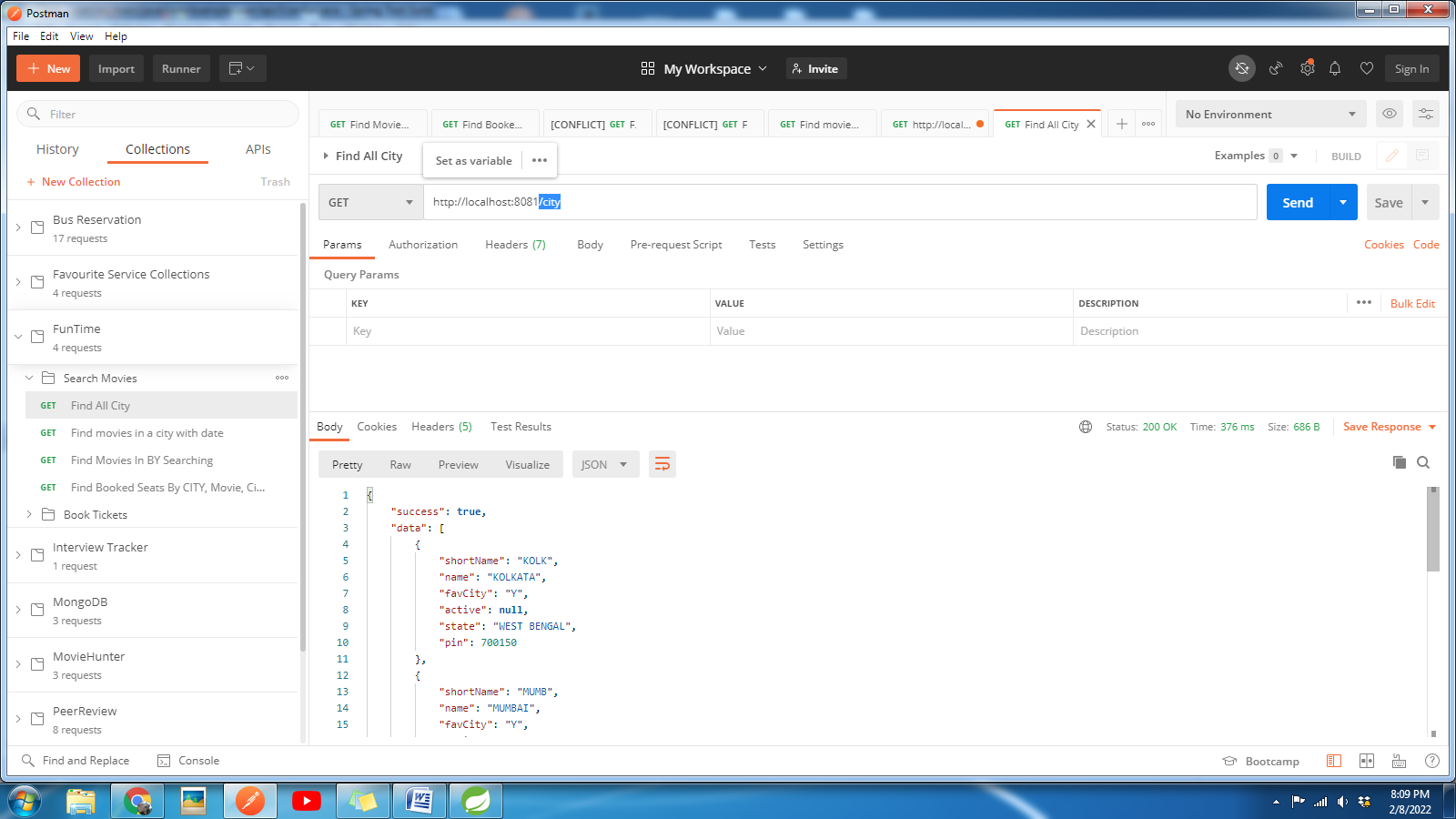
6. Data to be monitored, so setup some logs like grafanna, splunk or app dynamics

7. Always scan the project security threats with market standard scan engine like sonar, veracode , defend bot or cast.

**Rest Api Overviews Module Wise**

**Search Management (srchmgnt)**

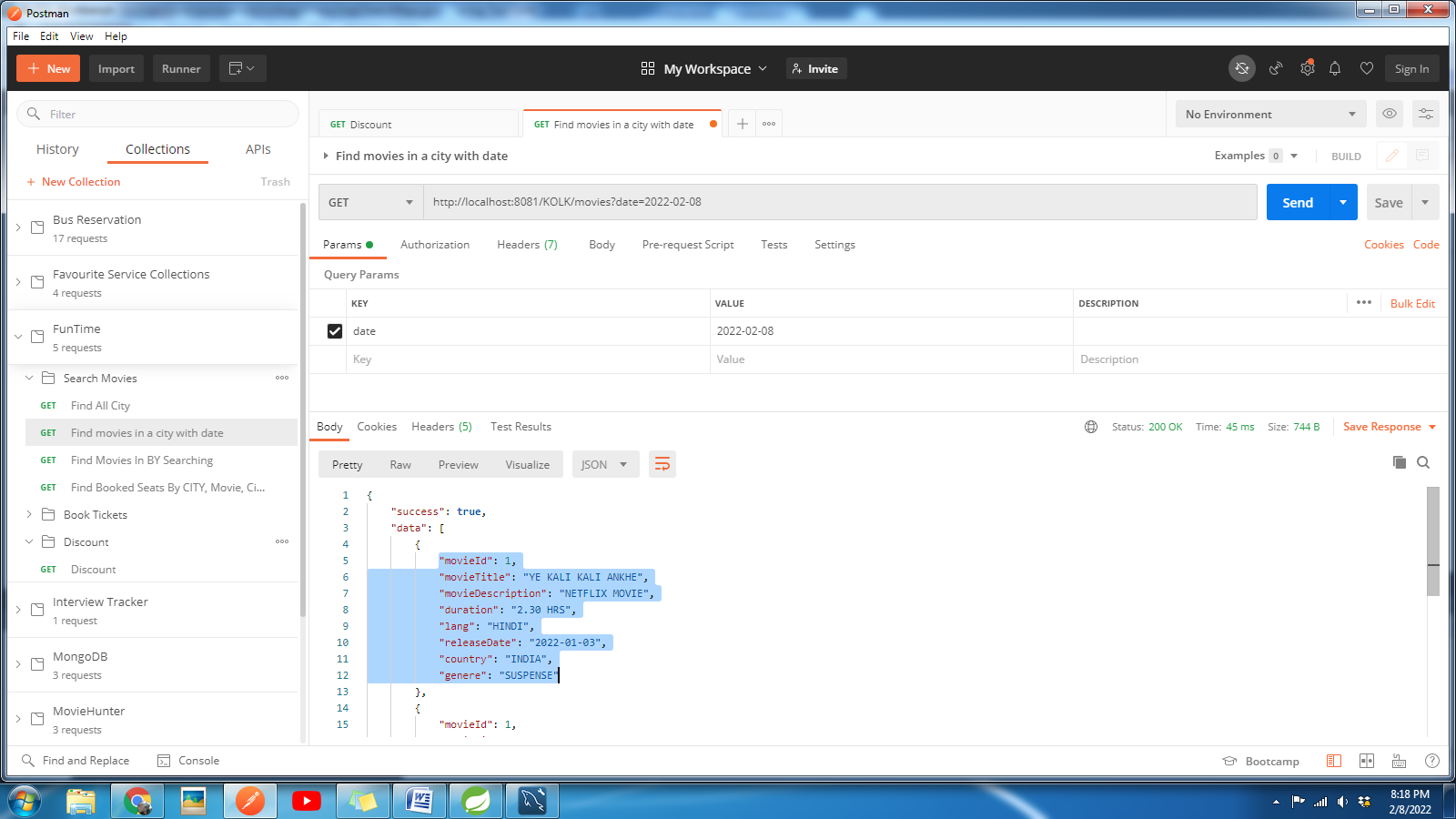
1. Search all CIty - /city

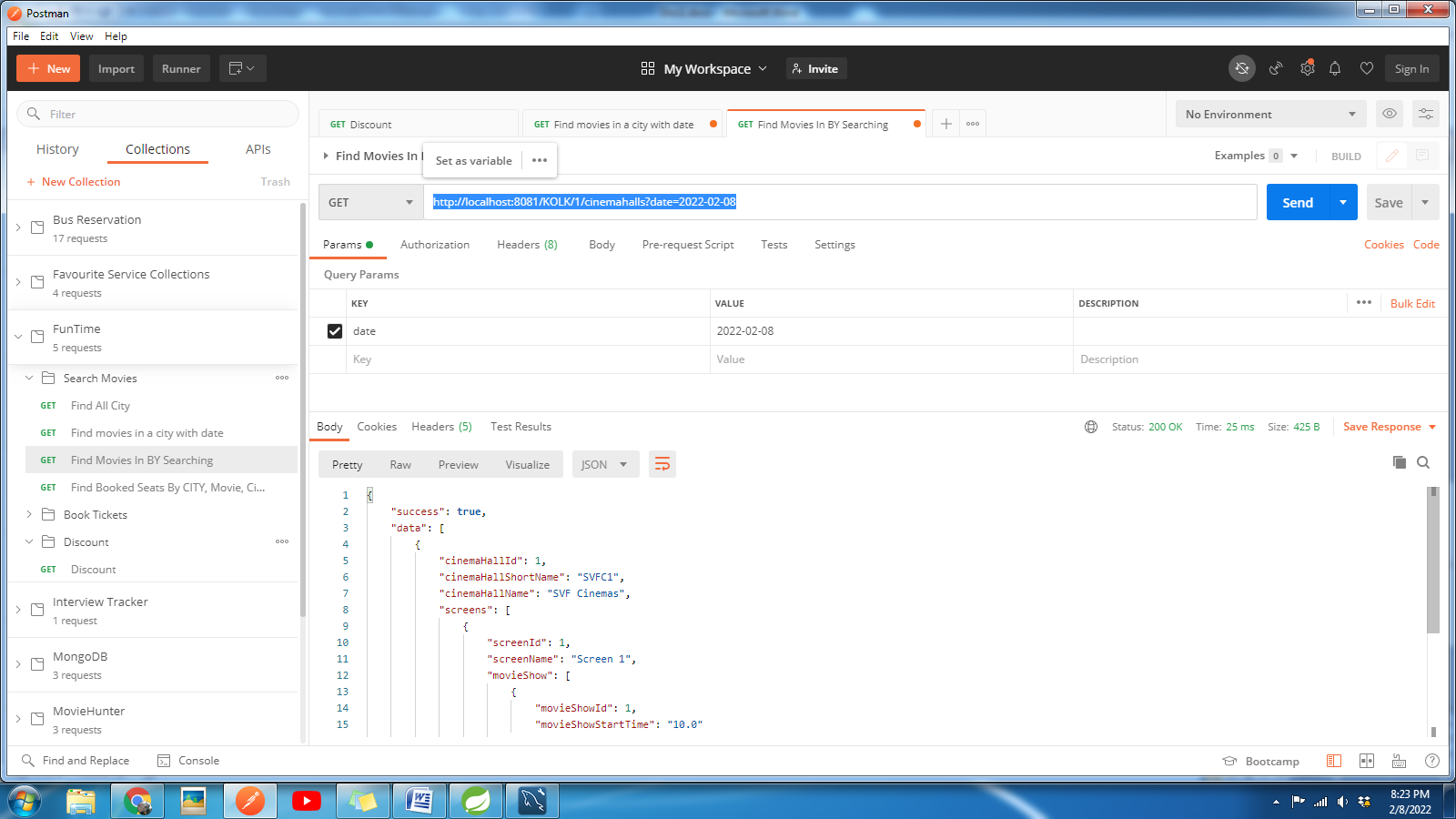
****

this will provide shortName that to be used in further Apis.

2. Search movie with city and date - http://localhost:8081/KOLK/movies?date=2022-02-08

I want to seach what are the movies currently in CITY in the given date.

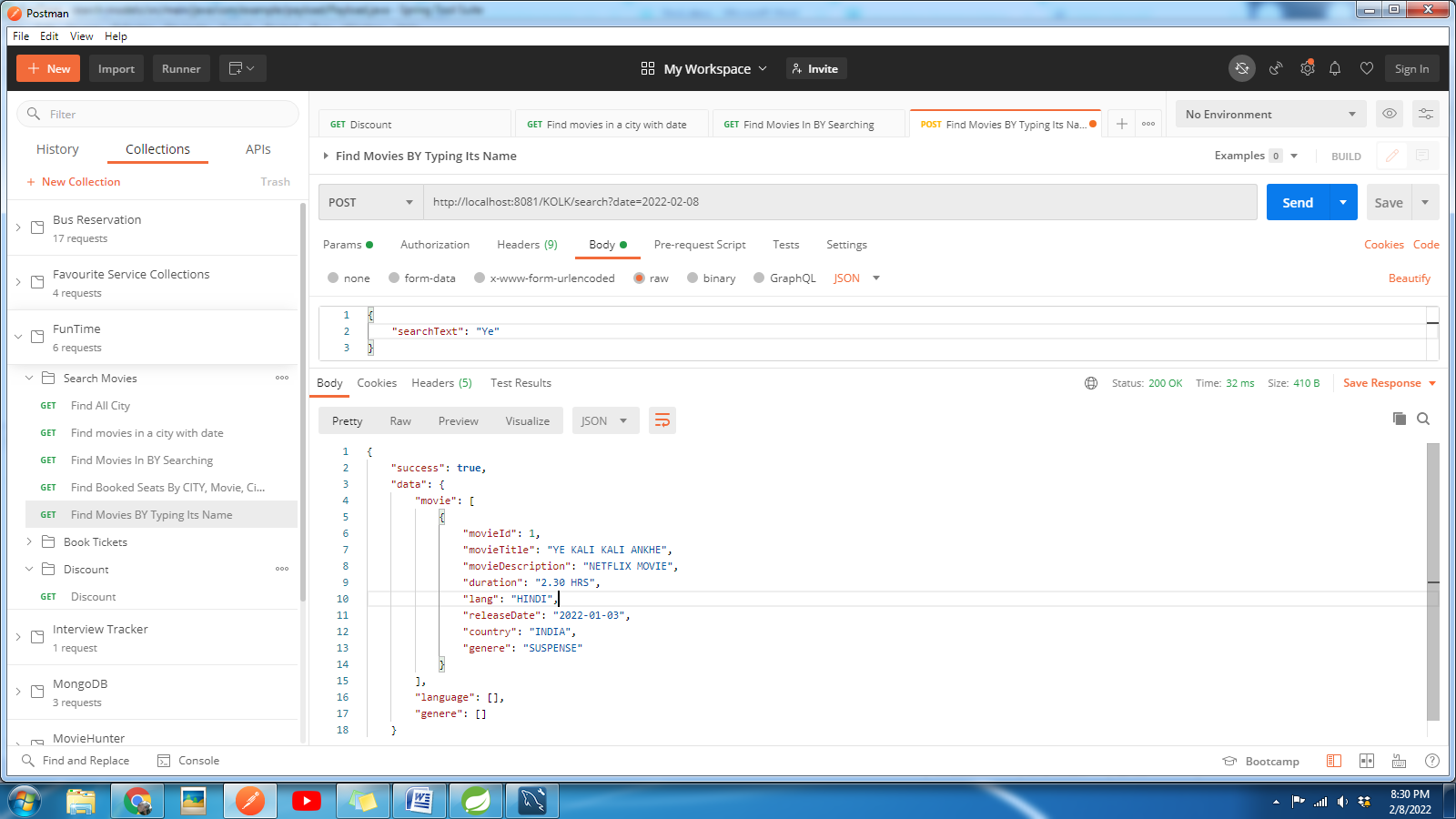
 movieId to be used

3. I want to search what are the movies currently in CITY and what are the cinema halls screening them. - /{city}/{movie}/cinemahalls? date

You will find the hall informations with Screens and and showTimings

4. As an user I want search movies by typing its name - /{city}/search? date

It will search movies by movies, lang and genere.



5. Check seat details of a show

Estimation

|  |
| --- |
| **Title** |
| [**Fun Time**](https://www.wrike.com/open.htm?id=842632816) |
| [**Phaze 1**](https://www.wrike.com/open.htm?id=842634186) |
| [**kick off**](https://www.wrike.com/open.htm?id=842634366) |
| [tech orientation](https://www.wrike.com/open.htm?id=842641537) |
| [high level architecture decision](https://www.wrike.com/open.htm?id=842641434) |
| [database design discussion](https://www.wrike.com/open.htm?id=842634561) |
| [kt](https://www.wrike.com/open.htm?id=842633119) |
| [**week 1**](https://www.wrike.com/open.htm?id=842634894) |
| [low level architecture](https://www.wrike.com/open.htm?id=842647924) |
| [KT with smes](https://www.wrike.com/open.htm?id=842647856) |
| [payment gateway decision](https://www.wrike.com/open.htm?id=842637059) |
| [setup repo and build](https://www.wrike.com/open.htm?id=842636830) |
| [setup db ddl](https://www.wrike.com/open.htm?id=842635459) |
| [**week 10**](https://www.wrike.com/open.htm?id=842648080) |
| [deploy to qa](https://www.wrike.com/open.htm?id=842648469) |
| [bug fix in qa](https://www.wrike.com/open.htm?id=842648388) |
| [uat setup](https://www.wrike.com/open.htm?id=842648260) |
| [**week 11**](https://www.wrike.com/open.htm?id=842648700) |
| [bugfix qa](https://www.wrike.com/open.htm?id=842648824) |
| [deploy to uat](https://www.wrike.com/open.htm?id=842648806) |
| [**week 12**](https://www.wrike.com/open.htm?id=842649055) |
| [bugfix uat](https://www.wrike.com/open.htm?id=842649104) |
| [sign off uat](https://www.wrike.com/open.htm?id=842649092) |
| [**week 13**](https://www.wrike.com/open.htm?id=842649185) |
| [release build](https://www.wrike.com/open.htm?id=842649245) |
| [sign off of for prod](https://www.wrike.com/open.htm?id=842649230) |
| [**week 14**](https://www.wrike.com/open.htm?id=842649384) |
| [prod release](https://www.wrike.com/open.htm?id=842649404) |
| [**Week 2**](https://www.wrike.com/open.htm?id=842637214) |
| [setup db mock dml](https://www.wrike.com/open.htm?id=842639092) |
| [setup dev](https://www.wrike.com/open.htm?id=842638751) |
| [discovery , router](https://www.wrike.com/open.htm?id=842638425) |
| [mock user service](https://www.wrike.com/open.htm?id=842638110) |
| [mock serch](https://www.wrike.com/open.htm?id=842637899) |
| [deploy mock service to dev and test](https://www.wrike.com/open.htm?id=842637389) |
| [**week 3**](https://www.wrike.com/open.htm?id=842640090) |
| [admin service](https://www.wrike.com/open.htm?id=842640789) |
| [user service](https://www.wrike.com/open.htm?id=842640683) |
| [serach](https://www.wrike.com/open.htm?id=842640624) |
| [**week 4**](https://www.wrike.com/open.htm?id=842640889) |
| [authentication](https://www.wrike.com/open.htm?id=842641390) |
| [booking](https://www.wrike.com/open.htm?id=842641332) |
| [theatre](https://www.wrike.com/open.htm?id=842641097) |
| [**week 5**](https://www.wrike.com/open.htm?id=842641690) |
| [payment module work start](https://www.wrike.com/open.htm?id=842641980) |
| [testing and bug fix](https://www.wrike.com/open.htm?id=842641889) |
| [deploy to dev](https://www.wrike.com/open.htm?id=842641826) |
| [**week 6**](https://www.wrike.com/open.htm?id=842642735) |
| [testing and bug fix](https://www.wrike.com/open.htm?id=842644790) |
| [inaps modules setup](https://www.wrike.com/open.htm?id=842644345) |
| [notificataion server setups](https://www.wrike.com/open.htm?id=842643847) |
| [notification service](https://www.wrike.com/open.htm?id=842643629) |
| [**week 7**](https://www.wrike.com/open.htm?id=842645468) |
| [test env setup](https://www.wrike.com/open.htm?id=842645958) |
| [testing and bugfix in dev](https://www.wrike.com/open.htm?id=842645705) |
| [**week 8**](https://www.wrike.com/open.htm?id=842646348) |
| [test and bug fix in test](https://www.wrike.com/open.htm?id=842647037) |
| [deploy to test](https://www.wrike.com/open.htm?id=842646523) |
| [**week 9**](https://www.wrike.com/open.htm?id=842647295) |
| [qa env setup](https://www.wrike.com/open.htm?id=842648165) |
| [test and bug fix the test](https://www.wrike.com/open.htm?id=842647449) |