Service management service:

Class diagram:

A diagram of a service

Description automatically generated

Features:

1. Any User should be able to add service in system.
2. Any user should be able to retrieve service details.
3. User who added service/ Admin should be able to update it.
4. User who added service/ Admin should be able to delete it.
5. User should be able to add photos to service.
6. Any user should be able to search a service (text search + phone number).
7. Any user should be able to filter services based on categories.
8. User can put one category or many categories for filtering services.
9. Any user should be able to filter services based on ratings.
10. User can put one rating or range of rating to filter services.

API Contracts:

**Any User should be able to add service in system.**

POST /services/

Request:

Header:

{

“Authorization”: Bearer <token>

}

body:

{

"name": "name",

"phone\_number": "number",

"email\_id": "email",

"address": {

"address\_line1": "line1",

"address\_line2": "line2",

"city": "city",

"postal\_code": "zip",

"state": "ST",

"country": "country"

},

"website": "website",

"service\_owner\_name": "owner\_name",

"category\_ids": [1,2,3],

"google\_map\_link": "map\_link",

"location": {

"lat": -33.866,

"lng": 151.232

},

"weekly\_open\_periods": [

{"day": 1, "open": "0900", "close": "1700"},

{"day": 2, "open": "0900", "close": "1700"},

{"day": 4, "open": "0900", "close": "1700"},

{"day": 5, "open": "0900", "close": "1700"},

{"day": 6, "open": "0900", "close": "1700"}

],

"service\_state": "OPERATIONAL"

}

Response:

Status: 200

body:

{

"service\_id": 1,

"phone\_number": "number",

"email\_id": "email",

"address": {

"address\_line1": "line1",

"address\_line2": "line2",

"city": "city",

"postal\_code": "zip",

"state": "ST",

"country": "country"

},

"website": "website",

"service\_owner\_name": "owner\_name",

"category\_ids": [1,2,3],

"google\_map\_link": "map\_link",

"creator\_id": 1,

"service\_state": "OPERATIONAL",

"open\_now": true,

"weekly\_open\_periods": [

{"day": 1, "open": "0900", "close": "1700"},

{"day": 2, "open": "0900", "close": "1700"},

{"day": 4, "open": "0900", "close": "1700"},

{"day": 5, "open": "0900", "close": "1700"},

{"day": 6, "open": "0900", "close": "1700"}

],

"photos":[

{

"id": 1,

"url": "url1"

},

{

"id": 2,

"url": "url2"

}

],

"rating": 0,

"reviews": [

{

"author\_name": "name",

"author\_url": "url to fetch author details",

"profile\_photo\_url": "url for author photo",

"language": "en",

"rating": 1,

"relative\_time\_description": "a week ago",

"text": "actual review comments",

"time": 1652286798

}

],

"total\_user\_ratings": 500

}

**Any user should be able to retrieve service details.**

GET /services/{service\_id}

Header:

{

“Authorization”: Bearer <token>

}

Request params: service\_id: UUID

Response:

status: 200 OK

body:

<same as POST>

**User who added service/ Admin, should be able to update service in system.**

PUT /services/{service\_id}

Request:

Header:

{

“Authorization”: Bearer <token>

}

Request params: service\_id: UUID

Body:

<same as POST, with updated values>

Response:

status: 200 OK

body:

<same as POST>

**User who added service/ Admin, should be able to delete service in system.**

DELETE /services/{service\_id}

Request:

Header:

{

“Authorization”: Bearer <token>

}

Request params: service\_id: UUID

Response:

status: 200 OK

body:

{

“service\_id”: UUID,

“message”: “Service deleted successfully.”

}

**User should be able to add photos to a service.**

POST / services/{service\_id}/photos

Request:

Header:

{

“Authorization”: Bearer <token>

“Content-type”: multipart/form-data

}

Params: service\_id: UUID

Body:

{

“file”: {file1.jpg, file2.jpg}

}

Response:

status: 200 OK

body:

{

service\_id: 1,

photo:[

{

"id": 1,

"url": "url1"

},

{

"id": 2,

"url": "url2"

}

]

}

**Any user should be able to search a service.**

GET /services/search/&input=museum&inputtype=textquery

Request:

Header:

{

“Authorization”: Bearer <token>

}

Params:

input: text to search (for name, address, category name)

Response:

Status 200 Ok

Body:

{

total\_count: 2,

services:[

{

"service\_id": 1,

"name": "name1",

"formatted\_address": "full address1",

"category": {"cat1", "cat2", "cat3"},

"open\_now": true

"phone\_number": 9876543210,

"website": "url1",

"google\_map\_link": "link1",

"service\_url": "url to get service1",

"rating": 4.5,

"reviews": [

...

]

},

{

"service\_id": 2,

"name": "name2",

"formatted\_address": "full address2",

"category": {"cat5", "cat3"},

"open\_now": false

"phone\_number": 9876543230,

"website": "url2",

"google\_map\_link": "link2",

"service\_url": "url to get service2",

"rating": 3,

"reviews": [

...

]

}

]

}

**Any user should be able to filter services based on categories.**

GET /services/search/&input=museum&categories=cat1,cat2,cat3

Request:

Header:

{

“Authorization”: Bearer <token>

}

Params:

input: text to search (for name, address)

categories: comma separated list of categories name.

Response

status: 200 ok

body:

{

total\_count: 2,

services:[

{

"service\_id": 1,

"name": "name1",

"formatted\_address": "full address1",

"category": {"cat1", "cat2", "cat3"},

"open\_now": true

"phone\_number": 9876543210,

"website": "url1",

"google\_map\_link": "link1",

"service\_url": "url to get service1",

"rating": 4.5,

"reviews": [

...

]

},

{

"service\_id": 2,

"name": "name2",

"formatted\_address": "full address2",

"category": {"cat5", "cat3"},

"open\_now": false

"phone\_number": 9876543230,

"website": "url2",

"google\_map\_link": "link2",

"service\_url": "url to get service2",

"rating": 3,

"reviews": [

...

]

}

]

}

**Any user should be able to filter services based on ratings.**

GET /services/search/&input=museum&ratings=4,5

Request:

Header:

{

“Authorization”: Bearer <token>

}

Params:

input: text to search (for name, address)

ratings: comma separated list of ratings.

Response:

status: 200 ok

body:

{

total\_count: 2,

services:[

{

"service\_id": 1,

"name": "name1",

"formatted\_address": "full address1",

"category": {"cat1", "cat2", "cat3"},

"open\_now": true

"phone\_number": 9876543210,

"website": "url1",

"google\_map\_link": "link1",

"service\_url": "url to get service1",

"rating": 4.5,

"reviews": [

...

]

}

]

}

**Error response for all API requests.**

status: 400 Bad Request

{

"code" : 4002,

"message" : "Invalid request",

"dev\_message" : "GET <API url> - No handler found.",

}