Linda Forlizzi 7/20/22
Promineo Tech Final Project Document

1. Include the GitHub Repo URL of your Project

https://github.com/thisLinda/dysphagia-med-reactions-db

2. *** DO NOT Screenshot Your Java Code! ***

3. Include Screenshots of the RESULTS of RUNNING each path in your WebAPI working: (e.g. GET Users, GET a User, CREATE a User, DELETE a User, UPDATE a User -- WITH RESULTS)

- Using Swagger — show the API Documentation, the expected HTTPStatuses, & the actual results of EACH Operation on each URL.
- Alternatively, use Postman, or another interface — show the URL, and the input, as well as the results for EACH Operation on each URL.

Medication Reaction Service

/v3/api-docs

default-user-controller ⌃

| GET | /user | Returns a list of users | ⌄ |

| PUT | /user | Updates an existing user | ⌄ |

| POST | /user | Creates a user | ⌄ |

| DELETE | /user | Deletes a user | ⌄ |

default-user-med-list-controller ⌃

| POST | /usermedlist | Creates a user and a medication | ⌄ |

default-reaction-controller ⌃

| GET | /reactions | Returns a severity and an adverse reaction. | ⌄ |

Name	Description

userPK * required
Integer($int32)
(query)

userPK

```
1
```

Execute

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/user?userPK-1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/user?userPK=1
```

Server response

Code	Details

200

Response body

```json
[
  {
    "pseudoName": "ClaireRF",
    "age": 50,
    "dateOfEval": "01/01/21",
    "dateOfDischarge": "02/02/21",
    "medDxIcd": "I69.391",
    "txDxIcd": "R13.12"
  }
]
```

userPK • required
integer($int32)
(query)

userPK

100

Execute	Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/user?userPK=100' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/user?userPK=100
```

Server response

Code	Details
404	Error: response status is 404

Response body

```
{
  "reason": "Not Found",
  "status code": 404,
  "message": "java.util.NoSuchElementException: No users found with userPK=100",
  "url": "/user",
  "timestamp": "Tue, 19 Jul 2022 13:24:17 -0400"
}
```

Response headers

Name	Description
newPseudoName * required string (query)	Linda
userPK * required integer($int32) (query)	1

Execute	Clear

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/user?newPseudoName=Linda&userPK=1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/user?newPseudoName=Linda&userPK=1
```

Server response

Code	Details

200
Undocumented
	Response headers
	```connection: keep-alive```
	```content-length: 0```
	```date: Tue,19 Jul 2022 17:25:40 GMT```
	```keep-alive: timeout=60```

integer($int32)   userPK
(query)

```
1
```

Execute	Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/user?userPK=1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/user?userPK=1
```

Server response

Code	Details

200

Response body

```
[
  {
    "pseudoName": "Linda",
    "age": 50,
    "dateOfEval": "01/01/21",
    "dateOfDischarge": "02/02/21",
    "medDxIcd": "I69.391",
    "txDxIcd": "R13.12"
  }
]
```

Response headers

Parameters

Name	Description
userPK * required integer($int32) (query)	16
pseudoName * required string (query)	notLinda
age * required integer($int32) (query)	16
dateOfEval * required string (query)	7/19/22
dateOfDischarge string (query)	dateOfDischarge
medDxIcd * required string (query)	R42
txDxIcd * required string (query)	R41.841

Execute	Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/user?userPK-16&pseudoName-notLinda&age-16&dateOfEval-7%2F19%2F22&medOx
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://localhost:8080/user?userPK-16&pseudoName-notLinda&age-16&dateOfEval-7%2F19%2F22&medDxIod
```

Server response

Code	Details
201	**Response body**

```
1
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Tue,19 Jul 2022 17:29:22 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```
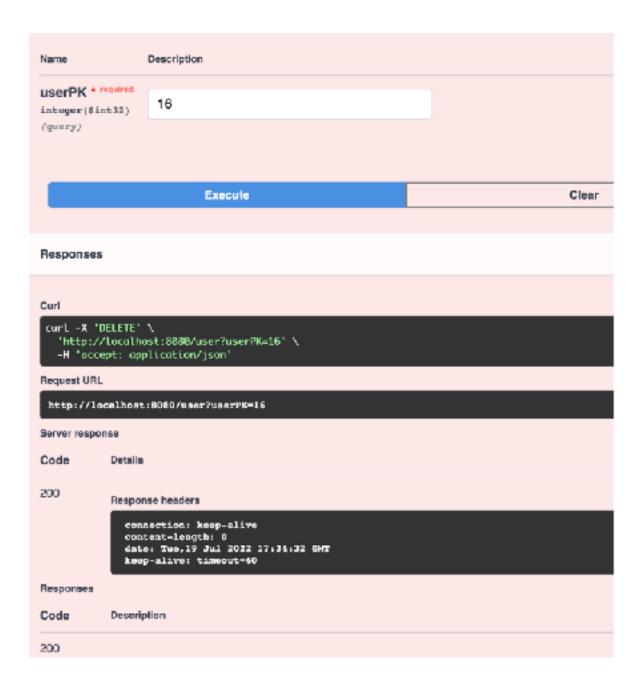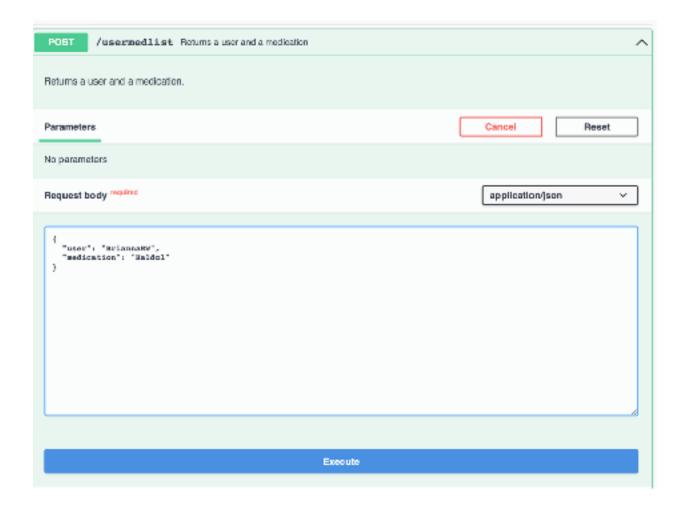
Responses

Code	Description
201	The created user is returned.

userPK • required

integer($int32)

(query)

userPK

```
16
```

Execute	Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/user?userPK=16' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/user?userPK=16
```

Server response

Code	Details
200	Response body

```
[
  {
    "pseudoName": "notLinda",
    "age": 16,
    "dateOfEval": "7/19/22",
    "dateOfDischarge": null,
    "medDxIcd": "R42",
    "txDxIcd": "R41.841"
  }
]
```

Response headers

Name	Description

userPK ▲ required
integer($int32)
(query)

```
16
```

Execute	Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/user?userPK=16' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/user?userPK=16
```

Server response

Code	Details

200

Response headers

```
connection: keep-alive
content-length: 0
date: Tue,19 Jul 2022 17:34:32 GMT
keep-alive: timeout=60
```

Responses

Code	Description

200

Name	Description
userPK * required integer($int32) *(query)*	userPK 16

Execute	Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/user?userPK=16' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/user?userPK=16
```

Server response

Code	Details
404	Error: response status is 404 Response body

```
{
  "reason": "Not Found",
  "status code": 404,
  "message": "java.util.NoSuchElementException: No users found with userPK=16",
  "uri": "/user",
  "timestamp": "Tue, 19 Jul 2022 13:35:01 -0400"
}
```

POST **/usermedlist** Returns a user and a medication

Returns a user and a medication.

Parameters | | Cancel | Reset

No parameters

Request body ^{required}   application/json ⌄

```
{
  "user": "BriannaRN",
  "medication": "Haldol"
}
```

Execute

Curl

```
curl -X 'POST' \
  'http://localhost:8080/usermedlist' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "user": "BriannaRW",
  "medication": "Haldol"
}'
```

Request URL

```
http://localhost:8080/usermedlist
```

Server response

Code	Details
201	**Response body**

```
{
  "pseudoName": {
    "pseudoName": "BriannaRW",
    "age": 31,
    "dateOfEval": "05/05/21",
    "dateOfDischarge": "06/06/21",
    "medDxIcd": "G31.83",
    "txDxIcd": "R13.11"
  },
  "brandName": {
    "brandName": "Haldol",
    "hasReaction": true
  }
}
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Tue,19 Jul 2021 17:36:41 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

reaction * required

string
(query)

The reaction as JSON

tardive dyskinesia

reactionSeverity * required

string
(query)

SEVERE ⌄

Execute	Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/reactions?reaction=tardive%20dyskinesia&reactionSeverity=SEVERE' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/reactions?reaction=tardive%20dyskinesia&reactionSeverity=SEVERE
```

Server response

Code	Details
200	Response body

```
[
  {
    "reactionSeverity": "SEVERE",
    "reaction": "tardive dyskinesia"
  }
]
```

Download

4. Include your Database Design ERD!

5. Include your own version of Web API Documentation:
For each URL, list:

- The allowed Requests (e.g. GET, POST, PUT, DELETE)
- The input required, and any path variables
- The expected outcome!
- Any appropriate HTTPStatuses, Error Messages, etc.

HTTP Method	Endpoint	Expected Outcome/Status
Get	/user	Required: userPK Description: list of user
Put	/user	Required: newPseudoName Description: updated user
Post	/user	Required: userPK, pseudoName, age, dateOfEval, (not required) dateOfDischarge, medDxIcd, txDxIcd Description: created user
Delete	/user	Required: userPK Description: deleted user
Post	/usermedlist	Required: user, medication Description: created user/ medication is returned
Get	/reactions	Required: reaction, severity Description: returns a reaction and its severity

REQUIREMENTS
1-person Project

- Database design which contains at least 3 entities and 3 tables
    - Entities: Reaction, Medication, User
    - Tables: generics, user_meds_list, med_adverse_reactions, adverse_reactions, medications, users
- Contains all CRUD operations (Create, Read, Update & Delete)
    - User: create, read, update, delete
    - Medication: create
    - Reactions: read
- **Each entity** should have **CRUD operations** with one entity having all 4 CRUD operations (Create, Read, Update & Delete).
    - see above
- Contains at least 1 one-to-many relationship
    - generics
- Contains at least 1 many-to-many relationship with one or more CRUD operations on this relationship.
    - med_adverse_reactions
    - user_meds_list