Michael Bamesberger
CS 496
Assignment 3: Part 2
April 24, 2016

Patient-Doctor Database

http://bamesberger-assignment-3.appspot.com

URL structure:

/patient
With a GET request, this URL displays the id of all entities in the Patient database. With a POST request, this allows a user to add a Patient to the database.

/patient/<id>
With a GET request, this URL displays all fields of Patient object specified by the id

/patient/search
With a POST request, this URL allows the user to search the Patient database by the entity's lastName variable

/patient/delete/<id>
With a DELETE request, this URL allows the user to delete a Patient object by specifying the id

/patient/update
With a PUT request, this URL allows the user to update various fields in a Patient object, as long as the "key" field contains the correct id to the object

/doctor
With a GET request, this URL displays all the Doctor objects in the database by id. With a POST request, this URL allows a user to add a Doctor object to the database

/doctor/<id>
With a GET request, this URL displays all fields of the Doctor object specified in the id

/doctor/delete/<id>
With a DELETE request, this URL allows the user to delete a Doctor object by specifying an id

RESTful constraints:

I did meet several of the RESTful requirements. I did separate the concerns of the client and server. My API is also stateless. My system is also layered (technically), because the App Engine scales and shards without the client knowing. Finally, I meet the uniform interface resource identifiers (though the client can only request json) as well as representation.

Because of time constraints (and constraints of my knowledge), there are many RESTful requirements I did not meet. For example, my resources were not marked as cacheable or non-cacheable. My messages are not self descriptive.

Changes to schema:

My idea was to create a patient medical record look-up system, for use in a hospital or among many hospitals. A key (or keys) to the doctor entity is included in the patient's information, so that there is a record of who treated the patient, and would be a resource to someone (like another doctor) who is looking for more information on a patient's condition or diagnoses.

In my original plan, I made the Doctor Entities children of the Patient Entities. However, for this API, I decided against that. As I began working, I realized that this is likely a many-to-many relationship sort of database (to borrow a term from SQL databases), though it will likely be used most for looking up Doctors only through a Patient. And, because of some time constraints, I decided to simply create a list of Doctor keys in the Patient entity. This was a simpler way to keep track of which doctors are related to which patient.

What I might do differently:

This was a very difficult assignment for me, because I had to come up to speed on how to use Python, Google App Engine and non-relational databases. If I were to do this assignment differently, I might have picked two entities which have fewer data fields and which have an easier -- or to say, more obvious -- relationship. For example, a Customer to Purchase database, I feel, would be less work to implement, because it is an obvious one to many relationship with fewer data points.

Finally, I was unable to implement a mechanism for cleaning up a Patient's doctor key after the referenced doctor has been deleted. I had trouble figuring out how to remove a single key from a Patient's doctor key list, so I scrapped the feature.

Test results:

PATIENT TESTS:

To add a Patient with only required data to the database: PASS

```
curl --data "firstName=Britney&lastName=Jones&age=18&gender=female&checkedIn=True" -H "Accept:
application/json" http://bamesberger-assignment-3.appspot.com/patient
```

Result:

```
{"lastVisit": null, "lastName": "Jones", "roomNo": null, "diseaseHistory": [], "medications": [], "allergies": [],
"age": 18, "key": 5733953138851840, "gender": "female", "condition": null, "doctors": [], "firstName": "Britney",
"checkedIn": true}
```

To add a Patient without all required data: PASS

```
curl --data "firstName=Joe&gender=male&condition=stable&diseaseHistory=asthma,
diabetes&medications=asprin&allergies=penicillin" -H "Accept: application/json"
http://bamesberger-assignment-3.appspot.com/patient
```

Result: entry not added to database (though no error message displayed)

To add a Patient with wrong checkedIn type (accepts True, true, yes, Yes or No, no, False, false): PASS

```
curl --data "firstName=Tina&lastName=Ryan&age=44&gender=female&checkedIn=Y" -H "Accept:
application/json" http://bamesberger-assignment-3.appspot.com/patient
```

Result: Information not added (though no error message was displayed)

To add a Patient without “Accept: application/json”: FAIL

```
curl --data "firstName=Craig&lastName=Brown&age=25&gender=male&checkedIn=False" -H “Accept:
text/html” http://bamesberger-assignment-3.appspot.com/patient
```

Result:

```
curl: (6) Could not resolve host: text
{"condition": null, "age": 25, "checkedIn": false, "medications": [], "diseaseHistory": [], "lastVisit": null, "doctors": [],
"key": 5118776383111168, "allergies": [], "roomNo": null, "gender": "male", "lastName": "Brown", "firstName": "Craig"
```

To display the keys of all Patients in the database: PASS

```
curl -H "Accept: application/json" http://bamesberger-assignment-3.appspot.com/patient
```

Result:

```
{"keys": [5047308127305728, 5118776383111168, 5328783104016384, 5681726336532480,
5733953138851840, 5838406743490560, 5891733057437696, 6454683010859008]}
```

To search a Patient by valid ID: PASS

```
curl -H "Accept: application/json" http://bamesberger-assignment-3.appspot.com/patient/6454683010859008
```

Result:

```
{"age": 25, "lastName": "Brown", "roomNo": null, "diseaseHistory": [], "medications": [], "allergies": [], "lastVisit": null,
"checkedIn": false, "firstName": "Craig", "condition": null, "gender": "male", "doctors": [], "key": 6454683010859008}
```

To search Patient by invalid ID: PASS

```
curl -H "Accept: application/json" http://bamesberger-assignment-3.appspot.com/patient/7454683015859008
```

Result:

```
   <h1>Internal Server Error</h1>
   <p>The server has either erred or is incapable of performing
   the requested operation.</p>
```

To update a Patient:

```
curl -X PUT -H "Accept: application/json" -d "key=6454683010859008&firstName=Jorge"
http://bamesberger-assignment-3.appspot.com/patient/update
```

Result:

```
{"age": 25, "lastName": "Brown", "roomNo": null, "diseaseHistory": [], "medications": [], "allergies": [], "lastVisit": null,
"checkedIn": false, "firstName": "Jorge", "condition": null, "gender": "male", "doctors": [], "key": 6454683010859008}
```

To search a Patient by valid lastName: PASS

```
curl --data "lastName=Jones" -H "Accept: application/json"
http://bamesberger-assignment-3.appspot.com/patient/search
```

Result:

```
{"condition": null, "age": 18, "checkedIn": true, "medications": [], "diseaseHistory": [], "lastVisit": null, "doctors":
[], "key": 5733953138851840, "allergies": [], "roomNo": null, "gender": "female", "lastName": "Jones", "firstName":
"Britney"}
```

To search a Patient by invalid lastName: PASS

```
curl --data "lastName=Hin" -H "Accept: application/json"
http://bamesberger-assignment-3.appspot.com/patient/search
```

Result: No information comes up

To delete a Patient by key: PASS

```
curl -X DELETE -H "Accept: application/json"
http://bamesberger-assignment-3.appspot.com/patient/delete/5733953138851840
```

Result: Key and information gone from database

DOCTOR TESTS:

Add a doctor with required information only: PASS

```
curl --data "firstName=Dean&lastName=Wilson&employer=Legacy Emanuel
Hospital&contact=wilsonm@legacy.com" -H "Accept: application/json"
http://bamesberger-assignment-3.appspot.com/doctor
```

Result:
```
{"firstName": "Dean", "specialty": [], "key": 5785905063264256, "certifications": [], "lastName": "Wilson",
"contact": "wilsonm@legacy.com", "employer": "Legacy Emanuel Hospital"}
```

Add a doctor without all required information: PASS

```
curl --data "firstName=Joann&employer=Providence Health" -H "Accept: application/json"
http://bamesberger-assignment-3.appspot.com/doctor
```

Result: Information not added to databasee (though no error message comes up)

To display the keys of all Doctors in the database: PASS

```
curl -H "Accept: application/json" http://bamesberger-assignment-3.appspot.com/doctor
```

Result:
```
{"keys": [5785905063264256]}
```

To add a Doctor with all fields of data to the database: PASS

```
curl --data "firstName=Miranda&lastName=White&employer=Oregon Health And Sciences
University&contact=mirandawhite@ohsu.com&certifications=NOAB Board Certified&specialty=cardiology,
pediatrics, oncology" -H "Accept: application/json" http://bamesberger-assignment-3.appspot.com/doctor
```

Result:
```
{"firstName": "Miranda", "specialty": ["cardiology, pediatrics, oncology"], "key": 5222955109842944,
"certifications": ["NOAB Board Certified"], "lastName": "White", "contact": "mirandawhite@ohsu.com", "employer":
"Oregon Health And Sciences University"}
```

To delete a Doctor from the database: PASS

```
curl -X DELETE -H "Accept: application/json"
http://bamesberger-assignment-3.appspot.com/doctor/delete/5222955109842944
```

Result :
```
curl -H "Accept: application/json" http://bamesberger-assignment-3.appspot.com/doctor
{"keys": [5785905063264256]}
```

To add a Doctor key to a Patient entity: PASS

```
curl -X PUT -H "Accept: application/json" -d "key=5118776383111168&doctors=5785905063264256"
http://bamesberger-assignment-3.appspot.com/patient/update
```

Result:

```
{"firstName": "Craig", "doctors": [5785905063264256], "roomNo": null, "lastVisit": null, "medications": [],
"condition": null, "allergies": [], "checkedIn": false, "age": 25, "key": 5118776383111168, "gender": "male",
"lastName": "Brown", "diseaseHistory": []}
```

Key is deleted from Patient entity when Doctor is deleted: FAIL

```
{"diseaseHistory": [], "allergies": [], "gender": "male", "lastVisit": null, "key": 5118776383111168,
"condition": null, "checkedIn": false, "doctors": [5785905063264256], "age": 25, "lastName": "Brown",
"medications": [], "firstName": "Craig", "roomNo": null}
```