ModelKB DB - Project Increment 2

Team Members

Brett Recker - Class ID: 22

Redwan Al Badawi - Class ID: 1

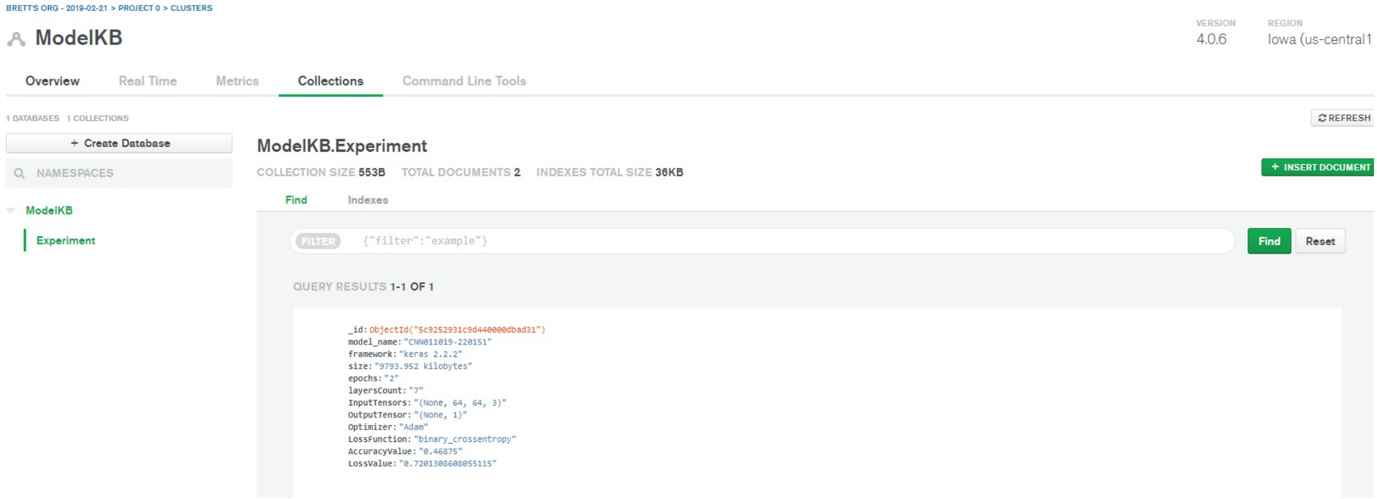
Increment Goals and Objectives

For this increment our main objectives were to upload the experiment artifacts into MongoDB. I was having difficulty figuring out how to select multiple files, parse through the flat file to post as name/value pairs, along with the pictures and metadata. Therefore, I implemented a REST API to post and get textual form data. This allowed me to become more comfortable with REST API's to build on for project increment 3.

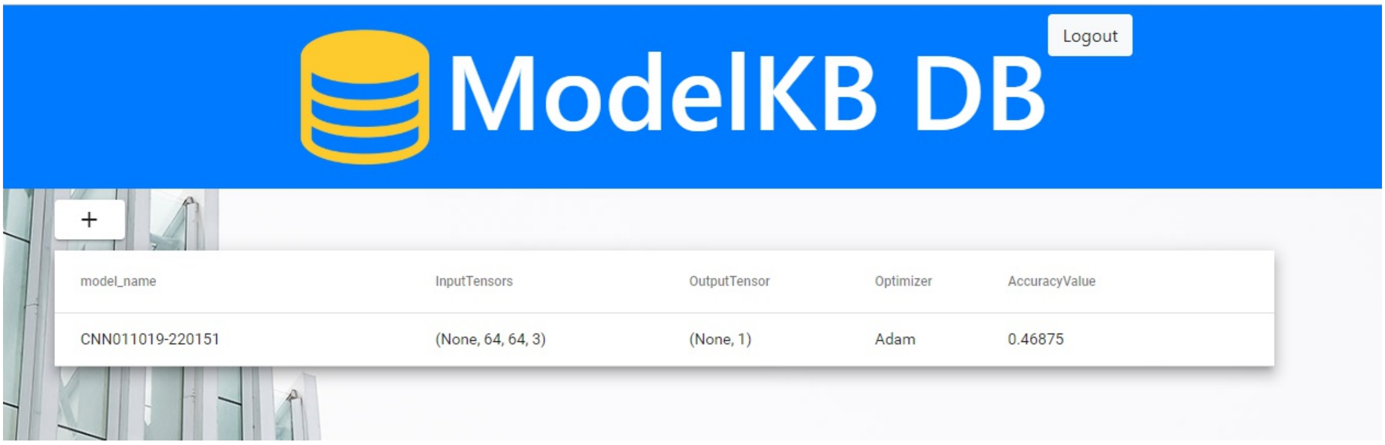
I did, however, add to the authentication service to ensure guards were placed on the other routes as well as adding a logout button to the navigation bar.

MODELDB Component

The MODELDB component loads all experiments which are currently stored in MongoDB for viewing. Moving forward I would like to add filtering so a user can view experiments based off the model_name, framework, or other criteria.



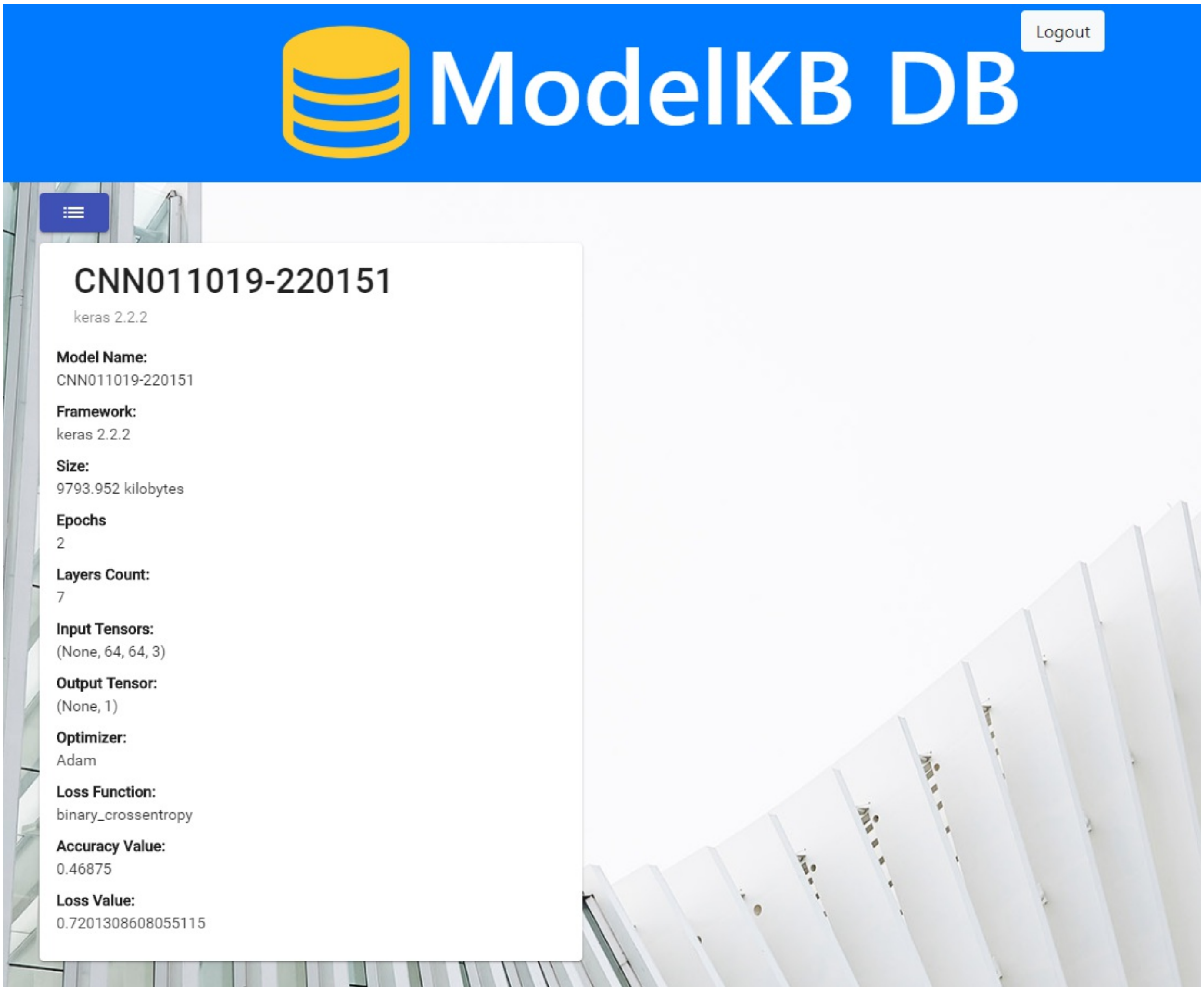
```
_id: ObjectId("5c9252931c9d440000dbad31")
model_name: "CNN011019-220151"
framework: "keras 2.2.2"
size: "9793.952 kilobytes"
epochs: "2"
layersCount: "7"
InputTensors: "(None, 64, 64, 3)"
OutputTensor: "(None, 1)"
Optimizer: "Adam"
LossFunction: "binary_crossentropy"
AccuracyValue: "0.46875"
LossValue: "0.7201308608055115"
```



model_name	InputTensors	OutputTensor	Optimizer	AccuracyValue
CNN011019-220151	(None, 64, 64, 3)	(None, 1)	Adam	0.46875

MODELDB-DETAIL Component

The MODELDB-DETAIL component is accessed by clicking on an experiment from the MODELDB page. This will load a form which contains all data for that experiment. Next, I want to be able to post the pictures related to the experiment so they can be viewed here as well as an option to download the metadata file.



CNN011019-220151

keras 2.2.2

Model Name: CNN011019-220151

Framework: keras 2.2.2

Size: 9793.952 kilobytes

Epochs 2

Layers Count: 7

Input Tensors: (None, 64, 64, 3)

Output Tensor: (None, 1)

Optimizer: Adam

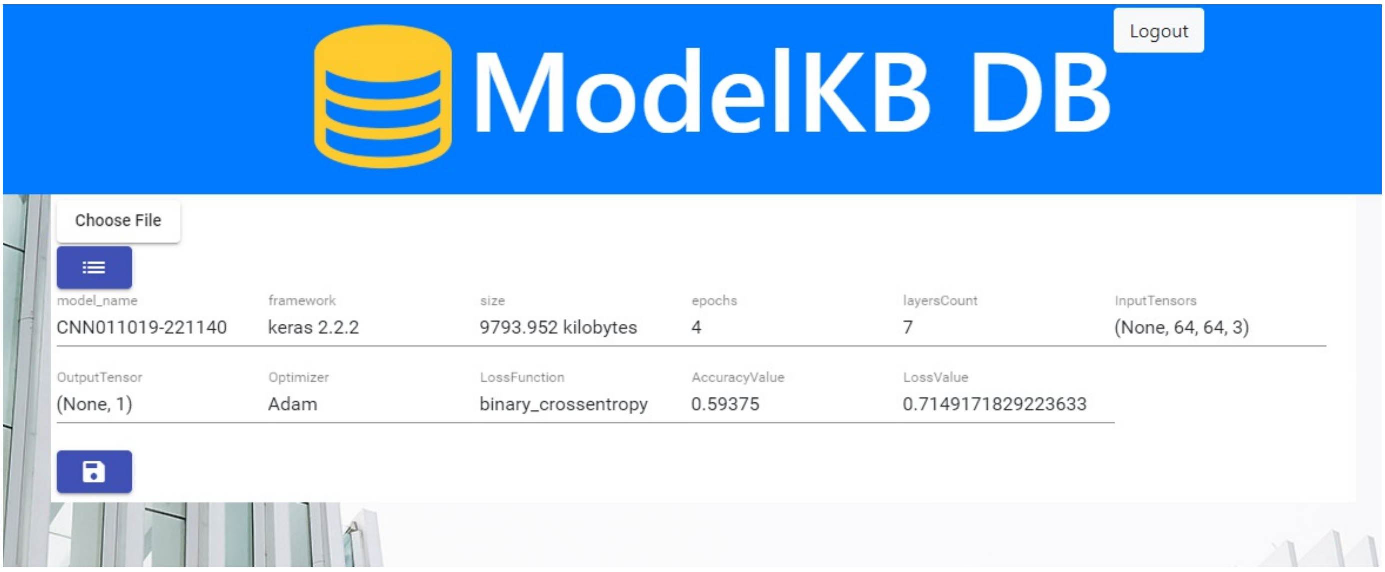
Loss Function: binary_crossentropy

Accuracy Value: 0.46875

Loss Value: 0.7201308608055115

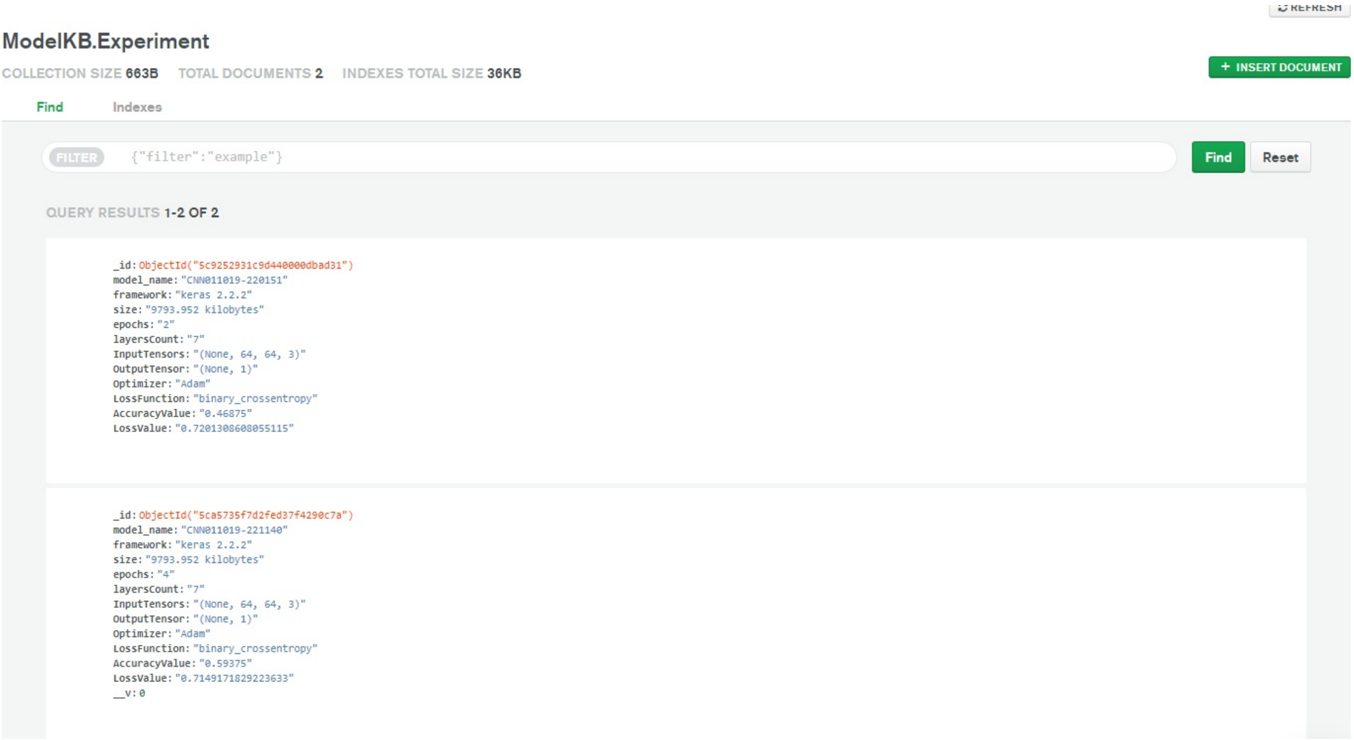
MODELDB-UPLOAD Component

The MODELDB-UPLOAD Component contains an input form for the user to post experiment details to MongoDB. It also has a button to choose multiple files for upload. Currently, this button is not functioning but moving forward I want to be able to select all experiment files and process accordingly. The .h5 will be parsed and automatically fill out the details form for viewing by the user. Then, the pictures will be uploaded to MongoDB and seen on this page. Lastly, the metadata file will post to MongoDB and notify the user when successfully uploaded.



model_name	framework	size	epochs	layersCount	InputTensors
CNN011019-221140	keras 2.2.2	9793.952 kilobytes	4	7	(None, 64, 64, 3)

OutputTensor	Optimizer	LossFunction	AccuracyValue	LossValue
(None, 1)	Adam	binary_crossentropy	0.59375	0.7149171829223633



ModelKB.Experiment

COLLECTION SIZE 663B   TOTAL DOCUMENTS 2   INDEXES TOTAL SIZE 36KB

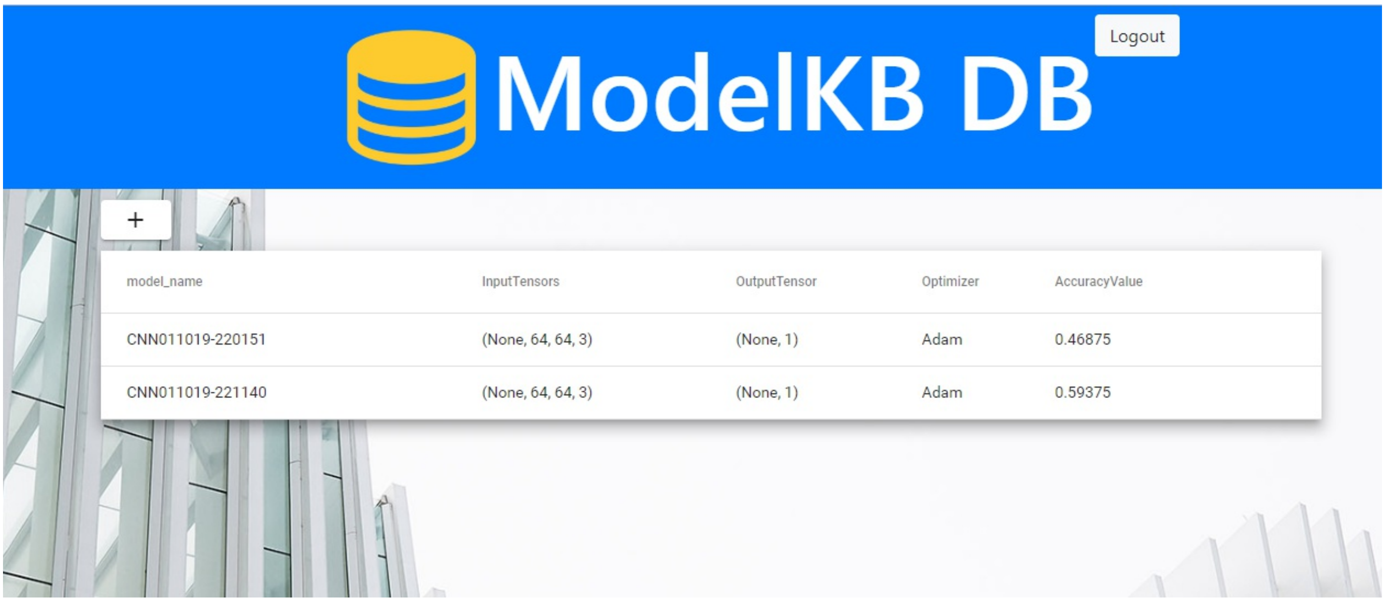
{"filter":"example"}

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("5c9252931c9d440000dbad31")
model_name: "CNN011019-220151"
framework: "keras 2.2.2"
size: "9793.952 kilobytes"
epochs: "2"
layersCount: "7"
InputTensors: "(None, 64, 64, 3)"
OutputTensor: "(None, 1)"
Optimizer: "Adam"
LossFunction: "binary_crossentropy"
AccuracyValue: "0.46875"
LossValue: "0.7201308608055115"
```

```
_id: ObjectId("5ca5735f7d2fed37f4290c7a")
model_name: "CNN011019-221140"
framework: "keras 2.2.2"
size: "9793.952 kilobytes"
epochs: "4"
layersCount: "7"
InputTensors: "(None, 64, 64, 3)"
OutputTensor: "(None, 1)"
Optimizer: "Adam"
LossFunction: "binary_crossentropy"
AccuracyValue: "0.59375"
LossValue: "0.7149171829223633"
__v: 0
```

This newly created experiment can then be seen on the MODELDB-DETAIL component:



model_name	InputTensors	OutputTensor	Optimizer	AccuracyValue
CNN011019-220151	(None, 64, 64, 3)	(None, 1)	Adam	0.46875
CNN011019-221140	(None, 64, 64, 3)	(None, 1)	Adam	0.59375

Outcome

This project still has a long ways to go and I hope to be able to implement all desired functionality. Thus far I have implement registration through Firebase and a REST API with MongoDB. From my research, it appears I will need to implement Multer or GridFS to be able to upload files to the database. This will be depenedent on the artifact file’s size as GridFS supports file with a maximum size of 16MB.