jupyter nbconvert --execute --to html MicrosoftPrediction.ipynb --ExecutePreprocessor.timeout=-1

Above is the command needed to convert this to an html file. Set the timeout to -1 so the converter doesn't end the process early.

The goal of this analysis is to familiarize myself with time-series analysis and to run a sample simulation at the end to see how the model's accuracy would play out in the real world, assuming I buy as sell a single stock at a time, buying at the opening price for a day and selling at the closing price.

Note: I read a previous approach piror to my analysis at

https://towardsdatascience.com/predicting-stock-price-with-lstm-13af86a74944 (https://towardsdatascience.com/predicting-stock-price-with-lstm-13af86a74944)

by Asutosh Nayak

I improved upon the ideas here to suit my own understanding of the problem and to simplify a lot of preprocessing required for the LSTM input. The architecture used for the LSTM was my own choice, as well as running a simulation.

In [1]:
```
from yahoo_fin import stock_info as si
import tensorflow as tf
from tensorflow import keras
from time import sleep
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.preprocessing import MinMaxScaler
import numpy as np
```

Use the last 2 months of data to predict new day value so 'TIME_STEPS' = 60 days. Since the predictions are a day in advance 'DAYS_TO_PREDICT' = 1.

In [2]:
```
TIME_STEPS = 60
DAYS_TO_PREDICT = 1

# Use scalers to normalize values between 0 and 1
dataScaler = MinMaxScaler(feature_range=(0, 1))
outputScaler = MinMaxScaler(feature_range=(0, 1))

def buildTimeSeries(start_date, end_date):
    msft = si.get_data('MSFT', start_date=start_date, end_date=end_date)

    # Normalize values for training
    msftScaled = dataScaler.fit_transform(
        msft[['open', 'close', 'high', 'low', 'adjclose', 'volume']].values)
    outputScaled = outputScaler.fit_transform(msft[['close']].values)

    # Keep track of each day's input (data) and output (labels) values to train and test predictiv
e model
    data = []
    labels = []

    for i in range(TIME_STEPS, len(msft)-DAYS_TO_PREDICT):

        # Get dataframe of info from 60 days prior
        oldPrices = msftScaled[i-TIME_STEPS:i, :]
        data.append(oldPrices)
        # Get closing price for day after the oldPrices
        desiredOutput = outputScaled[i: i + DAYS_TO_PREDICT]
        labels.append(desiredOutput)

    data = np.array(data)
    labels = np.array(labels)
    labels = np.squeeze(labels)

    return data, labels
```

In [3]:
```
# Train on 13 years worth of data and test on 2018-2019 prices
train_start='01/01/2005'
train_end='12/31/2017'
xTrain, yTrain = buildTimeSeries(train_start, train_end)

test_start='01/01/2018'
test_end='05/31/2019'
xTest, yTest = buildTimeSeries(test_start, test_end)
```

In [4]:
```
# Ensure the shapes are correct for NN input/output
print(xTrain.shape)
print(yTrain.shape)

print(xTest.shape)
print(yTest.shape)
```

```
(3211, 60, 6)
(3211,)
(293, 60, 6)
(293,)
```

LSTM models are particularly useful when doing time series analysis because of the RNN architecture which allows a kind of memory to be stored of previous outputs. Although the prediction is only one day in advance, this general architecture should be able to be used for forecasting multiple days in advance (albeit with less accuracy).

tensorboard --logdir=./logs

The above terminal command can be used for tensorflow visualization at

In [5]:
```
keras.backend.clear_session()

lstm_model = keras.models.Sequential()
lstm_model.add(keras.layers.Conv1D(32, kernel_size=3, input_shape=(TIME_STEPS, 6)))
lstm_model.add(keras.layers.MaxPooling1D(4))
lstm_model.add(keras.layers.LSTM(32, return_sequences=True))
lstm_model.add(keras.layers.LSTM(32, return_sequences=False))
lstm_model.add(tf.keras.layers.Dense(DAYS_TO_PREDICT))

lstm_model.summary()

optimizer = keras.optimizers.RMSprop(lr=0.0001)
lstm_model.compile(loss='mse', optimizer=optimizer)

# Tensorboard is primarily used for visualization of the model here
# tensorboard = keras.callbacks.TensorBoard(log_dir="./logs".format(time()))

lstm_model.fit(xTrain, yTrain, epochs=100, validation_split=0.2)
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1d (Conv1D)              (None, 58, 32)            608       
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 14, 32)            0         
_________________________________________________________________
lstm (LSTM)                  (None, 14, 32)            8320      
_________________________________________________________________
lstm_1 (LSTM)                (None, 32)                8320      
_________________________________________________________________
dense (Dense)                (None, 1)                 33        
=================================================================
Total params: 17,281
Trainable params: 17,281
Non-trainable params: 0
_________________________________________________________________
Train on 2568 samples, validate on 643 samples
Epoch 1/100
2568/2568 [==============================] - 2s 826us/step - loss: 0.0035 - val_loss: 0.0086
Epoch 2/100
2568/2568 [==============================] - 1s 338us/step - loss: 6.0008e-04 - val_loss: 0.0063
Epoch 3/100
2568/2568 [==============================] - 1s 331us/step - loss: 5.5964e-04 - val_loss: 0.0047
Epoch 4/100
2568/2568 [==============================] - 1s 332us/step - loss: 5.1786e-04 - val_loss: 0.0058
Epoch 5/100
2568/2568 [==============================] - 1s 331us/step - loss: 4.8550e-04 - val_loss: 0.0043
Epoch 6/100
2568/2568 [==============================] - 1s 329us/step - loss: 4.5621e-04 - val_loss: 0.0042
Epoch 7/100
2568/2568 [==============================] - 1s 335us/step - loss: 4.2451e-04 - val_loss: 0.0029
Epoch 8/100
2568/2568 [==============================] - 1s 328us/step - loss: 4.0983e-04 - val_loss: 0.0040
Epoch 9/100
2568/2568 [==============================] - 1s 333us/step - loss: 4.0144e-04 - val_loss: 0.0059
Epoch 10/100
2568/2568 [==============================] - 1s 331us/step - loss: 3.7961e-04 - val_loss: 0.0052
Epoch 11/100
2568/2568 [==============================] - 1s 331us/step - loss: 3.7349e-04 - val_loss: 0.0021
Epoch 12/100
2568/2568 [==============================] - 1s 333us/step - loss: 3.7002e-04 - val_loss: 0.0025
Epoch 13/100
2568/2568 [==============================] - 1s 332us/step - loss: 3.5830e-04 - val_loss: 0.0015
Epoch 14/100
2568/2568 [==============================] - 1s 330us/step - loss: 3.5298e-04 - val_loss: 0.0017
Epoch 15/100
2568/2568 [==============================] - 1s 331us/step - loss: 3.4327e-04 - val_loss: 0.0012
Epoch 16/100
2568/2568 [==============================] - 1s 328us/step - loss: 3.4660e-04 - val_loss: 0.0021
Epoch 17/100
2568/2568 [==============================] - 1s 336us/step - loss: 3.3840e-04 - val_loss: 0.0017
Epoch 18/100
2568/2568 [==============================] - 1s 343us/step - loss: 3.3015e-04 - val_loss: 0.0023
Epoch 19/100
2568/2568 [==============================] - 1s 336us/step - loss: 3.2738e-04 - val_loss: 0.0012
Epoch 20/100
2568/2568 [==============================] - 1s 332us/step - loss: 3.3057e-04 - val_loss: 0.0018
Epoch 21/100
2568/2568 [==============================] - 1s 327us/step - loss: 3.1833e-04 - val_loss: 0.0018
Epoch 22/100
2568/2568 [==============================] - 1s 347us/step - loss: 3.1911e-04 - val_loss: 0.0014
Epoch 23/100
2568/2568 [==============================] - 1s 330us/step - loss: 3.1321e-04 - val_loss: 0.0016
Epoch 24/100
2568/2568 [==============================] - 1s 333us/step - loss: 3.1659e-04 - val_loss: 0.0011
Epoch 25/100
2568/2568 [==============================] - 1s 331us/step - loss: 3.0984e-04 - val_loss: 0.0023
Epoch 26/100
2568/2568 [==============================] - 1s 344us/step - loss: 3.0762e-04 - val_loss: 0.0014
Epoch 27/100
2568/2568 [==============================] - 1s 346us/step - loss: 3.1251e-04 - val_loss: 0.0025
Epoch 28/100
2568/2568 [==============================] - 1s 340us/step - loss: 2.9871e-04 - val_loss: 0.0034
Epoch 29/100
2568/2568 [==============================] - 1s 329us/step - loss: 2.9592e-04 - val_loss: 0.0019
```

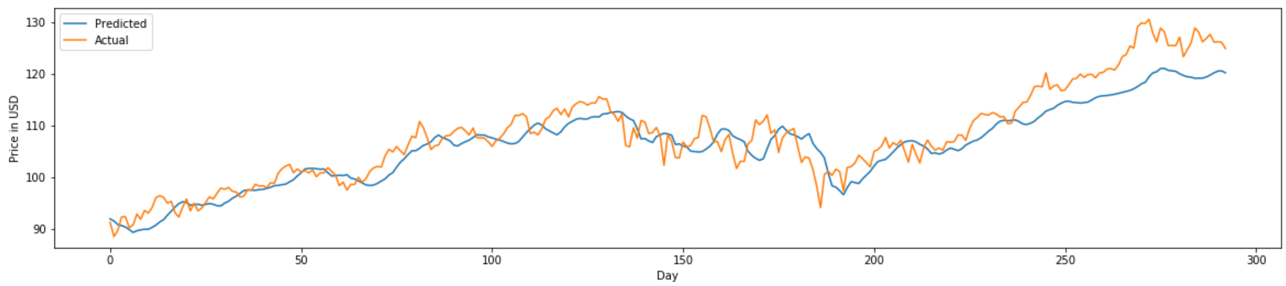
```
Epoch 30/100
2568/2568 [==============================] - 1s 332us/step - loss: 2.9778e-04 - val_loss: 0.0019
Epoch 31/100
2568/2568 [==============================] - 1s 332us/step - loss: 2.9097e-04 - val_loss: 0.0015
Epoch 32/100
2568/2568 [==============================] - 1s 345us/step - loss: 2.9229e-04 - val_loss: 0.0021
Epoch 33/100
2568/2568 [==============================] - 1s 331us/step - loss: 2.8665e-04 - val_loss: 0.0017
Epoch 34/100
2568/2568 [==============================] - 1s 331us/step - loss: 2.8193e-04 - val_loss: 0.0049
Epoch 35/100
2568/2568 [==============================] - 1s 328us/step - loss: 2.8147e-04 - val_loss: 0.0023
Epoch 36/100
2568/2568 [==============================] - 1s 333us/step - loss: 2.7800e-04 - val_loss: 0.0026
Epoch 37/100
2568/2568 [==============================] - 1s 329us/step - loss: 2.7514e-04 - val_loss: 0.0018
Epoch 38/100
2568/2568 [==============================] - 1s 332us/step - loss: 2.7238e-04 - val_loss: 0.0019
Epoch 39/100
2568/2568 [==============================] - 1s 329us/step - loss: 2.6675e-04 - val_loss: 0.0016
Epoch 40/100
2568/2568 [==============================] - 1s 330us/step - loss: 2.7120e-04 - val_loss: 0.0033
Epoch 41/100
2568/2568 [==============================] - 1s 329us/step - loss: 2.6537e-04 - val_loss: 0.0035
Epoch 42/100
2568/2568 [==============================] - 1s 328us/step - loss: 2.6562e-04 - val_loss: 0.0021
Epoch 43/100
2568/2568 [==============================] - 1s 329us/step - loss: 2.6239e-04 - val_loss: 0.0014
Epoch 44/100
2568/2568 [==============================] - 1s 333us/step - loss: 2.5843e-04 - val_loss: 0.0031
Epoch 45/100
2568/2568 [==============================] - 1s 327us/step - loss: 2.6136e-04 - val_loss: 0.0024
Epoch 46/100
2568/2568 [==============================] - 1s 334us/step - loss: 2.4979e-04 - val_loss: 0.0028
Epoch 47/100
2568/2568 [==============================] - 1s 331us/step - loss: 2.5317e-04 - val_loss: 0.0027
Epoch 48/100
 736/2568 [=======>......................] - ETA: 0s - loss: 2.4893e-0
4
```

NOTE/ REMINDER FOR LATER: LSTM output is of dimensions/shape (X,) where X is number of test cases, but to inverse transform using the scaler, you need an output shape of (X, 1). Although both represent a linear array, the first one (X,) is considered a one-dimensional numpy array and the second one (X, 1) is considered a two-dimensional numpy array. Thus, using reshape(-1, 1) converts (X,) to (X, 1). Also note that np.squeeze removes all dimensions where dimension = 1. So for example, (X, 1) becomes (X,). In this case, the lstm prediction output is the same as yPred.

In [6]:
```
# Changes values from between 0 and 1 back to USD values for plotting
prediction = outputScaler.inverse_transform(lstm_model.predict(xTest).reshape(-1, 1))
yPred = np.squeeze(prediction)
test = np.squeeze(yTest)
actual = outputScaler.inverse_transform(test.reshape(-1, 1))
actual = np.squeeze(actual)
```

In [7]:
```
fig = plt.figure(figsize=(20, 4))
ax = fig.add_subplot(111)
fig.patch.set_facecolor('white')
ax.plot(yPred, label='Predicted')
ax.plot(actual, label='Actual')
ax.legend()
ax.set_xlabel('Day')
ax.set_ylabel('Price in USD')
```

```
Text(0,0.5,'Price in USD')
```



In [8]:
```
# Run the simulation using test data, assume closing price for a day equals opening price for next
day

balance = 0
balanceValues = []

for index, openingVal in enumerate(actual[ :-1]):
    day = index
    closingPredicted = yPred[index + 1]
    closingActual = actual[index + 1]

    # If prediction is larger than the opening value, then BUY
    if closingPredicted > openingVal:
        balance += closingActual - openingVal

    # If prediction is less than the opening value, do NOTHING
    else:
        pass
    balanceValues.append(balance)
    sleep(.1)
    print('Day:', day, 'Opening Price:', round(openingVal, 3), 'Predicted Closing Price:', round(c
losingPredicted, 3),
         'Actual Closing Price:', round(closingActual, 3), 'Balance', round(balance, 3))

print('Final Balance:', balance)
```

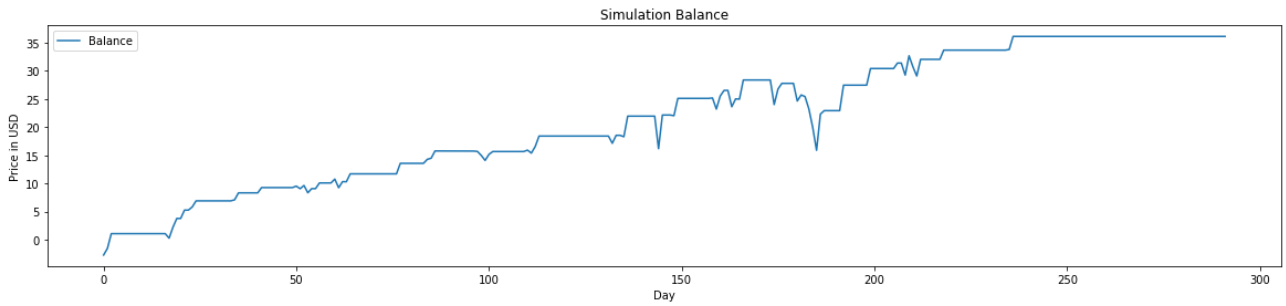
```
Day: 0 Opening Price: 91.27 Predicted Closing Price: 91.54 Actual Closing Price: 88.52 Balance -2.75
Day: 1 Opening Price: 88.52 Predicted Closing Price: 90.868 Actual Closing Price: 89.71 Balance -1.56
Day: 2 Opening Price: 89.71 Predicted Closing Price: 90.655 Actual Closing Price: 92.33 Balance 1.06
Day: 3 Opening Price: 92.33 Predicted Closing Price: 90.378 Actual Closing Price: 92.38 Balance 1.06
Day: 4 Opening Price: 92.38 Predicted Closing Price: 89.88 Actual Closing Price: 90.23 Balance 1.06
Day: 5 Opening Price: 90.23 Predicted Closing Price: 89.334 Actual Closing Price: 90.77 Balance 1.06
Day: 6 Opening Price: 90.77 Predicted Closing Price: 89.654 Actual Closing Price: 92.88 Balance 1.06
Day: 7 Opening Price: 92.88 Predicted Closing Price: 89.835 Actual Closing Price: 91.86 Balance 1.06
Day: 8 Opening Price: 91.86 Predicted Closing Price: 89.953 Actual Closing Price: 93.58 Balance 1.06
Day: 9 Opening Price: 93.58 Predicted Closing Price: 89.937 Actual Closing Price: 93.08 Balance 1.06
Day: 10 Opening Price: 93.08 Predicted Closing Price: 90.313 Actual Closing Price: 94.17 Balance 1.06
Day: 11 Opening Price: 94.17 Predicted Closing Price: 90.759 Actual Closing Price: 96.07 Balance 1.06
Day: 12 Opening Price: 96.07 Predicted Closing Price: 91.364 Actual Closing Price: 96.44 Balance 1.06
Day: 13 Opening Price: 96.44 Predicted Closing Price: 91.808 Actual Closing Price: 96.11 Balance 1.06
Day: 14 Opening Price: 96.11 Predicted Closing Price: 92.677 Actual Closing Price: 95.0 Balance 1.06
Day: 15 Opening Price: 95.0 Predicted Closing Price: 93.45 Actual Closing Price: 95.35 Balance 1.06
Day: 16 Opening Price: 95.35 Predicted Closing Price: 94.257 Actual Closing Price: 93.12 Balance 1.06
Day: 17 Opening Price: 93.12 Predicted Closing Price: 94.907 Actual Closing Price: 92.31 Balance 0.25
Day: 18 Opening Price: 92.31 Predicted Closing Price: 95.265 Actual Closing Price: 94.26 Balance 2.2
Day: 19 Opening Price: 94.26 Predicted Closing Price: 95.128 Actual Closing Price: 95.82 Balance 3.76
Day: 20 Opening Price: 95.82 Predicted Closing Price: 94.586 Actual Closing Price: 93.52 Balance 3.76
Day: 21 Opening Price: 93.52 Predicted Closing Price: 94.703 Actual Closing Price: 95.0 Balance 5.24
Day: 22 Opening Price: 95.0 Predicted Closing Price: 94.796 Actual Closing Price: 93.51 Balance 5.24
Day: 23 Opening Price: 93.51 Predicted Closing Price: 94.582 Actual Closing Price: 94.07 Balance 5.8
Day: 24 Opening Price: 94.07 Predicted Closing Price: 94.766 Actual Closing Price: 95.16 Balance 6.89
Day: 25 Opening Price: 95.16 Predicted Closing Price: 94.928 Actual Closing Price: 96.22 Balance 6.89
Day: 26 Opening Price: 96.22 Predicted Closing Price: 94.77 Actual Closing Price: 95.81 Balance 6.89
Day: 27 Opening Price: 95.81 Predicted Closing Price: 94.504 Actual Closing Price: 96.94 Balance 6.89
Day: 28 Opening Price: 96.94 Predicted Closing Price: 94.473 Actual Closing Price: 97.91 Balance 6.89
Day: 29 Opening Price: 97.91 Predicted Closing Price: 95.01 Actual Closing Price: 97.7 Balance 6.89
Day: 30 Opening Price: 97.7 Predicted Closing Price: 95.364 Actual Closing Price: 98.03 Balance 6.89
Day: 31 Opening Price: 98.03 Predicted Closing Price: 95.953 Actual Closing Price: 97.32 Balance 6.89
Day: 32 Opening Price: 97.32 Predicted Closing Price: 96.349 Actual Closing Price: 97.15 Balance 6.89
Day: 33 Opening Price: 97.15 Predicted Closing Price: 96.885 Actual Closing Price: 96.18 Balance 6.89
Day: 34 Opening Price: 96.18 Predicted Closing Price: 97.432 Actual Closing Price: 96.36 Balance 7.07
Day: 35 Opening Price: 96.36 Predicted Closing Price: 97.546 Actual Closing Price: 97.6 Balance 8.31
Day: 36 Opening Price: 97.6 Predicted Closing Price: 97.538 Actual Closing Price: 97.5 Balance 8.31
Day: 37 Opening Price: 97.5 Predicted Closing Price: 97.471 Actual Closing Price: 98.66 Balance 8.31
Day: 38 Opening Price: 98.66 Predicted Closing Price: 97.625 Actual Closing Price: 98.31 Balance 8.31
Day: 39 Opening Price: 98.31 Predicted Closing Price: 97.653 Actual Closing Price: 98.36 Balance 8.31
Day: 40 Opening Price: 98.36 Predicted Closing Price: 97.87 Actual Closing Price: 98.01 Balance 8.31
Day: 41 Opening Price: 98.01 Predicted Closing Price: 98.064 Actual Closing Price: 98.95 Balance 9.25
Day: 42 Opening Price: 98.95 Predicted Closing Price: 98.391 Actual Closing Price: 98.84 Balance 9.25
Day: 43 Opening Price: 98.84 Predicted Closing Price: 98.436 Actual Closing Price: 100.79 Balance 9.25
Day: 44 Opening Price: 100.79 Predicted Closing Price: 98.586 Actual Closing Price: 101.67 Balance 9.25
Day: 45 Opening Price: 101.67 Predicted Closing Price: 98.682 Actual Closing Price: 102.19 Balance 9.25
Day: 46 Opening Price: 102.19 Predicted Closing Price: 99.142 Actual Closing Price: 102.49 Balance 9.25
Day: 47 Opening Price: 102.49 Predicted Closing Price: 99.535 Actual Closing Price: 100.88 Balance 9.25
Day: 48 Opening Price: 100.88 Predicted Closing Price: 100.263 Actual Closing Price: 101.63 Balance 9.25
Day: 49 Opening Price: 101.63 Predicted Closing Price: 100.885 Actual Closing Price: 101.05 Balance 9.25
Day: 50 Opening Price: 101.05 Predicted Closing Price: 101.61 Actual Closing Price: 101.31 Balance 9.51
Day: 51 Opening Price: 101.31 Predicted Closing Price: 101.724 Actual Closing Price: 100.85 Balance 9.05
Day: 52 Opening Price: 100.85 Predicted Closing Price: 101.742 Actual Closing Price: 101.42 Balance 9.62
Day: 53 Opening Price: 101.42 Predicted Closing Price: 101.649 Actual Closing Price: 100.13 Balance 8.33
Day: 54 Opening Price: 100.13 Predicted Closing Price: 101.564 Actual Closing Price: 100.86 Balance 9.06
Day: 55 Opening Price: 100.86 Predicted Closing Price: 101.608 Actual Closing Price: 100.86 Balance 9.06
Day: 56 Opening Price: 100.86 Predicted Closing Price: 100.881 Actual Closing Price: 101.87 Balance 10.07
Day: 57 Opening Price: 101.87 Predicted Closing Price: 100.263 Actual Closing Price: 101.14 Balance 10.07
Day: 58 Opening Price: 101.14 Predicted Closing Price: 100.337 Actual Closing Price: 100.41 Balance 10.07
Day: 59 Opening Price: 100.41 Predicted Closing Price: 100.4 Actual Closing Price: 98.39 Balance 10.07
Day: 60 Opening Price: 98.39 Predicted Closing Price: 100.336 Actual Closing Price: 99.08 Balance 10.76
Day: 61 Opening Price: 99.08 Predicted Closing Price: 100.51 Actual Closing Price: 97.54 Balance 9.22
Day: 62 Opening Price: 97.54 Predicted Closing Price: 99.861 Actual Closing Price: 98.63 Balance 10.31
Day: 63 Opening Price: 98.63 Predicted Closing Price: 99.672 Actual Closing Price: 98.61 Balance 10.29
Day: 64 Opening Price: 98.61 Predicted Closing Price: 99.306 Actual Closing Price: 100.01 Balance 11.69
Day: 65 Opening Price: 100.01 Predicted Closing Price: 98.916 Actual Closing Price: 99.05 Balance 11.69
Day: 66 Opening Price: 99.05 Predicted Closing Price: 98.514 Actual Closing Price: 99.76 Balance 11.69
Day: 67 Opening Price: 99.76 Predicted Closing Price: 98.427 Actual Closing Price: 101.16 Balance 11.69
Day: 68 Opening Price: 101.16 Predicted Closing Price: 98.507 Actual Closing Price: 101.85 Balance 11.69
Day: 69 Opening Price: 101.85 Predicted Closing Price: 98.833 Actual Closing Price: 102.12 Balance 11.69
Day: 70 Opening Price: 102.12 Predicted Closing Price: 99.269 Actual Closing Price: 101.98 Balance 11.69
Day: 71 Opening Price: 101.98 Predicted Closing Price: 99.72 Actual Closing Price: 104.19 Balance 11.69
Day: 72 Opening Price: 104.19 Predicted Closing Price: 100.455 Actual Closing Price: 105.43 Balance 11.69
Day: 73 Opening Price: 105.43 Predicted Closing Price: 100.902 Actual Closing Price: 104.91 Balance 11.69
Day: 74 Opening Price: 104.91 Predicted Closing Price: 101.97 Actual Closing Price: 105.95 Balance 11.69
Day: 75 Opening Price: 105.95 Predicted Closing Price: 102.927 Actual Closing Price: 105.12 Balance 11.69
```

```
Day: 76 Opening Price: 105.12 Predicted Closing Price: 103.589 Actual Closing Price: 104.4 Balance 11.69
Day: 77 Opening Price: 104.4 Predicted Closing Price: 104.405 Actual Closing Price: 106.27 Balance 13.56
Day: 78 Opening Price: 106.27 Predicted Closing Price: 105.141 Actual Closing Price: 107.97 Balance 13.56
Day: 79 Opening Price: 107.97 Predicted Closing Price: 105.159 Actual Closing Price: 107.66 Balance 13.56
Day: 80 Opening Price: 107.66 Predicted Closing Price: 105.605 Actual Closing Price: 110.83 Balance 13.56
Day: 81 Opening Price: 110.83 Predicted Closing Price: 106.18 Actual Closing Price: 109.62 Balance 13.56
Day: 82 Opening Price: 109.62 Predicted Closing Price: 106.452 Actual Closing Price: 107.68 Balance 13.56
Day: 83 Opening Price: 107.68 Predicted Closing Price: 106.873 Actual Closing Price: 105.37 Balance 13.56
Day: 84 Opening Price: 105.37 Predicted Closing Price: 107.721 Actual Closing Price: 106.08 Balance 14.27
Day: 85 Opening Price: 106.08 Predicted Closing Price: 108.186 Actual Closing Price: 106.28 Balance 14.47
Day: 86 Opening Price: 106.28 Predicted Closing Price: 107.672 Actual Closing Price: 107.57 Balance 15.76
Day: 87 Opening Price: 107.57 Predicted Closing Price: 107.41 Actual Closing Price: 108.04 Balance 15.76
Day: 88 Opening Price: 108.04 Predicted Closing Price: 107.046 Actual Closing Price: 108.13 Balance 15.76
Day: 89 Opening Price: 108.13 Predicted Closing Price: 106.217 Actual Closing Price: 108.88 Balance 15.76
Day: 90 Opening Price: 108.88 Predicted Closing Price: 106.072 Actual Closing Price: 109.49 Balance 15.76
Day: 91 Opening Price: 109.49 Predicted Closing Price: 106.541 Actual Closing Price: 109.67 Balance 15.76
Day: 92 Opening Price: 109.67 Predicted Closing Price: 106.892 Actual Closing Price: 109.0 Balance 15.76
Day: 93 Opening Price: 109.0 Predicted Closing Price: 107.193 Actual Closing Price: 108.21 Balance 15.76
Day: 94 Opening Price: 108.21 Predicted Closing Price: 107.698 Actual Closing Price: 109.56 Balance 15.76
Day: 95 Opening Price: 109.56 Predicted Closing Price: 108.242 Actual Closing Price: 107.66 Balance 15.76
Day: 96 Opening Price: 107.66 Predicted Closing Price: 108.185 Actual Closing Price: 107.64 Balance 15.74
Day: 97 Opening Price: 107.64 Predicted Closing Price: 108.175 Actual Closing Price: 107.58 Balance 15.68
Day: 98 Opening Price: 107.58 Predicted Closing Price: 107.822 Actual Closing Price: 106.87 Balance 14.97
Day: 99 Opening Price: 106.87 Predicted Closing Price: 107.634 Actual Closing Price: 105.98 Balance 14.08
Day: 100 Opening Price: 105.98 Predicted Closing Price: 107.389 Actual Closing Price: 107.06 Balance 15.16
Day: 101 Opening Price: 107.06 Predicted Closing Price: 107.209 Actual Closing Price: 107.56 Balance 15.66
Day: 102 Opening Price: 107.56 Predicted Closing Price: 106.943 Actual Closing Price: 108.4 Balance 15.66
Day: 103 Opening Price: 108.4 Predicted Closing Price: 106.634 Actual Closing Price: 109.6 Balance 15.66
Day: 104 Opening Price: 109.6 Predicted Closing Price: 106.479 Actual Closing Price: 110.26 Balance 15.66
Day: 105 Opening Price: 110.26 Predicted Closing Price: 106.545 Actual Closing Price: 112.02 Balance 15.66
Day: 106 Opening Price: 112.02 Predicted Closing Price: 106.948 Actual Closing Price: 111.95 Balance 15.66
Day: 107 Opening Price: 111.95 Predicted Closing Price: 107.823 Actual Closing Price: 112.33 Balance 15.66
Day: 108 Opening Price: 112.33 Predicted Closing Price: 108.679 Actual Closing Price: 111.71 Balance 15.66
Day: 109 Opening Price: 111.71 Predicted Closing Price: 109.377 Actual Closing Price: 108.49 Balance 15.66
Day: 110 Opening Price: 108.49 Predicted Closing Price: 110.097 Actual Closing Price: 108.74 Balance 15.91
Day: 111 Opening Price: 108.74 Predicted Closing Price: 110.494 Actual Closing Price: 108.21 Balance 15.38
Day: 112 Opening Price: 108.21 Predicted Closing Price: 110.13 Actual Closing Price: 109.38 Balance 16.55
Day: 113 Opening Price: 109.38 Predicted Closing Price: 109.393 Actual Closing Price: 111.24 Balance 18.41
Day: 114 Opening Price: 111.24 Predicted Closing Price: 108.978 Actual Closing Price: 111.71 Balance 18.41
Day: 115 Opening Price: 111.71 Predicted Closing Price: 108.586 Actual Closing Price: 112.91 Balance 18.41
Day: 116 Opening Price: 112.91 Predicted Closing Price: 108.229 Actual Closing Price: 113.37 Balance 18.41
Day: 117 Opening Price: 113.37 Predicted Closing Price: 108.811 Actual Closing Price: 112.14 Balance 18.41
Day: 118 Opening Price: 112.14 Predicted Closing Price: 109.757 Actual Closing Price: 113.21 Balance 18.41
Day: 119 Opening Price: 113.21 Predicted Closing Price: 110.434 Actual Closing Price: 111.7 Balance 18.41
Day: 120 Opening Price: 111.7 Predicted Closing Price: 110.844 Actual Closing Price: 113.57 Balance 18.41
Day: 121 Opening Price: 113.57 Predicted Closing Price: 111.245 Actual Closing Price: 114.26 Balance 18.41
Day: 122 Opening Price: 114.26 Predicted Closing Price: 111.417 Actual Closing Price: 114.67 Balance 18.41
Day: 123 Opening Price: 114.67 Predicted Closing Price: 111.275 Actual Closing Price: 114.45 Balance 18.41
Day: 124 Opening Price: 114.45 Predicted Closing Price: 111.311 Actual Closing Price: 113.98 Balance 18.41
Day: 125 Opening Price: 113.98 Predicted Closing Price: 111.646 Actual Closing Price: 114.41 Balance 18.41
Day: 126 Opening Price: 114.41 Predicted Closing Price: 111.741 Actual Closing Price: 114.37 Balance 18.41
Day: 127 Opening Price: 114.37 Predicted Closing Price: 111.69 Actual Closing Price: 115.61 Balance 18.41
Day: 128 Opening Price: 115.61 Predicted Closing Price: 112.263 Actual Closing Price: 115.15 Balance 18.41
Day: 129 Opening Price: 115.15 Predicted Closing Price: 112.314 Actual Closing Price: 115.17 Balance 18.41
Day: 130 Opening Price: 115.17 Predicted Closing Price: 112.545 Actual Closing Price: 112.79 Balance 18.41
Day: 131 Opening Price: 112.79 Predicted Closing Price: 112.637 Actual Closing Price: 112.13 Balance 18.41
Day: 132 Opening Price: 112.13 Predicted Closing Price: 112.737 Actual Closing Price: 110.85 Balance 17.13
Day: 133 Opening Price: 110.85 Predicted Closing Price: 112.573 Actual Closing Price: 112.26 Balance 18.54
Day: 134 Opening Price: 112.26 Predicted Closing Price: 111.869 Actual Closing Price: 106.16 Balance 18.54
Day: 135 Opening Price: 106.16 Predicted Closing Price: 111.335 Actual Closing Price: 105.91 Balance 18.29
Day: 136 Opening Price: 105.91 Predicted Closing Price: 110.995 Actual Closing Price: 109.57 Balance 21.95
Day: 137 Opening Price: 109.57 Predicted Closing Price: 109.416 Actual Closing Price: 107.6 Balance 21.95
Day: 138 Opening Price: 107.6 Predicted Closing Price: 107.458 Actual Closing Price: 111.0 Balance 21.95
Day: 139 Opening Price: 111.0 Predicted Closing Price: 107.501 Actual Closing Price: 110.71 Balance 21.95
Day: 140 Opening Price: 110.71 Predicted Closing Price: 107.055 Actual Closing Price: 108.5 Balance 21.95
Day: 141 Opening Price: 108.5 Predicted Closing Price: 106.741 Actual Closing Price: 108.66 Balance 21.95
Day: 142 Opening Price: 108.66 Predicted Closing Price: 107.88 Actual Closing Price: 109.63 Balance 21.95
Day: 143 Opening Price: 109.63 Predicted Closing Price: 108.196 Actual Closing Price: 108.1 Balance 21.95
Day: 144 Opening Price: 108.1 Predicted Closing Price: 108.535 Actual Closing Price: 102.32 Balance 16.17
Day: 145 Opening Price: 102.32 Predicted Closing Price: 108.41 Actual Closing Price: 108.3 Balance 22.15
Day: 146 Opening Price: 108.3 Predicted Closing Price: 108.152 Actual Closing Price: 106.96 Balance 22.15
Day: 147 Opening Price: 106.96 Predicted Closing Price: 106.366 Actual Closing Price: 103.85 Balance 22.15
Day: 148 Opening Price: 103.85 Predicted Closing Price: 106.479 Actual Closing Price: 103.73 Balance 22.03
Day: 149 Opening Price: 103.73 Predicted Closing Price:
```

In [9]:
```
# Plot the overall trend of the balance

fig = plt.figure(figsize=(20, 4))
ax = fig.add_subplot(111)
fig.patch.set_facecolor('white')
ax.plot(balanceValues, label='Balance')
ax.legend()
ax.set_xlabel('Day')
ax.set_ylabel('Price in USD')
ax.set_title('Simulation Balance')
```

```
Text(0.5,1,'Simulation Balance')
```



Final thoughts: This model seems to predict the up/down trend pretty well, but there were a few simplifying assumptions made along the way of this analysis which probably influenced the outcome. Also its important to keep in mind that the actual vs predicted graph is expected to similar to each other since the LSTM is only predicting one value at a time, and by picking something similar to the previous 60 values, it will always fall within range of the actual value. The important prediction here is if it can reliably predict if a stock is going up or down the next day, which is what the balance graph shows.

Although deep learning seems to have some potential in modeling relatively stochastic trends, it is not likely the best way to go about investing. Regarding possible ways of improving this model, the addition of more input data could help. Using the information from similar tech stocks to predict Microsoft's trends would probably reduce loss. Additional information regarding news could also potentially included in the model through seniment analysis from different news sources, although this would require a much more sophisticated approach than the one used here.