**Data and Specification Rules**

* + Path to picture file: VARCHAR(255)
    - Up to 255 characters long
  + Size (MB): Unsigned SMALLINT
    - Positive values only
    - 0-65,535
  + Name : VARCHAR(100)
    - Up to 100 characters long
    - No special characters (?@#$!%^&\*() )
  + Resolution
    - Width: Unsigned SMALLINT
      * No negative values
      * 0 – 65,535
    - Length: Unsigned SMALLINT
      * No negative values
      * 0 – 65,535
  + Orientation: ENUM(“Portrait”,”Landscape”,”Unknown”)
  + Manufacturer:ENUM(“Canon”,”Casio”,”Epson”,”Fijifilm”,”GoPro”,”HP”,”Kodak”,”Panasonic”,”Polaroid”,”Nikon”,”Samsung”,”Sony”,”Other”)
  + Model: VARCHAR(26)
    - Up to 26 characters long
  + Focal Length (mm): Unsigned SMALLINT
    - No negative values
    - 0 – 65,535
  + Exposure Time (seconds): DECIMAL(5,2)
    - (precision, scale)- 5 significant digits, 2 digits past decimal point
    - Positive values only
    - 0.00- 999.99
  + Aperture (*f*/): DECIMAL(5,2)
    - (precision, scale)- 5 significant digits, 2 digits past decimal point
    - Positive values only
    - 0.00 – 999.99
  + ISO*:* Unsigned SMALLINT
    - 0 – 65,535
  + Date: DATETIME
    - Any date after 1/1/1900
  + File type: CHAR(3)
    - Up to 3 characters
    - No special characters (?!@#$%^&\*() )
  + User inputted tags: VARCHAR(26)
  + Color values
    - Average Red: Unsigned TINYINT
      * 0 - 255
    - Average Blue: Unsigned TINYINT
      * 0-255
    - Average Green: Unsigned TINYINT
      * 0-255
* Output files: Excel spreadsheets (.XLS)
* Input files: Image files (.JPG, .PNG, .PPM,.GIF)
* Other Specifications:
  + User created tags are to populate TAG entities
  + A listing of TAG entities will be used to populate a drop-down list of premade tags that a user can choose from, or the user will also be given the option to create new tags.
  + All of the above ENUM types will also have an entity associated with them in the database and will be pre-populated at release of the system. Future updates may populate the table with new models, etc.

**Edit Checks**

The following edit checks for valid data values are presented in pseudo code.

**For numeric types:**

bool valid Numeric(input.attribute,attMin,attMax)

*//input.attribute is a placeholder for whichever numeric attribute is being tested. attMin and attMax are the minimum and maximum boundaries for that attribute, respectively*

{

if (input.attribute < attMin || input.attribute > attMax)

{

print(“Invalid image. Please try another”);

return False;

}

else

{

return True;

}

}

**For character types:**

bool validChar(input.attribute,attLength,attNo)

*//attLength is the maximum length that the attribute can have. attNo is an array of invalid characters*

{

if(input.attribute.length > attLength || input.attribute.contains(attNo))

{

Print(“Invalid. Please try again”);

return False;

}

else

{

return True;

}

}

**For enumerations:**

bool validEnum(input.attribute,enumTable)

*//enumTable refers to which enumeration attribute that is being queried.*

{

if(enumTable.queryExists(input.attribute)==False)  
*//the function queryExists(attribute) will run a mySQL query on the actual database table to see if the attribute is a valid entry*

{

Print(“Invalid. Please try again”);

return False;

}

else

{

return True;

}

}

**System Tests**

1. The first way we are going to test our system is input an image and have our system save the values accordingly as we have designed it to do. It will auto populate the values such as ISO, Aperture, Exposure, Shutter Speed, etc… Once our program has finished and saved the photo we will open the photo outside of the program, open the properties, and compare the values to the values we have stored in our database. If the values match our system is archiving the values correctly and is functioning properly.
2. The second test we are going to perform is take 20 shots at a very specific resolution (very low) and import these into the system. We will then run a query and search for that very specific resolution. Once we have the query we will check to see if all 20 pictures were found according to the set parameters.

**Test Data**

