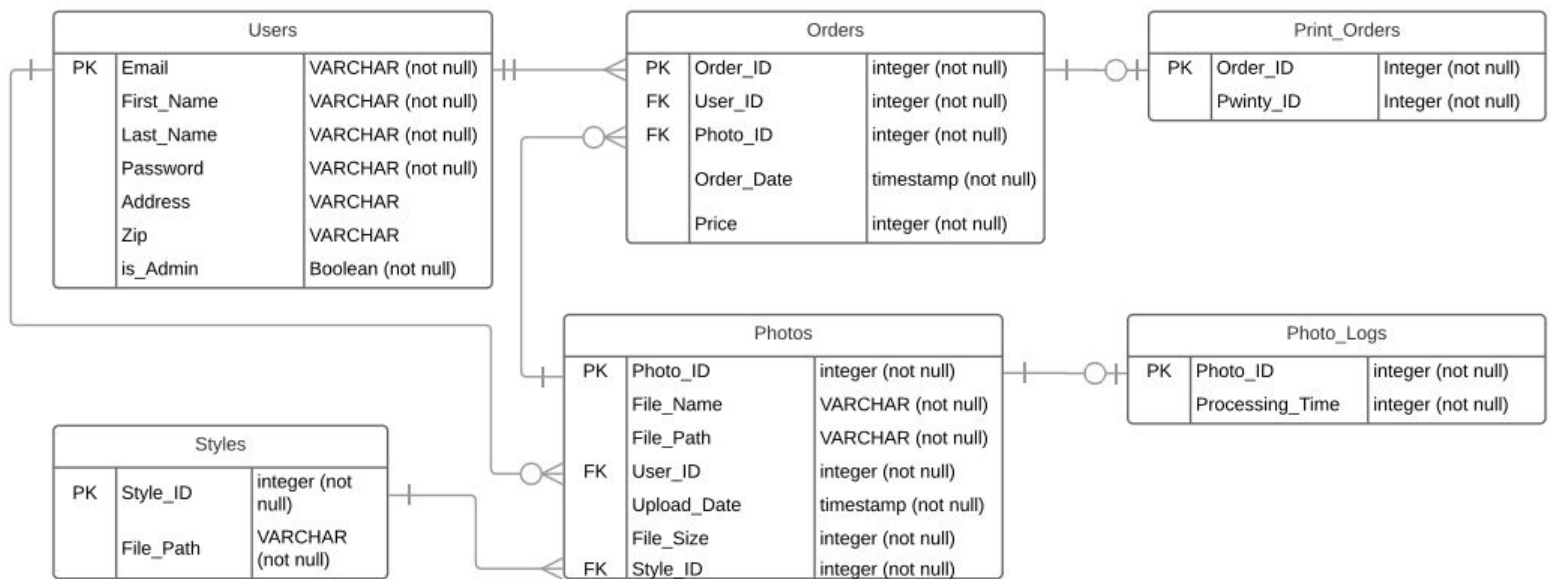


Kevin Kleinschmidt  
 Capping Group 3  
 ASP Project  
 Prof. Rivas  
 9/19/2018

## ASP Project ERD and IT Requirements

### Entity Relationship Diagram:



### Entity Relationship Diagram Documentation:

The database design that our team has put together for our artistic stylizer platform is made up of six tables to hold user account information, photo details, logs, and orders. The Users table includes all user information including email, password, name, the user's mailing address, and administrator status. The user email address serves as the primary key for this table and is referenced as a foreign key in the Photos and Orders tables. Other than the is\_Admin attribute, all of the attributes in the Users table are variable character data types. User passwords will be encrypted for security purposes. The is\_Admin attribute is a boolean value of either true or false,

which determines if a user will be given administrator privileges. All users that have an `is_Admin` value of true will receive administrator privileges. All of the attributes in the Users table cannot have a null value, other than the address and zip code, which can have a null value. The Photos table is used to store information on each content photo that is uploaded onto the system. Photos will not be stored on the database, but instead will be stored on another server. After discussion we felt that storing the photos on the database may bog the system down and cause the database to be slow. When a user uploads a photo, each photo will be given a unique `Photo_ID`. Users will define the file name and the file will be placed on our storage server. We have a `File_Path` attribute that stores the file path where the photo is located on the server. The photos table will also hold the photo's upload date, file size, the user ID tied to the user who uploaded the photo, and the `Style_ID`, which tells us what style was used for the style transfer. None of the attributes in the Photos table can have a null value. `User_ID` is a foreign key referencing the `User_ID` in the Users table and `Style_ID` is a foreign key referencing the `Style_ID` in the Styles table. The relationship between the Users and Photos table is a one to zero or many relationship. A user can upload zero or many photos and a photo can but uploaded by one user. The Photos table shares a relationship with the Photo\_Logs table as well. Photo\_Logs holds the processing time for each photo that is stylized using the API. This table will hold the processing time and the `Photo_ID`. The relationship between Photos and Photo\_Logs is a one to zero or one relationship. Not all photos that are uploaded onto the system will be stylized and processed. Therefore, a photo can have zero or more logs and a log can have one photo. All uploaded content photos that are not stylized and paid for within twenty-four hours of the upload date will be removed from the system. The Orders table stores all information about orders that

have been placed by a user. Each order that is placed will receive an Order\_ID that is unique to that order, making Order\_ID the primary key of this table. The order date and price will also be stored in the Orders table. User\_ID and Photo\_ID are foreign keys in the Orders table that reference the Users and Photos tables. We had considered making a candidate key for the Orders table using the User\_ID and Photo\_ID, but ruled it out due to the fact that a user could order an image more than once, so an Order\_ID is needed to specify each unique order. The relationship between the Users and Orders table is a one and only one to many relationship while Photos and Orders have a one to zero or many relationship. A user can place many orders but an order can be made by one and only one user. A photo can be ordered zero or many times while an order can include one photo. The Orders table also shares a relationship with the Print\_Orders table. Print\_Orders holds order information for orders that are printed using the print service API, Pwinty. This table will hold the Order\_ID and the Pwinty\_ID, which is the ID that is given by Pwinty. Orders and Print\_Orders have a one to zero or one relationship. An order can be zero or one print order and a print order can be one order. The last table in our design is the Styles table, which holds information on each of the styles that a user can use for the art transfer process. The system will have default styles that the users will be able to choose from to use on their content images. The Styles table will include a Style\_ID, serving as the primary key for each unique style, and the File\_Path, which will hold the file path to the actual file on the storage server. Both of these attributes must have a value and cannot be null. The Styles table shares a one to many relationship with the Photos table, as the Style\_ID is referenced in the Photos table as a foreign key. A style can be used on many photos while a photo can only have one style.

## **IT Requirements:**

### **1. Server Platform**

- LAPP stack (Linux Apache Postgres PHP)
- Test server
- Photo storage server (photos stored locally)

#### **1.1. Physical system requirements**

##### **1.1.1.Storage capacity**

- 100mb per user.

##### **1.1.2.Speed requirements / response time parameters**

- Stylizer should take a maximum of 15 seconds.

##### **1.1.3.Scalability plans**

- More storage per user

#### **1.2. Virtual system requirements**

##### **1.2.1.OS to be supported**

- Linux (Ubuntu)

##### **1.2.2.Number of images expected**

- 1 (Ubuntu)

#### **1.3. Connectivity**

##### **1.3.1.Network considerations**

- Access through browser

##### **1.3.2.Interconnection to what other systems**

- DeepAI (picture stylizing API)
- Printing service (Pwinty)
- Payment service

### **2. Reliability**

#### **2.1. Service Level Agreements**

##### **2.1.1. Uptime requirements**

- Will maintain uptime during 7am-11pm hour range

##### **2.1.2. Response time requirements**

- Within 24 hour period of outage

### **3. Recoverability**

#### **3.1. Where are things backed up? How often?**

- Separate VM server, backed up every two days

#### **3.2. Access to backups?**

- System Administrators

#### **3.3. What data is transient and doesn't need to be stored long term?**

- Original photos
- Photos that have not been paid for in 24 hours since upload

#### 4. Security and Privacy

##### 4.1. Database

###### 4.1.1. Access controls by userid / roles

- System Administrators

###### 4.1.2. Update vs. Access

- System Administrators

##### 4.2. Account information

###### 4.2.1. User data

- System Administrators

##### 4.3. Admin access controls

###### 4.3.1. Adding new users, deleting old

- Users can be deleted after 90 days of inactivity

#### 5. Maintenance

##### 5.1. Planned down time requirements

###### 5.1.1. Database maintenance

- To be completed during inactive site hours (12am-3am)

###### 5.1.3. Times of year when IT does maintenance

- Every three months

###### 5.1.4. Times of year when the systems are not available?

- During maintenance