Xuanyi Chen U24557377

This project will focus on the basic function of album and photos, the tables are created below.
Album_user table is not used since ownerId is already in the album table.
I assume that everyone can comment, but only user can add a tag to the photo.

The installation is included in the readme file. The code is organized in MVC.

Schema:

```
CREATE DATABASE `photo_schema` /*!40100 DEFAULT CHARACTER SET latin1 */;

CREATE TABLE `albums` (
  `albumId` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `ownerId` int(11) NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`albumId`),
  UNIQUE KEY `albumId_UNIQUE` (`albumId`),
  UNIQUE KEY `index3` (`name`,`ownerId`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
;

CREATE TABLE `comment` (
  `commentId` int(11) NOT NULL AUTO_INCREMENT,
  `text` varchar(255) NOT NULL,
  `ownerId` int(11) DEFAULT '0',
  `date` date NOT NULL,
  `photoId` int(11) NOT NULL,
  PRIMARY KEY (`commentId`),
  UNIQUE KEY `commentId_UNIQUE` (`commentId`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

CREATE TABLE `friendship` (
  `userId` int(11) NOT NULL,
  `friendId` int(11) NOT NULL,
  PRIMARY KEY (`userId`,`friendId`),
  UNIQUE KEY `index1` (`userId`,`friendId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `like_photo` (
  `photoId` int(11) NOT NULL,
  `userId` int(11) NOT NULL,
  PRIMARY KEY (`photoId`,`userId`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `photo_album` (
  `photoId` int(11) NOT NULL,
  `albumId` int(11) NOT NULL,
  UNIQUE KEY `index1` (`photoId`,`albumId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


CREATE TABLE `photos` (
  `photoId` int(11) NOT NULL AUTO_INCREMENT,
  `albumId` int(11) NOT NULL,
  `caption` varchar(45) NOT NULL,
  `data` longblob NOT NULL,
  `likeNum` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`photoId`)
) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=latin1;

CREATE TABLE `photo_tag` (
  `photoId` int(11) NOT NULL,
  `tagDescription` varchar(45) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `tags` (
  `description` varchar(45) NOT NULL,
  PRIMARY KEY (`description`),
  UNIQUE KEY `tagId_UNIQUE` (`description`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `users` (
  `userId` int(11) NOT NULL AUTO_INCREMENT,
  `firstName` varchar(100) NOT NULL,
  `lastName` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `birthday` date NOT NULL,
  `hometown` varchar(100) DEFAULT NULL,
  `gender` varchar(6) DEFAULT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`userId`),
  UNIQUE KEY `userId_UNIQUE` (`userId`),
  UNIQUE KEY `email_UNIQUE` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
```

Triggers:

```
DELIMITER //
CREATE trigger add_photo_album_relation after insert on photos
for each row
begin
insert into photo_album (photoId, albumId) values (NEW.photoId, NEW.albumId);
end;//
DELIMITER ;
```

this trigger will auto add the record into photo_album after upload a picture into the database

```
CREATE DEFINER=`root`@`localhost` TRIGGER
`photo_schema`.`albums_BEFORE_DELETE` BEFORE DELETE ON `albums` FOR EACH
ROW
BEGIN
delete from photos where albumId = old.albumId;
delete from photo_album where albumId = old.albumId;
END
```

this trigger will delete all the photo in the album before the album is deleted

```
CREATE DEFINER=`root`@`localhost` TRIGGER
`photo_schema`.`photos_BEFORE_DELETE` BEFORE DELETE ON `photos` FOR EACH
ROW
BEGIN
delete from photo_album where photoId = old.photoId;
delete from photo_tag where photoId = old.photoId;
END
```

this trigger will delete the photo album relation before a photo is deleted

this trigger will auto add a tag into tag table if user insert a new tag

```
DELIMITER //
CREATE DEFINER=`root`@`localhost` TRIGGER
`photo_schema`.`photo_tag_AFTER_INSERT` AFTER INSERT ON `photo_tag` FOR EACH
ROW
BEGIN
        IF NEW.tagDescription not in (SELECT description from tag) then
                insert into tag (description) values (NEW.tagDescription);
        END IF;
END
DELIMITER ;
```

this trigger will auto add a tag into tag table if user insert a new tag