Abstract & Overview

Our team created a web application for a photography studio. In addition to a portfolio for the studio, the application required the ability to manage photographer schedules, handle and organize orders and provide secure user accounts with different access levels for the site.

Theory

Relational database systems and design, based on relational algebra. This relational system dictated most of the design decisions for our project, and is well suited to a project of this type with clearly defined entities.

Requirements Gathering

Requirements were determined through use case diagrams and interviews with art students at SFA. These use cases were created for each role and access level.

Photography Application

Customers
Customers view a portfolio site, create accounts, and place orders.

Photographers
Photographers have a different role, and can view and create schedules so they can be assigned to jobs.

Administrators
Administrators can log in to view photographer schedules, manage users, and manage orders.

Database Design

The database was designed using the relational model and implemented with a SQL database.

The entities and relationships present are the results of assignment constraints and requirements gathering.

Technologies

The LAMP stack is a software suite providing software for the server, scripting, and client-side languages.

AWS provides a scalable architecture and full control over the server instance. It is widely used in industry today. We used a custom EC2 instance that we created from scratch to host the application.

Security & Access Levels

Forms and views are representative of users’ access levels and roles within the system. We queried fields associated with each user that allowed user-group appropriate system access, as established by the owner of the system.

Best practices for account security were used by adding a salt to user's password and then using the current cryptographic hashing standard defined in PHP to preserve system-wide data integrity.