

# Project: UB Room Reservation System

Juana Wofford 1014901

*CPSC551-Project Fall 2018*

PHASE 1

# Abstract

In this paper I present developing an application using Python, MySQL and Django running on Google App Engine. Django is an open-source framework that is designed on top of Python and supports data-driven architecture. The intent is to take advantage of Django's rapid development and security features to quickly implement the UB Reservation System. Google App Engine provides the power and flexibility of being able to write the application code, test it locally and upload it to Google Cloud. App Engine has an automatic scaling feature that will let the application scale to meet future requirements irrespective of the number of users or the amount of data that the application stores.

## Problem Statement

The UB Room Reservation System aims to enable users to easily perform functions related to reserving a room as well as provide useful insights to the administrators, into the reservation process. Reservations are for facilities located on the University of Bridgeport's campus. Facilities are comprised of Meeting Rooms, Study Rooms, Study Halls and Conference Rooms. Students, campus departments, and off-campus groups sponsoring nonprofit events can use the system. The system also allows reservations to be created for Audio / Visual Equipment and Special Events Planning.

At present, there is not an easy way to view insights from reservation data and the reservation processes, which could be used to make the system more economically manageable and provide increased user satisfaction and usability. The approach that will be used to fulfill this goal, is to create a application implemented in MySQL, Python and Django, using Google Cloud Computing.

## Entities and Relationships

### Entities

### Design

The system is organized around a room reservation process. The following provides a high level overview of the requirements for the system.

- Reservation has a unique identifier, description name, room identifier, person who requested the reservation, service rating and the date and time for request. The creation date and creation time of the reservation is tracked.
- A reservation can be associated with any room for a specific time period.
- A room is associated with only one active reservation for a finite time period.
- A room has a description name, room category and room status.
- The database will store each registered user's full name, password, email, city and state.
- A user is identified by a unique identifier and user email. For students, the unique identifier is their student id number. For users other than students, the unique identifier will be their social security number.
- A User can be of type of Admin or Normal exclusively.

- User of type Admin can manage entire system, can access all data and run reports.
- User of type Normal does not have any privileges of user type Admin.
- User types of Admin and Normal can register as users, book a reservation, modify a reservation, cancel a reservation and view status of reservations.
- User of type Normal can send messages to user of type Admin.
- User of type Normal can rate service and quality of a reservation.

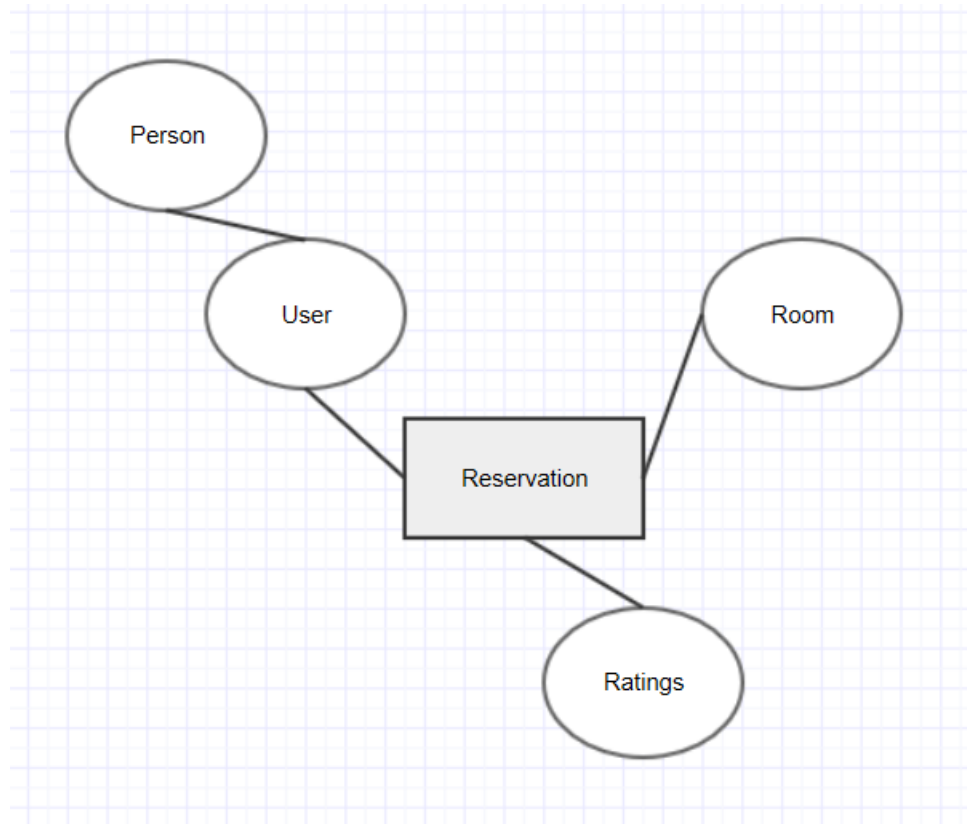
Based on the system design requirements, the following entity types are in the Reservation database:

- Reservation, Reservation Service Rating
- Room, Room Category, Room Status
- Person, User, User Role(Admin, Normal)

## Assumption about the database (in English)

### *Relationships*

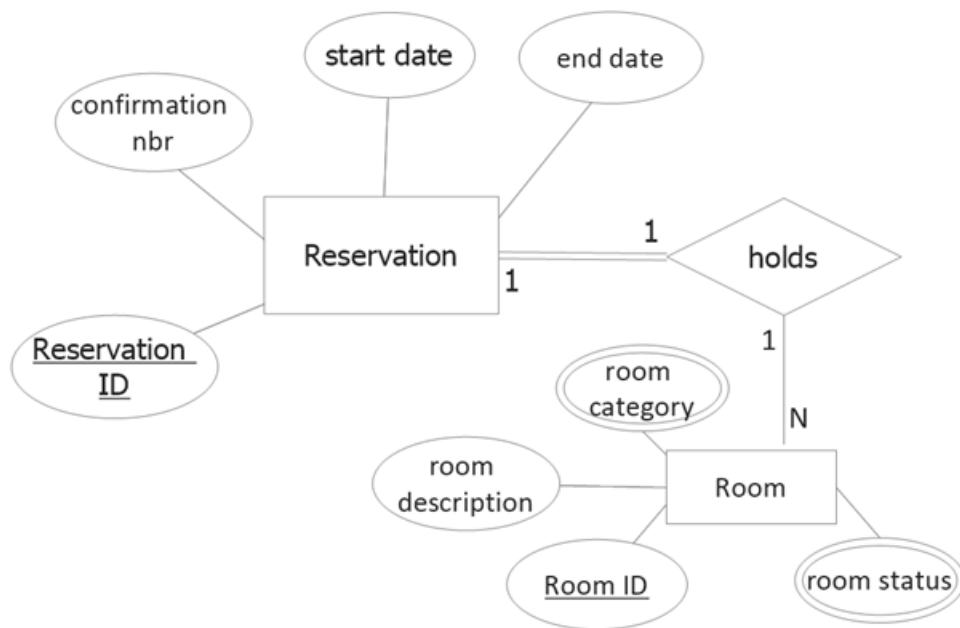
The following is a high level overview of the entities for the reservations system.



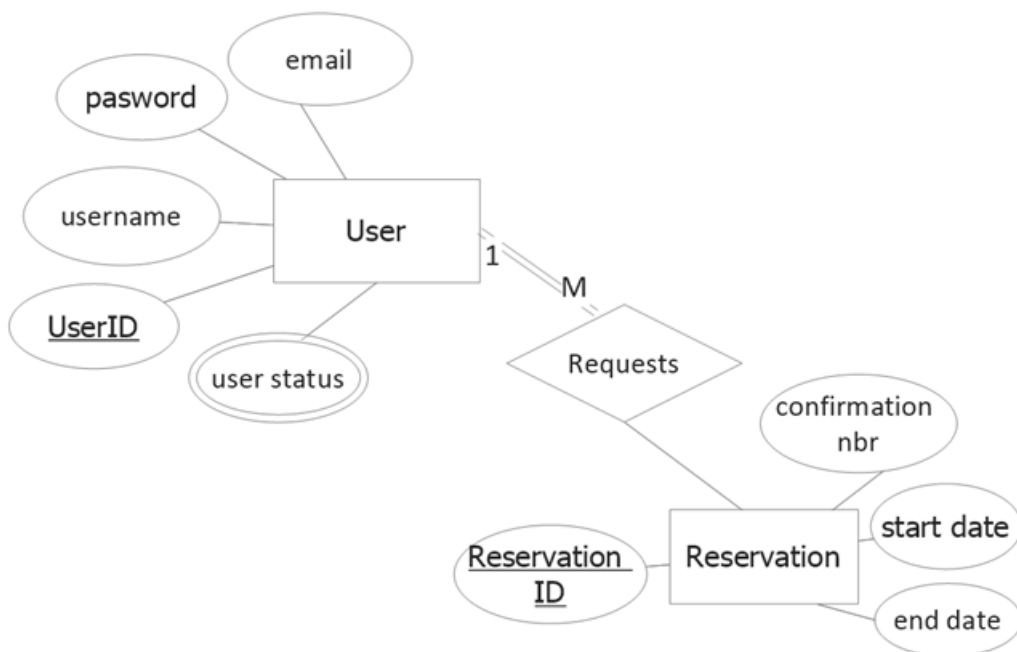
*Relationship - Reservation, Room, User*

Relationship Set Example: A Reservation Holds a Room

HOLDS (between RESERVATION, ROOM)



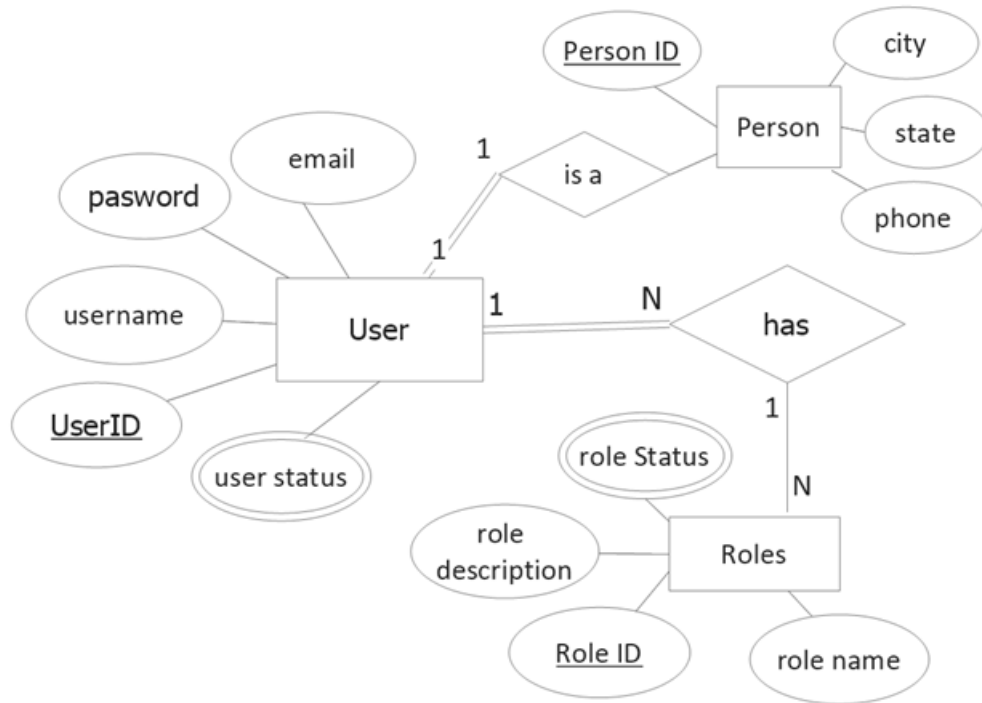
Relationship Set Example: A Reservation is Requested for a Room



Relationship Set Examples: A User has roles in the system , A User Is A Person

HAS (between USER, ROLE)

IS (between USER, PERSON)



## Data Queries (in English)

### *Reservation Information*

1. How many Active Reservations are there in the system today?
2. How many Cancelled Reservations are there in the system today?
3. What time of day are there the most Booked Reservations?
4. What time of day are there the least Booked Reservations?
5. What User cancels the most Reservations?
6. What user modifies the most Reservations?

### *Room Information*

1. What Room is the most requested?
2. What Room is the least requested?
3. What are the Rooms requested by User?
4. What type of Rooms are available for holding more than 5 people?

## *Miscellaneous*

1. What are my open Reservation Requests?
2. How many times have I scheduled the 2nd Floor Conference Room?
3. What conference rooms are available at 1pm on September 6?
4. How many tables are in the 3rd Floor Study Room?
5. What Rooms, dates and times are available that have SmartBoards next month?

## **Application Implementation Architecture**

### *Google Cloud Architecture Application*

- Database - MySQL
- Code - Python
- Framework - Django

## **Conclusion - What Was Learned (Learning)**

### *Google Cloud Architecture*

Learned differences of Google Compute Engine and Google App Engine and what each could be used for. App Engine is used to deploy code and the platform handles its maintenance and growth. Compute Engine is a virtual machine running in Google Cloud. It gives you more control over managing your code and applications. Also learned that both could be used together.

### *MySQL*

Learning how to configure MySQL on Compute Engine. Learning how to deploy MySQL on App Engine through Python.

### *Database Concepts*

Learning how to apply Data Modeling Using the Entity Relationship Model for the project.