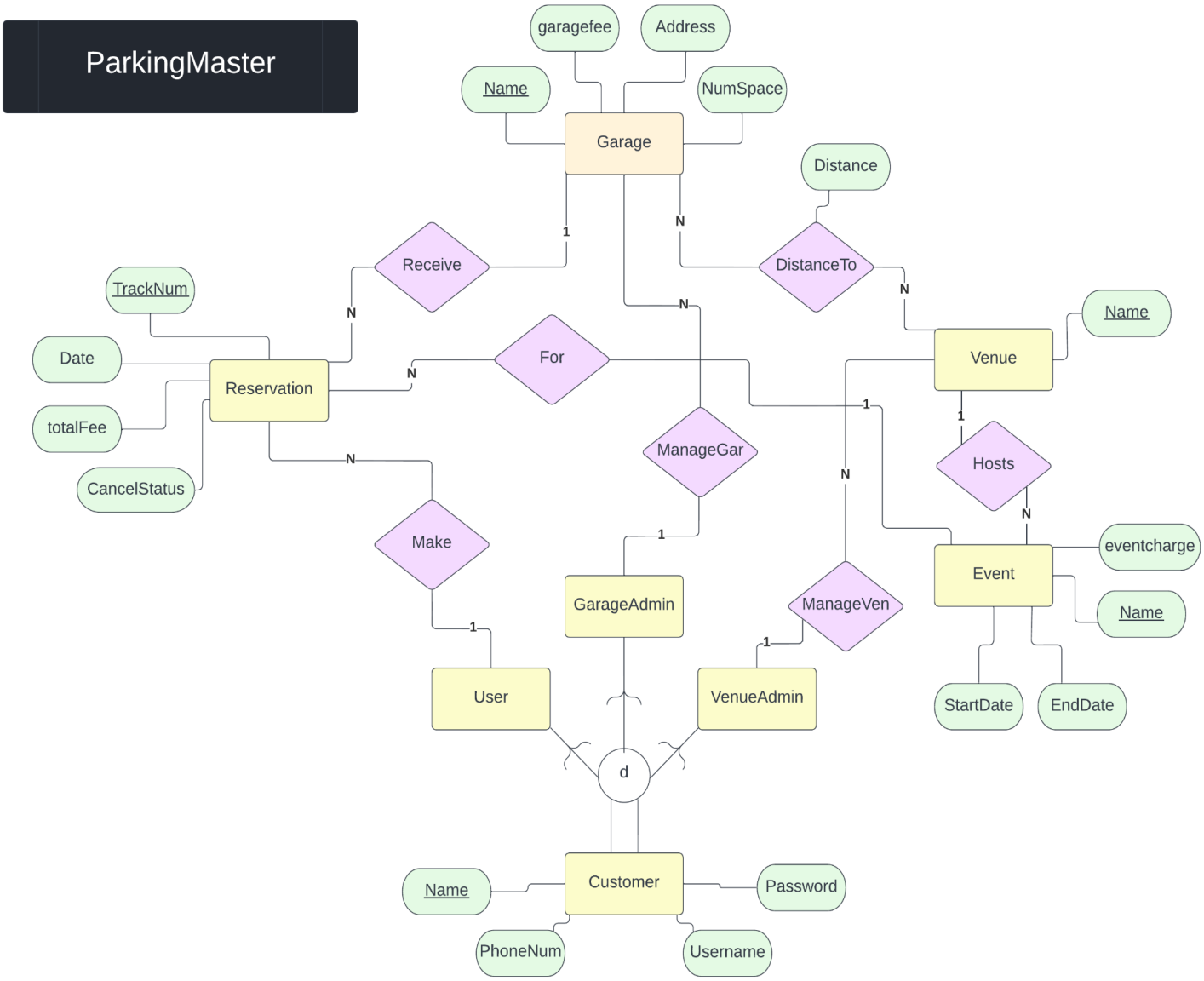
EER Diagram: (made with lucidcharts.com)



Description of requirements and implementation:

Key attributes are underlined, all attributes are shaded green

Entities (Yellow):

Customer - Generated for anyone who wishes to use the ParkingMaster application. They must first provide information fulfilling the attributes: Name, Phone Number, Username, and Password before continuing. This is basic profile information that will help uniquely identify each individual customer.

User - A subclass of Customer, this will be the profile type for most users as these will the customers that are trying to place parking reservations.

VenueAdmin- A subclass of Customer, this will be the profile type for a venue administrator wishing to use the application in order to manage venues/events. This will help identify who is responsible for creating/supervising new events at venues.

GarageAdmin- A subclass of Customer, this will be the profile type for a garage administrator wishing to use the application in order to manage garages/parking fees. This will help identify who is responsible for supervising individual garages.

Reservation - Created anytime a user wishes to create a parking reservation for a certain event. This holds information such as: Tracking Number, Date, Fee, Cancellation Status, Event, and Customer. Making each reservation hold so much information will help with sorting and searching queries for the application functionalities.

Garage - This entity will be assumed to be pre-existing as garages must exist for the scenario described in the project overview. Each individual garage will contain information such as: Name, Address, and Number of total parking spaces.

Event - Created/edited by VenueAdmin and holds information: Name, Venue, StartDate, EndDate. These events play a huge role in deciding garage pricing as relative distance to the event will be correlated to the fee.

Venue - Managed by VenueAdmin and has a unique attribute of “Name”. Venues are where events will be hosted.

Relationships(Purple):

Receive - A garage receives a reservation after showing all of its availability on the desired date for a desired event. This relationship will also hold the total number of reservations currently placed for each individual garage.

DistanceTo- Each garage shares a proximity relationship with each venue that directly correlates to how much the parking will cost and has information about the total distance.

Make - Users will make reservations as they please, not being limited in any way. These reservations act as transaction records.

For - Each reservation will be placed for a certain event, maintaining a relationship where any number of reservations can be placed for the same event, as long as parking space permits.

ManageVen - Venue Administrator will manage the venues and be able to change information about it that may change the respective parking fees or dates of availability. They can also create/delete/manipulate events at the existing venues

ManageGar - Garage Administrator will manage the parking garages serviced by the application. Being able to change the actively accessible number of spaces at any garage.