**Project Description:**

The PFA Employee Organizer is an application developed for managing the database of the fictional business PFA Balloon Company. PFA Balloon company offers hot air balloon rides and has a lot of data to keep track of daily. This application allows the user to access the company’s database and view data about the balloon rides scheduled for that day, as well as the customers and employees who will be participating; this is done via a graphical user interface, without the user having to use SQL commands.

Specifically, users can see which employees are assigned to which rides, all employee identification information, which customers have paid for tickets to which rides, customer id’s names and phone numbers, and where each ride is located, its price, duration, and number of passengers.

**Business rules:**

The PFA Balloon Company is a new company that offers balloon rides at an affordable price. As the company’s popularity has increased it’s become difficult to keep track of all the day-to-day operations. Therefore, the company created a database of its operations and an employee organizer to access the data. The data in the database is as follows:

The PFA Balloon Company has many employees. Each employee has a name, employee ID, salary, phone number, date of birth, certification date, and date of hiring. The employees operate the balloon rides. The rides have a ride ID, location, price, duration, and date and time. An employee may operate many rides, and each ride is operated by 1 or more employees.

Customers pay for the balloon rides. The customers have a name, customer ID, phone number, and date of birth. A customer may pay for many rides, and each ride is paid for by many customers.

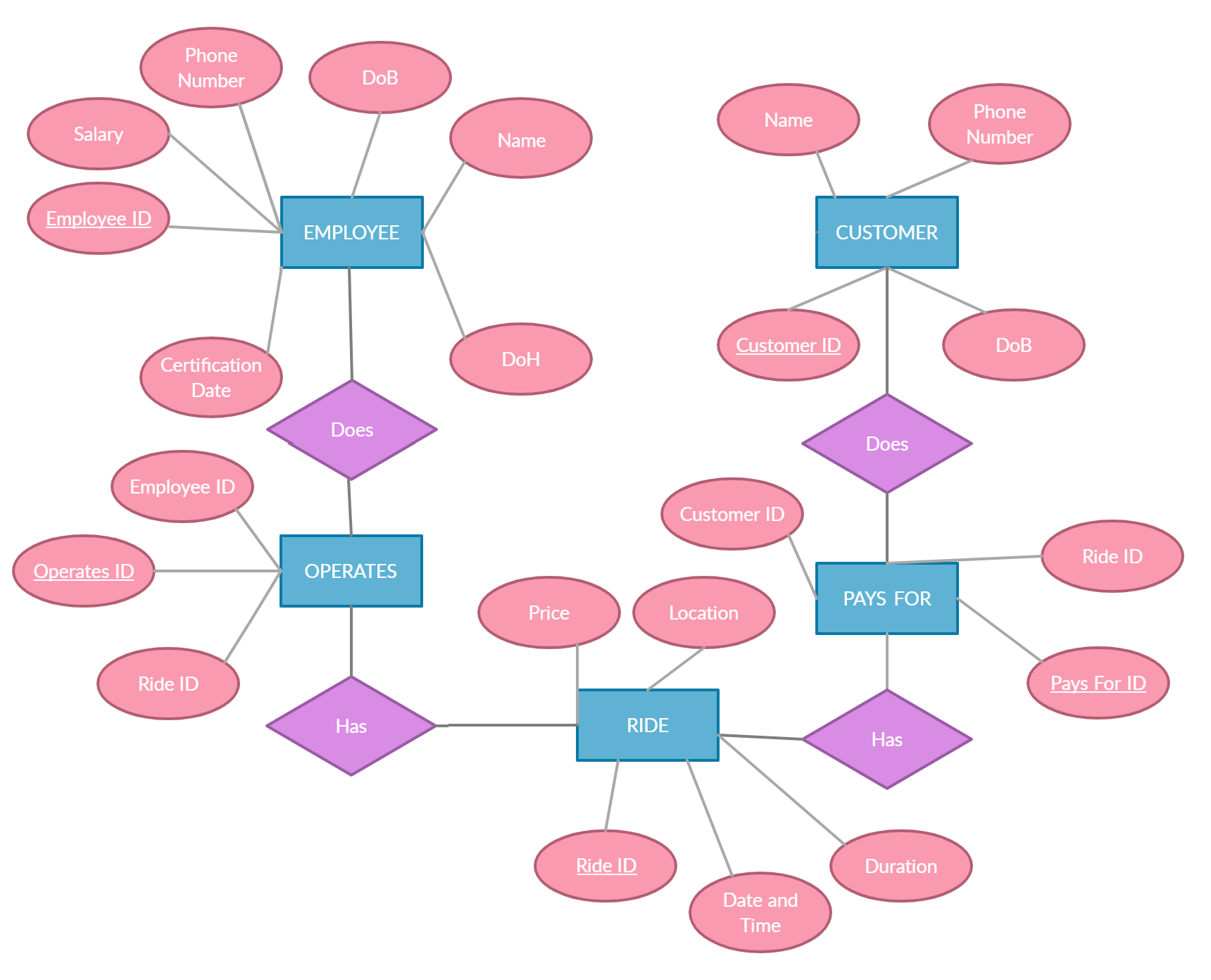
**Database Structure:**

|  |  |
| --- | --- |
| **Database Table** | **Description** |
| Customer | Contains the customers’ personal identifying information. |
| Employee | Contains the employees’ personal identifying information. |
| Ride | Contains information about the rides such as their location, price and duration. |
| PaysFor | Links customer and ride—which customers paid for which rides |
| Operates | Links Employee and ride—which employees work on which rides |

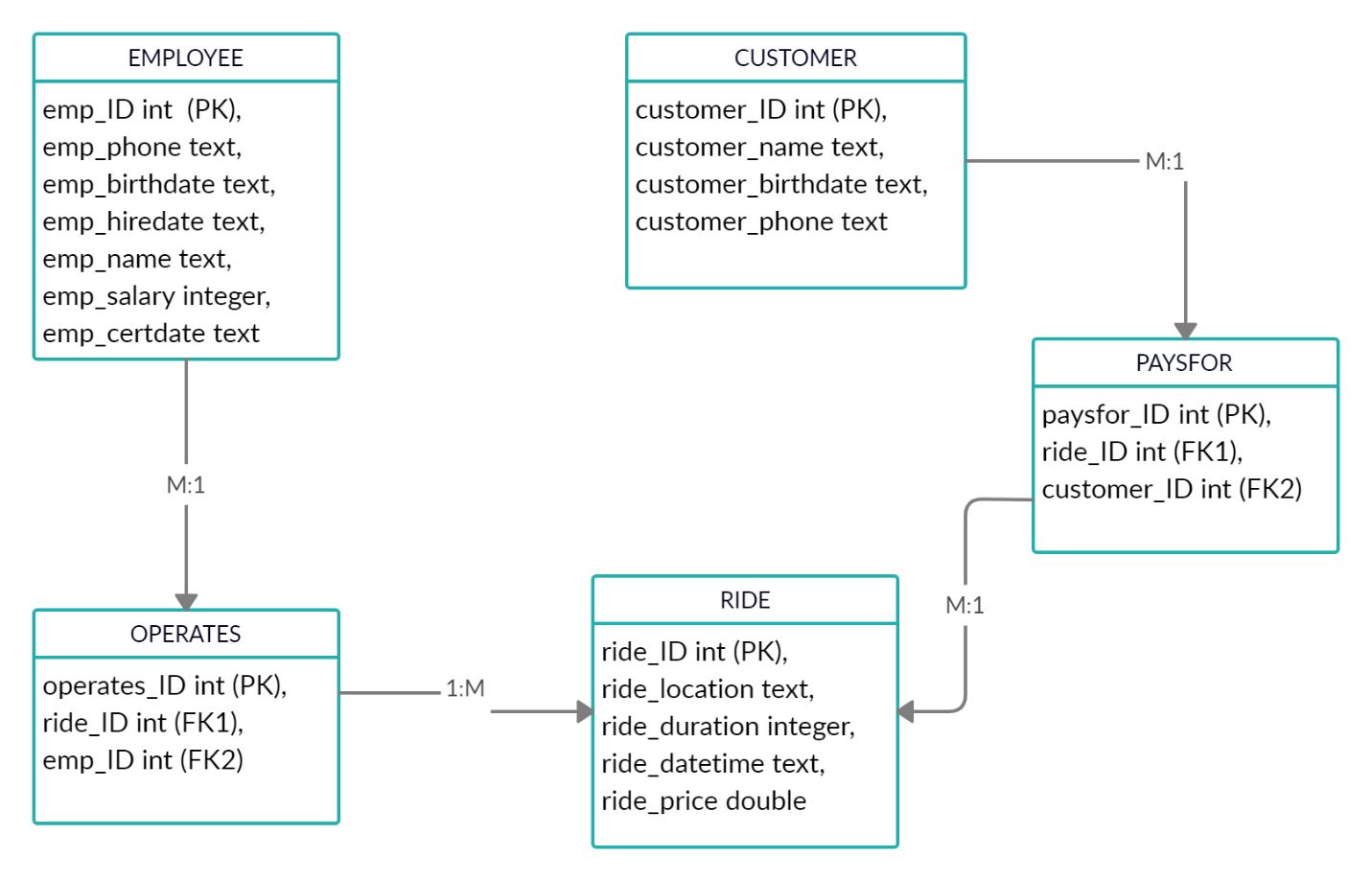
**Database Design Note:**

Because a customer can pay for many rides and a ride can have many customers, it is a many to many relationship, and we created the new table Paysfor to link them together. We did the same with the relationship of Employee to Ride: we created the table Operates to link them. We could not have a many to many relationship in 3rd Normal Form.

**Chen’s Model ER Diagram:**



**UML Model ER Diagram:**



**List of Source Code and Description:**

Our GUI application has 3 dropdown menus on the main screen: Employee, Customer, and Ride. The user make a selection from any one of those categories, and a screen will pop up giving information. Users can view the identifying information of the employees and customers, as well as their upcoming ride schedules. The Ride selection gives information about the ride and a list of the employees who will be operating it.

**Customer.java** stores the customer information that was retrieved from the database

**Employee.java** stores the Employee information that was retrieved from the database

**Ride.java** stores the Ride information that was retrieved from the database

**CustomerDataAccessor.java** retrieves information about Customer

**EmployeeDataAccessor.java** retrieves information about Employee

**RideDataAccessor.java** retrieves information about Ride

**EmployeeWindow.java** popup window that presents Employee information

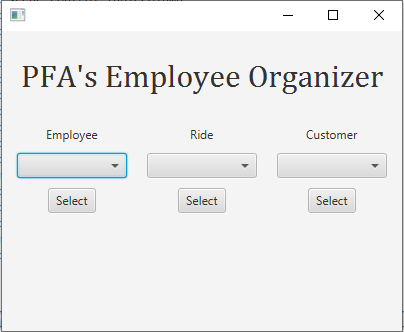
**RideWindow.java** popup window that presents Ride information

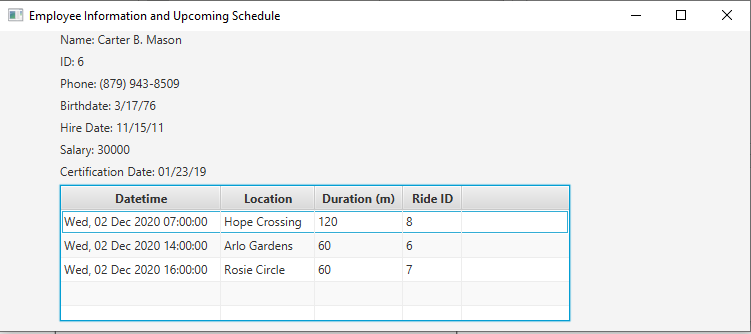
**CustomerWindow.java** popup window that presents Customer information

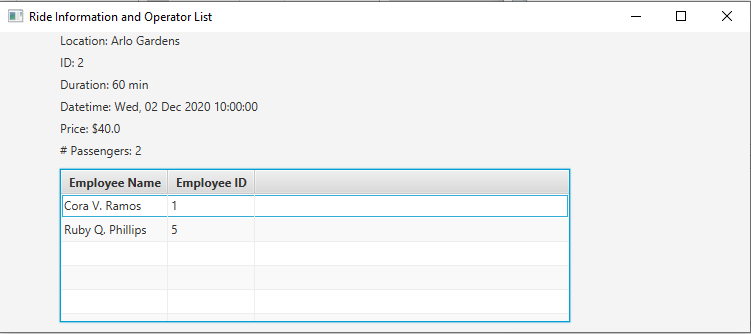
**Main.java** the entry point to the program

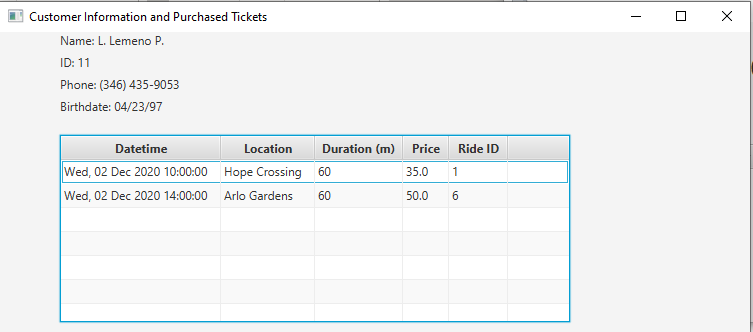
**pfa\_db** directory that contains the data for the project

**Screenshots of GUI:**









**SQL Table Descriptions:**

