

# 233 Project: PasoEats

Connor Healey-Wolf, Reese Werling, Brian Rey

## Project Design

Each section of our project is separated into different modules that work together.

There are three User classes, which inherit the abstract class User. These are Driver, Customer, and Administrator. These classes' objects and storage are managed using UserManager.

Additionally, the driver pool functionality has its own class, DriverPool.

The restaurant functionality is implemented using the Menu and MenuItem classes. These are utilized by the Restaurant class, which is managed using the Restaurant Manager.

Orders are handled with the Order class and its manager, OrderManager.

There is a class called FileManager which reads and writes from files in the correct formatting.

This is utilized through the OrderManager, RestaurantManager, and UserManager classes.

Finally, there is a command line interface implemented using the CLI class. The CLI uses another class, AppController, for its logic. This allows for further development into other interfaces, like a GUI, without rewriting the logic that ties all the classes together functionally.

Class diagrams are located at the bottom of this document.

## Data Structures

For handling files, we chose to use HashMaps. This efficiently stores objects based on unique identifiers, allowing for quick addition and removal to active memory.

For storing things inside classes, we used ArrayLists because of their dynamic sizing, which prevents wasting space. They also allow for efficient lookup and addition.

In OrderManager, a Queue is used to store Orders. This allows for them to be handled in the correct order.

DriverPool uses a Priority Queue. This allows us to retrieve Drivers by not only the order they come in, but also by a ranked rating system.

Driver uses an array of the fixed size 10 to store ratings, because Drivers can only have 10 ratings at a time.

## Functionality

Because we all worked together in a past project, we knew we needed to separate all of our different functionalities into their own sections. For example, we successfully separated the logic for the interface from the actual interface. This was successful, and now it is easier to maintain them separately from one another. Keeping this in mind from the beginning was very beneficial to the project, as now our project is very functional.

## Challenges

Our group did not always understand our project design the same way, leading to some stumbles that had to be fixed later on. Additionally, we did not manage our time as efficiently as we could have.

## Future Work

In the future, it would be ideal to store persisting data somewhere that is not local to a single computer.

For easier access to the project, having an online graphical user interface rather than command line prompts would be a good idea.

## Group Work Breakdown

In addition to random fixes here and there by everyone, this is the general breakdown:

Administrator - Connor  
AppController - Brian, Reese  
CLI - Reese, Brian, Connor  
Customer - Connor  
Driver - Connor  
DriverPool - Brian  
FileManager - Reese  
Menu - Reese  
MenuItem - Brian  
OrderManager - Reese  
Order - Reese  
Restaurant - Brian  
RestaurantManager - Brian  
User - Connor  
UserManager - Reese, Brian

## Class Diagrams

[Google Drive of PNGs for ease of access](#)

## **Administrator**

extends User

### **Variables**

*(No attributes)*

### **Methods**

```
public Administrator(UUID id, String name, String  
username, String email)
```

```
public void manageRestaurants()
```

## AppController

### Variables

```
protected FileManager fileManager
protected RestaurantManager restaurantManager
protected OrderManager orderManager
protected DriverPool driverPool
protected UUID currentUserID
protected UserRole currentUserRole
package-private return null
```

### Methods

```
public AppController()
public Customer loginCustomer(String username)
public Driver loginDriver(String username)
public Administrator loginAdministrator(String username)
public void logout()
public boolean isLoggedIn()
public List getAllRestaurants()
public String getAllRestaurantsString()
public Restaurant getRestaurant(UUID restaurantId)
public boolean addRestaurant(String name, String category)
public boolean removeRestaurant(UUID restaurantId)
public boolean updateRestaurant(UUID restaurantId, String newName, String newCategory)
public String getRestaurantMenuString(UUID restaurantId)
public boolean addMenuItemToRestaurant(String name, String category, double price, UUID restaurantId)
public Customer getCurrentCustomer()
public boolean addCustomer(String name, String username, String email)
public Customer createCustomerAccount(String name, String username, String email)
public boolean deleteCustomer(UUID customerId)
public boolean addNewDriver(String name, String username, String email)
public Driver createDriverAccount(String name, String username, String email)
public boolean deleteDriver(UUID driverId)
public boolean addAdministrator(String username, String name, String email)
public Administrator createAdministratorAccount(String username, String name, String email)
public boolean deleteAdministrator(UUID adminId)
public boolean updateDriverAvailability(boolean available)
public double getCurrentDriverRating()
public OrderManager.Order placeOrder(UUID customerId, List items)
public UUID getCurrentUserID()
public UserRole getCurrentUserRole()
public FileManager getFileManager()
public RestaurantManager getRestaurantManager()
public DriverPool getDriverPool()
public OrderManager getOrderManager()
```

## CLI

extends AppController

### Variables

```
private Scanner scanner
```

### Methods

```
public CLI()  
public void start()  
public void shutdown()  
public void printWelcome()  
public void printLogin()  
public void mainMenuLogic(int input)  
public void printCustomerMenu()  
public void printDriverMenu()  
public void printAdministratorMenu()  
public void handleCustomerMenu()  
public void handleDriverMenu()  
public void handleAdministratorMenu()  
private int readIntInput(String prompt)  
private String readStringInput(String prompt)  
private void waitForEnter()
```

## **Customer**

extends User

### **Variables**

*(No attributes)*

### **Methods**

```
public Customer(UUID id, String name, String username,  
String email)
```

```
public String getRestraunts(RestaurantManager  
restaurantManager)
```

```
public String getMenu(RestaurantManager  
restaurantManager, UUID restaurantId)
```

```
public void placeOrder(OrderManager orderManager, List  
items)
```

```
public void rateDriver(OrderManager orderManager,  
UserManager userManager, int rating)
```

## Driver

extends User implements Comparable<Driver>

### Variables

```
private boolean available  
private double avgRating  
private int rateCounter  
private int[] ratings  
private UUID currentOrder
```

### Methods

```
public Driver(UUID id, String name, String username, String email)  
public boolean isAvailable()  
public void setAvailable(boolean available)  
public double getAvgRating()  
public void setAvgRating(double avgRating)  
public int getRateCounter()  
public void setRateCounter(int rateCounter)  
public int[] getRatings()  
public void setRatings(int[] ratings)  
public UUID getCurrentOrder()  
public void setCurrentOrder(UUID currentOrder)  
public void updateOrderStatus(OrderManager  
orderManager, OrderManager.Status status)  
public void calcAvgRating()  
public void addRating(int newRating)  
public int compareTo(Driver other)
```

## DriverPool

### Variables

```
private PriorityQueue driverPool  
private FileManager fileManager
```

### Methods

```
public DriverPool(FileManager fileManager)  
public void addDriver(Driver driver)  
public void removeDriver(Driver driver)  
public Driver getNextAvailableDriver()  
public void completeDelivery(Driver driver)  
public int getPoolSize()  
public boolean isEmpty()  
public void clearPool()  
public void clearAndUpdatePool(FileManager  
fileManager)  
public void updatePoolDrivers(FileManager fileManager)
```

## FileManager

### Variables

```
private String MENU_FILE
private String RESTAURANTS_FILE
private String CUSTOMERS_FILE
private String DRIVERS_FILE
private String ADMINS_FILE
private String ORDERS_FILE
private List menuItemsList
private List restaurantsList
private UUID orderId
private UUID customerId
private UUID restaurantId
private List itemIds
private String status
private UUID driverId
private String createdAt
private double totalPrice
```

### Methods

```
public FileManager()
package-private public OrderData(UUID orderId, UUID
customerId, UUID restaurantId, List itemIds, String status,
UUID driverId, String createdAt, double totalPrice)
public UUID getOrderId()
public UUID getCustomerId()
public UUID getRestaurantId()
public List getItemIds()
public String getStatus()
public UUID getDriverId()
public String getCreatedAt()
public double getTotalPrice()
public void setStatus(String status)
public void setDriverId(UUID driverId)
private void createFiles()
private void createFile(String filename, String header)
private void loadFromFile(String filename, String
entityName, java.util.function.Consumer<parser>)
private void loadRestaurants()
private void loadMenuItems()
private void loadCustomers()
private void loadDrivers()
private void loadAdmins()
private void loadOrders()
public boolean addRestaurant(UUID restaurantId, String
name, String category)
public boolean removeRestaurant(UUID restaurantId)
public boolean updateRestaurant(UUID restaurantId,
String newName, String newCategory)
public Restaurant getRestaurant(UUID restaurantId)
public List getAllRestaurants()
public boolean addMenuItem(UUID itemId, String name,
String category, BigDecimal price, UUID restaurantId)
```

```
public boolean removeMenuItem(UUID itemId)
public boolean updateMenuItem(UUID itemId, String
newName, String newCategory, BigDecimal newPrice, UUID
newRestaurantId)
public MenuItem getMenuItem(UUID itemId)
public List getMenuItemsForRestaurant(UUID
restaurantId)
public List getAllMenuItems()
public boolean addCustomer(UUID customerId, String
username, String name, String email)
public boolean removeCustomer(UUID customerId)
public Customer getCustomer(UUID customerId)
public Customer getCustomerByUsername(String
username)
public boolean customerUsernameExists(String username)
public boolean customerEmailExists(String email)
public boolean addDriver(UUID driverId, String username,
String name, String email, boolean available)
public boolean removeDriver(UUID driverId)
public boolean updateDriver(UUID driverId, boolean
available, double avgRating)
public Driver getDriver(UUID driverId)
public Driver getDriverByUsername(String username)
public boolean driverUsernameExists(String username)
public boolean driverEmailExists(String email)
public boolean addAdmin(UUID adminId, String
username, String name, String email)
public boolean removeAdmin(UUID adminId)
public Administrator getAdmin(UUID adminId)
public Administrator getAdminByUsername(String
username)
public boolean adminUsernameExists(String username)
public boolean adminEmailExists(String email)
public boolean appendOrder(UUID orderId, UUID
customerId, UUID restaurantId, List itemIds, String status,
UUID driverId, String createdAt, double totalPrice)
private String formatOrderLine(UUID orderId, UUID
customerId, UUID restaurantId, List itemIds, String status,
UUID driverId, String createdAt, double totalPrice)
public boolean updateOrder(UUID orderId, String
newStatus, UUID driverId)
public OrderData getOrder(UUID orderId)
private boolean appendToFile(String filename, String line)
private boolean processFileLines(String filename, String
operation, Unknown java.util.function.FunctionString>
processor)
private boolean removeFromFile(String filename, String
id)
private boolean updateInFile(String filename, String id,
String newLine)
private boolean updateInFile(String filename, UUID id,
Unknown java.util.function.FunctionString> replacer)
private boolean updateInFile(String filename, String id,
Unknown java.util.function.FunctionString> replacer)
private boolean updateDriverInFile(UUID driverId, Driver
driver, boolean available, double avgRating)
private boolean updateOrderInFile(UUID orderId,
OrderData order)
public static void main(String[] args)
```

## Menu

### Variables

*(No attributes)*

### Methods

```
public void add(MenuItem item)
public boolean remove(String name)
public MenuItem get(String name)
public Collection all()
public void save(Path path)
```

## MenuItem

### Variables

```
private UUID itemId  
private String name  
private String category  
private double price  
private UUID restaurantId
```

### Methods

```
public MenuItem(String name, String category, double  
price, UUID restaurantId)  
public String getName()  
public void setName(String name)  
public String getCategory()  
public void setCategory(String category)  
public double getPrice()  
public void setPrice(double price)  
public UUID getItemId()  
public UUID getRestaurantId()  
public void setRestaurantId(UUID restaurantId)  
public String detailsToString()
```

## OrderManager

### Variables

```
private UUID id  
private UUID customerId  
private List items  
private Status status  
private UUID assignedDriverId  
private String createdAt  
private ArrayDeque intake  
package-private return byld
```

### Methods

```
public OrderManager()  
package-private public Order()  
package-private public Order(UUID id, UUID customerId,  
List items)  
public UUID getId()  
public UUID getCustomerId()  
public List getItems()  
public Status getStatus()  
public UUID getAssignedDriverId()  
public String getCreatedAt()  
public void setStatus(Status status)  
public void setAssignedDriverId(UUID driverId)  
public Order place(UUID customerId, List items)  
public Order acceptNext(UUID driverId)  
public void markStatus(UUID orderId, Status newStatus)  
public Order get(UUID orderId)  
public static void main(String[] args)
```

## Restaurant

### Variables

```
private UUID restaurantId  
private String name  
private String category
```

### Methods

```
public Restaurant(UUID restaurantId, String name, String  
category)  
public void setRestaurantId(UUID restaurantId)  
public UUID getRestaurantId()  
public String getName()  
public String getCategory()  
public void setName(String name)  
public void setCategory(String category)  
public List getMenuItems(FileManager fileManager)  
public String getMenuItemsToString(FileManager  
fileManager)  
public String toFileFormat()  
public String detailsToString()
```

## RestaurantManager

### Variables

```
private FileManager fileManager
```

### Methods

```
public RestaurantManager(FileManager fileManager)
public FileManager getFileManager()
public List getRestaurants()
public boolean addRestaurant(String name, String
category)
public boolean removeRestaurant(UUID restaurantId)
public Restaurant findRestaurantByld(UUID restaurantId)
public boolean updateRestaurant(UUID restaurantId,
String newName, String newCategory)
public boolean addRestaurantMenuItem(String name,
String category, BigDecimal price, UUID restaurantId)
public boolean deleteRestaurantMenuItem(UUID itemId)
public boolean modifyRestaurantMenuItem(UUID itemId,
String newName, String newCategory, BigDecimal
newPrice, UUID newRestaurantId)
public List getMenuItemsForRestaurant(UUID
restaurantId)
public String getAllRestaurantsString()
```

# User

## Variables

```
protected UUID id  
protected String name  
protected String username  
protected String email  
protected ArrayList orders
```

## Methods

```
public User()  
public User(UUID id, String name, String username, String  
email)  
public UUID getId()  
public void setId(UUID id)  
public String getName()  
public void setName(String name)  
public String getUsername()  
public void setUsername(String username)  
public String getEmail()  
public void setEmail(String email)  
public ArrayList getOrders()  
public void setOrders(ArrayList orders)  
public void addOrder(UUID orderId)
```

## UserManager

### Variables

```
private String USERS_FILE  
private String DRIVERS_FILE  
private String ADMINS_FILE
```

### Methods

```
public UserManager()  
public Driver getDriver(UUID driver)
```